Exploring use of the Bridge21 model as a 21st Century method of Continuing Professional Development (CPD) in Computer Science (CS) for Teachers in Ireland

2019

Lorraine Fisher

B.A (RGU), Pg. Dip (RGU), M.Sc. (RGU), M.Sc. (DUB)

A thesis submitted in fulfilment of the requirements for the award of Doctor of Philosophy

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis on the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Signed: ____

Lorraine Fisher, B.A (RGU), Pg. Dip (RGU), M.Sc. (RGU), M.Sc. (DUB)

Acknowledgements

For Niamh and Mike

To Professor Brendan Tangney and Dr Jake Rowan Byrne

School of Computer Science and Statistics, Trinity College Dublin, University of Dublin, Ireland

To Dr Katriona O'Sullivan

School of Education, National University of Ireland, Maynooth, Ireland

For my colleagues in Bridge21.ie, Centre for Research into I.T. in Education (CRITE), Trinity Access Programme, University College Dublin, Scouting Ireland, OurKidsCode.ie, and Google, Inc.

"*It is hard to know the potential of a new medium unless someone provides a starting point*" (Ridgway & Passey, 1991, p. 7).

Dedication

In memory of Donald Kirkpatrick (1924-2014)

"*I also hope that you acknowledge and remember the power of genuine person-to-person interaction in training and evaluation. Surveys and technologies are fine, but the bridges that you build with your trainees and their managers, and the conversations you have with them, will add humanness to training content and evaluation data*" (Kirkpatrick, 2016, p. xi).

"Science is an ongoing process. It never ends. There is no single ultimate truth to be achieved, after which all the scientists can retire. And because this is so, the world is far more interesting, both for the scientists and for the millions of people in every nation who, while not professional scientists, are deeply interested in the methods and findings of science" (Sagan, 1980, p. xix).

Tables

Table 1 Research Questions....5

Table 2 Bridge21 Pedagogical Model (Lawlor et al., 2018)....7

Table 3 Bridge21 Activity Model (Byrne, Fisher, & Tangney, 2016)....7

Table 4 Kirkpatrick (1994) Levels....9

Table 5 SWOT Analysis applied to Kirkpatrick (1994)....11

Table 6 Philosophical Approach adapted from Crotty (1998)....12

Table 7 Methodological Approach Adapted from Yin (2003)....13

Table 8 Kirkpatrick (1994) Adapted for Bridge21 CS CPD Evaluation....14

Table 9 Search Strategy Applied to Databases....23

Table 10 Meta-Synthesis Model and Procedure....24

Table 11 Computer Programming Difficulties (Du Boulay, 1986)....31

Table 12 Common Programming Mistakes (Du Boulay, 1986)....35

Table 13 Barriers to Learning Programming....37

Table 14 Barriers to Teaching Programming (Robins et al., 2003)....39

Table 15 Rational for Collaborative, Project-Based Teaching in CS....42

Table 16 Collaborative CS CPD Model for Teachers (Sentance & Csizmadia, 2017b)....52

Table 17 Comparison between Guskey (2000) and Kirkpatrick Model (1956)....63

Table 18 Bridge21 Pedagogical / Activity Models Applied in a Subject Context....77

Table 19 TCD Pg. Cert 21st Century Teaching and Learning Module List (TCD, 2018)....79

Table 20 Ontological Paradigms Adapted from Guba and Lincoln (2000)....85

Table 21 Interpretivist Research Designs adapted from Creswell (2005)....90

Table 22 CPD and School Samples....93

Table 23 Table Shell Linking Research Questions to Data Sets (Miles and Huberman, 1994)....98

Table 24 Piloting CPD and School Questionnaires....103

Table 25 Validity Framework for Case Study adapted from Yin (2003)....110

Table 26 Data Analysis Model....115

Table 27 Logical Models for Developing Concepts (LeCompte & Schensul, 2013)....120

Table 28 Using a Taxonomy to Guide Coding (LeCompte & Schensul, 1999)....127

Table 29 Algorithms for Generating Statistical Results....130

Table 30 Reaction Post-Workshop Variables All Years 140

Table 31 Reaction Variables – Mean – Year on Year 144

Table 32 Learning Outcome Pre-Workshop Variables adapted from TCD (2017) 155

Table 33 Learning Outcome Post-Workshop Variables adapted from TCD (2017) 156

Table 34 Learning Outcome Variables – Mean – Year on Year 162

Table 35 Module Workshop and Paired Matched Sub-Sample Size...................................... 168

Table 36 Wilcoxon (1945) Signed Rank Test First Pre Learning Outcome Results 172

Table 37 Wilcoxon (1945) Signed Rank Test Last Post Learning Outcome Results 172

Table 38 Behavioural Intention Post-Workshop Variable All Years....................................... 177

Table 39 Intention to use Bridge21 All Years 177

Table 40 Post Workshop Intentions – Year on Year Results 178

Table 41 Bridge21 CS CPD Computing Workshops Attended by CPD Teachers...................... 187

Table 42 Bridge21 elements (Lawlor et al., 2018; Lawlor et al., 2013; Lawlor et al., 2016) 190

Table 43 Barriers Post CPD Variables (Hogarty, Lang, & Kromrey, 2003; Kirkpatrick, 2007) ... 194

Table 44 Other Methods for Teaching Computing (Sentance & Csizmadia, 2017a) 200

Table 45 Further CPD in using the Bridge21 Model (Lawlor et al., 2013) 203

Figures

Figure 3-1 Bridge21 Learning Pod Space (Reprinted with Permission, CPD Teacher, 2013) 73

Figure 3-2 Bridge21 Brainstorming Computers in Every Day Life Activity Example..................75

Figure 3-3 Bridge21 Auditorium Space (Reprinted with Permission, CPD Teacher, 2013) 76

Figure 3-4 Scratch Visual Programming Language Example 81

Figure 3-5 Using the Python programming language to activate a switch 83

Figure 4-1 Quantitative Research Database IBM SPSS 107

Figure 4-2 Qualitative Research Database QSR International NVIVO 107

Figure 4-3 Ethics Database – Microsoft Excel 108

Figure 6-1 Teacher CPD Demographics Age Profile...138

Figure 6-2 CPD Workshop Module Frequency Analysis 139

Figure 6-3 Reaction Variable Analysis – Summary - All Years 140

Figure 6-4 Reaction Post-Workshop Variables Year 1 145

Figure 6-5 Reaction Post-Workshop Variables Year 1 / Year 2 146

Figure 6-6 Reaction Post-Workshop Variables Year 2 / Year 3 147

Figure 6-7 Reaction Post-Workshop Variables Year 3 / Year 4 148

Figure 6-8 Reaction Post-Workshop Variables Year 4 / Year 5 149

Figure 6-9 A Team working on a Bridge21 Computing Activity (Fisher et al., 2016)................151

Figure 6-10 Pre Workshop Learning Outcome Variable Analysis – Summary 157

Figure 6-11 Post Workshop Learning Outcome Variable Analysis – Summary........................157

Figure 6-12 Learning Outcome Post-Workshop Variables Year 2 163

Figure 6-13 Learning Outcome Post-Workshop Variables Year 2 / Year 3 164

Figure 6-14 Learning Outcome Pre and Post-Workshop Variables Year 4 165

Figure 6-15 Learning Outcome Pre and Post-Workshop Variables Year 5 166

Figure 6-16 Teachers Programming in Scratch using the Raspberry Pi (Byrne, Fisher, & Tangney, 2015b)...174

Figure 6-17 Intentions Post-Workshop Variables Year 3, 4 and 5...179

Figure 6-18 Teachers Operating Scratch using Makey-Makey Controllers 181

Figure 6-19 Breakdown of Primary and Post-Primary Teacher Sample 188

Figure 6-20 Teaching Computing in Schools Profile 189

Figure 6-21 Analysis of the First Four Bridge21 Elements used for Teaching CS 191

Figure 6-22 Analysis of the Second Four Bridge21 Elements used for Teaching CS 192

Figure 6-23 Barrier Variable Analysis – First Four Variables 195

Figure 6-24 Barrier Variable Analysis – Second Four Variables 196

Figure 6-25 Barrier Variable Analysis – Remaining Four Variables 197

Figure 6-26 Other Methods Variable Analysis – First Four Variables 200

Figure 6-27 Other Methods Variable Analysis – Remaining Four Variables 201

Contents

1 INTRODUCTION 1

1.1 RESEARCH CONTEXT 1

1.2 RESEARCH QUESTIONS 5

1.3 BRIDGE21 CS CPD PROGRAMME 6

1.4 EVALUATION THEORY AND CPD EVALUATION 8

1.5 METHODOLOGICAL APPROACH 12

1.6 ANALYTICAL APPROACH 15

1.7 SUMMARY FINDINGS 17

1.8 ROAD MAP 19

2 LITERATURE REVIEW 21

2.1 LITERATURE REVIEW METHODOLOGY 21

2.2 EDUCATIONAL TECHNOLOGY CONTEXT 25

2.3 LEARNING AND TEACHING PROGRAMMING 34

2.4 PROFESSIONAL DEVELOPMENT IN STEM AND CS 45

2.5 PROGRAMME EVALUATION DESIGN AND METHODS 55

2.6 LIMITATIONS 68

3 BRIDGE21 CS CPD CONTEXT 69

3.1 COMPUTING AND THE IRISH CONTEXT 69

3.2 BRIDGE21 71

3.3 TCD POST GRADUATE CERTIFICATE IN 21ST CENTURY TEACHING AND LEARNING 77

3.4 SUMMARY 83

4 METHODOLOGICAL APPROACH 84

4.1 PHILOSOPHICAL FRAMEWORK 84

4.2 RESEARCH QUESTIONS 88

4.3 METHODOLOGICAL APPROACH 88
4.4 APPLYING YIN'S (2003) CASE STUDY PROTOCOL 92
4.5 APPLYING YIN'S (2003) DATA COLLECTION PRINCIPLES 103
4.6 APPLYING YIN'S (2003) CASE STUDY VALIDITY FRAMEWORK 109
4.7 LIMITATIONS..113

5 ANALYTICAL FRAMEWORK 115

5.1 THEORIES FOR ANALYSING DATA 115
5.2 ANALYSING MIXED METHODS DATA SETS 118
5.3 USING LOGICAL MODELS TO GUIDE ANALYSIS 120
5.4 STATISTICS AND ANALYSIS OF QUANTITATIVE DATA 122
5.5 THE ROLE OF CODING IN QUALITATIVE DATA ANALYSIS 124
5.6 MAPPING EACH OF KIRKPATRICK LEVELS TO THE DATA ANALYSIS MODEL 127
5.7 ALGORITHMS USED TO PROCESS QUANTITATIVE AND QUALITATIVE DATA SETS 129
5.8 SUMMARY..133

6 FINDINGS 135

6.1 UNITS OF DATA ANALYSIS 135
6.2 TEACHERS' REACTIONS TO THE CPD WORKSHOP CONTENT 137
6.3 CONTENT KNOWLEDGE DEVELOPED BY TEACHERS 153
6.4 STRATEGIES TEACHERS INTEND USING FOR TEACHING COMPUTING 176
6.5 TEACHING COMPUTING IN SCHOOLS 186
6.6 ELEMENTS OF THE BRIDGE21 MODEL USED FOR TEACHING COMPUTING 189
6.7 LIMITATIONS WITH IMPLEMENTING BRIDGE21 208
6.8 SUMMARY..209

7 DISCUSSION 211

7.1 TEACHING CS IN A 21ST CENTURY CONTEXT 211

7.2 BRIDGE21 CS CPD PROGRAMME AIMS AND OBJECTIVES 215

7.3 REVISITING THE RESEARCH QUESTIONS 216

7.4 TEACHERS' PERCEPTIONS OF THE BRIDGE21 MODEL 216

7.5 TEACHER EXPERIENCES OF USING THE BRIDGE21 MODEL 227

7.6 RESEARCH LIMITATIONS 233

7.7 RESEARCH CONCLUSIONS 234

7.8 RESEARCH CONTRIBUTIONS 236

7.9 AREAS FOR FURTHER WORK 237

8 REFERENCES 238

9 APPENDIX 294

9.1 CS CPD PARTICIPANT CONSENT SHEET 294

9.2 CS CPD PARTICIPANT INFORMATION SHEET 295

9.3 CS CPD PARTICIPANT ATTENDANCE, ETHICS, AND DATA SET 296

9.4 CS CPD REACTION VARIABLES ADAPTED FROM KRISTIANSEN (2007) 298

9.5 CS CPD INTENTION VARIABLES ADAPTED FROM KRISTIANSEN, (2007) 300

9.6 CS CPD LEARNING OUTCOME VARIABLES ADAPTED FROM TCD (2017) 300

9.7 CS CPD LEARNING PRE OUTCOMES QUESTIONNAIRE EXAMPLE 1 304

9.8 CS CPD LEARNING POST OUTCOMES QUESTIONNAIRE EXAMPLE 2 305

9.9 CS CPD REACTIONS/INTENTIONS POST QUESTIONNAIRE EXAMPLE 3 306

9.10 CS CPD FIELD NOTE TEMPLATE ADAPTED FROM EMERSON (1995) 307

9.11 TEACHING COMPUTING IN SCHOOLS VARIABLES AND SCALES 308

9.12 TEACHING COMPUTING IN SCHOOLS QUESTIONNAIRE 311

9.13 QUALITATIVE DATA ANALYSIS CODING EXAMPLES 319

9.14 QUANTITATIVE DATA ANALYSIS STATISTICS TABLES 324

9.15 VISUAL LITERATURE MAPS 354

9.16 PUBLICATIONS 356

Abstract

Across Europe there is strong push to teach Computer Science (CS) in post-primary schools (Forbes & Messina, 2002). Sentance and Csizmadia (2017a) advise that CS is perceived as a difficult subject, and call for Continuing Professional Development (CPD) to empower teachers (both in terms of the subject content knowledge and pedagogical approaches) to teach CS. Students struggle with transferring computing concepts from one context to another (Pea, 1987), while teachers lack the specialist pedagogical content knowledge to teach the subject (M. Webb et al., 2017). Traditional CPD programmes are criticised for using didactic methods, which are perceived to limit the sharing of expertise and the development of content knowledge, which can be used in a practical context.

Bridge21 is a collaborative, project-based, technology-mediated pedagogical model designed to facilitate 21st Century learning experiences which nurture student autonomy (Lawlor, Marshall, & Tangney, 2016). The Bridge21 pedagogical model supports collaborative, technology-mediated, project-based learning, and is used in a number of post-primary schools across Ireland. Trinity College Dublin (TCD) provides CPD for teachers specialising in 21st Century Teaching and Learning, using the Bridge21 model to deliver Computational Thinking, Programming, and Hardware modules. The aim of the CPD is to equip teachers with the content knowledge and the practical expertise for teaching CS. The adaptation of the Bridge21 model as a CS CPD method provides an opportunity to explore what impact a 21st Century approach to professional learning (covering the combined elements of facilitation, teamwork, project-based and technology-mediated learning) plays in equipping teachers with the pedagogical content knowledge, confidence and expertise to teach computing.

The researcher used a mixed methods approach to data collection in both CPD and school contexts to understand the impact of a 21st Century approach to professional learning. Data was collected over a consecutive five-year period. The researcher designed two mixed method questionnaires to collect data from a self-selecting sample of *N* = 1,215 teachers attending Bridge21 CPD workshops and to collect data from a self-selecting sub-set of *N* = 385 CPD teachers involved in using the Bridge21 model to teach computing in schools. Each questionnaire was adapted from existing Kirkpatrick (2007) training programme evaluation instrumentation. The CPD instrument examined teacher reactions to the CPD; teacher perceptions of their learning and intentions to use elements learned in the CPD for teaching computing in schools. The analysis of field note data collected during CPD workshops added context to the quantitative results. The teaching computing in school instrument explored the actual use of the Bridge21 model for teaching computing.

Two research questions are addressed in this thesis. The first question explores what are teachers perceptions of the Bridge21 model as a method of CPD; with further questions investigating what are teachers' reactions to the CPD workshop content?; what content knowledge did teachers learn?; and what strategies did teachers intend using for teaching computing? A second research question examined what are teachers' experiences of using the Bridge21 model to teach computing; with a further question investigating what elements of the Bridge21 model did teachers identify as most relevant for teaching computing in practice?

The CPD results confirm that teacher perceptions of the CPD, involving the use of the Bridge21 model as a method for learning computing, were positive. Teachers' reactions to the workshop content were positive, with teachers self-reporting gains in content knowledge, and confidence in facilitating collaborative, project-based, technology-mediated activities. Teachers also reported that the CPD met their expectations and that they intended using what they had learned in teaching computing.

The school results confirm that teachers' experience of teaching computing following the Bridge21 model, led to an observed increase in student engagement in computing. Teachers also observed an increase in student autonomy, with students taking the lead in computing projects, assisting peers, and working together to share computing knowledge and expertise.

Three contributions emerged from the research. First, the research findings confirm that using the Bridge21 model in a CPD context played a core role in assisting teachers' master CS content knowledge and methods. Second, research evidence is provided which reports that teachers observed an increase in student engagement in computing through the use of the Bridge21 model. And third, that the adaptation of the Kirkpatrick (1994) training programme evaluation framework provided a structure for investigating teacher perceptions and experiences of 21st Century CS CPD.

1 Introduction

This chapter begins by describing the research context, with further analysis exploring the issues around teaching computing, both internationally and in Ireland. The research questions are then described and this is followed by a summary of the research contributions. An overview of the Bridge21 Computer Science (CS) Continuing Professional Development (CPD) programme follows which provides a summary of the research context. The methodological approach includes a description of the philosophical frameworks used to govern data collection. The analytical approach covers the algorithms that were designed to process quantitative and qualitative data sets. The final section provides a summary of the findings, with this chapter concluding with a road map to the topics covered in remaining chapters.

1.1 Research Context

There is a strong push, both nationally and internationally, to teach Computer Science (CS) in schools. Across the EU, member states are introducing computing into post-primary systems (Forbes & Messina, 2002) with conferences such as the European Computer Science Summit (2018) and databases covering European Commission Computer Science Projects (EC, 2018) providing exemplars in teaching and learning computing in a pan European context. The EC (2017) argues that computer programming, also referred to as 'Coding,' *"enhances creativity, teaches people to cooperate, to work together across physical and geographical boundaries and to communicate in a universal language"* (p. 1). In Ireland, work is underway to implement a new syllabus in Coding for primary schools (NCCA, 2018c) with a lower secondary short course in Coding already published and available for teaching (NCCA, 2014a) and an upper secondary advanced curricula in Computer Science published, which is being rolled out from September 2018 in forty post-primary schools (NCCA, 2018c). The rapid introduction of computing across primary and secondary schools in Ireland has generated the need for high quality professional development to assist teachers in their preparation to teach computing. In recognition of this need, the Irish Government's Action Plan for Education 2016-2019 recommends that success in teaching CS in Ireland is dependent on making *"time available for teachers to develop new learning methods with top-class professional support"* (DES, 2018a, pp. 3-4). Currently, both primary and post-primary in-service teachers across Ireland can avail of a number of CPD offerings to prepare for teaching these new computing courses. These options range from whole school training provided by Professional Development Service for Teachers (2017), as well as educational centre courses (ATECI, 2018), self-study options with professional organisations (ICS, 2017), online courses (JCT, 2017) and third level accredited certificate programmes (NUI Galway, 2018; NUI Maynooth, 2017; UCD, 2017; UL, 2017).

1.1.1 CPD for Teaching Computing

This research takes place against a backdrop of significant educational reform in Ireland. The recent push to teach CS, both nationally and internationally, has created the need for CPD which equips teachers with the content knowledge, subject expertise and the confidence to teach CS (Mishra & Henriksen, 2018). Traditional didactic methods, using lectures and textbooks, are perceived to limit teachers in their ability to help students understanding computing concepts and methods (Hazzan, Lapidot, & Ragonis, 2014). In response to this teachers are looking for new pedagogical approaches which utilise collaboration, projects and technology-mediated activities to engage students in computing and computational activities (Grover & Pea, 2018; Shah et al., 2013).

Grover and Pea (2018) remind us that the "*twenty-first century is arguably the century of computing*" (p. 20). Grover and Pea continue that "*collaboration and creativity, now seen as cross-cutting skills for the twenty-first century learner, are also viewed as CT (Computational Thinking) practices that often require a unique flavour in a CT context*" (2018, p. 34). Shah et al. (2013) add that teachers need to be empowered in using methods which create opportunities for students to gain confidence in collaborative, project-based learning experiences that they can use to express their creativity. Access to 'peer relationships' in the computer science classroom are stressed as important for helping students engage with computing and programming concepts (Shah et al., 2013, p. 265). This thesis provides results which suggest that a 21st century approach to CS CPD (which uses a collaborative, project-based approach to professional learning) empowers teachers to teach CS, with teachers reporting that 21st Century learning experiences have a positive impact on the teaching and student engagement.

CS is perceived as a difficult subject to teach hence there is a need for Continuing Professional Development (CPD) to empower teachers (both in terms of the subject content knowledge and pedagogical approaches) to teach CS (Bosse & Gerosa, 2017; Du Boulay, 1986; Milne & Rowe, 2002). Computing is also perceived as a difficult subject for students to learn, with Pea (1987) suggesting that one of the core problems that students encounter when learning computing for the first time is the issue of 'transfer' which means developing the knowledge, expertise and ability to apply computing concepts learned in one context, in a different context. Teachers may also lack specialist pedagogical content knowledge which they need to support students learning independently of the teacher (M. Webb et al., 2017). In response, Robins, Rountree, and Rountree (2003); Major, Kyriacou, and Brereton (2012); Ridgway and Passey (1991); Ben-Ari (1998); Voogt et al. (2015); and Sentance and Csizmadia

(2017a) agree that a collaborative, project-based and activity led approach to learning offers one way for teachers to help students explore the meaning of computing.

Traditional CPD programmes are criticised for using what are perceived as didactic or lecture based ‘chalk and talk’ methods which fail to give teachers opportunities to direct their own learning (Taylor, 2018). In a computing context, lectures and problem solving continue to play an important role in enabling teachers to cover content that students need to learn (Lister et al., 2007). However computing is also a practical subject, involving students in the development of projects and the design, testing and implementation of computing artefacts (Hazzan et al., 2014). Teaching computing not only requires a blend of methods; it requires an innovative approach which encourages collaboration, giving students’ the chance to demonstrate their conceptual understanding of computing content as well as the ability to implement computing tasks.

As a consequence of this, CPD programmes are emerging which aim to provide learning experiences designed to give teachers the opportunity to explore strategies which encourage collaboration, project-based, and student-centred learning (Hargreaves & O'Connor, 2018). Collaborative professional development programmes provide learning experiences which encourage teachers to share their content knowledge as well as their professional practices (Kennedy, 2011). Collaborative programmes in computing are built around problem solving tasks, and involve teachers working in teams to build computing artefacts, giving teachers practical experience in computing processes and programming languages (Sentance & Csizmadia, 2017a). A collaborative, project-based approach to learning computing aims to facilitate the sharing of ideas, with peers encouraged to share professional experiences, present concepts, and reflect on their learning (Walker, 2018).

Sentance and Csizmadia (2017a) advise that computing topics, including Computational Thinking and programming are difficult to teach, and highlight two professional challenges that impact teacher preparation. The first challenge relates to understanding what assistance teachers need to develop computing pedagogical content knowledge (M. Webb et al., 2017) and the second concerns helping teachers develop the confidence to use facilitation and collaborative, project-based methods for teaching computing (Caspersen, 2018).

Professional development services play an essential role in enabling professionals to consult with peers and explore strategies and approaches for enhancing subject teaching. To ensure that educational programmes meet their objectives, there is an opportunity to evaluate and use metrics within CPD programmes, enabling CPD providers to tailor their programmes to teacher’s needs. Guskey (2000); Cutts, Robertson, Donaldson, and O’Donnell

(2017); and Sentance, Humphreys, and Dorling (2014) add that it is essential that we evaluate professional development programmes to ensure that the pedagogical approaches that are used within CPD programmes support teachers make the changes that they want to make in their teaching and professional practice.

Research exploring the impact of a facilitator led, collaborative, project-based approach to learning reveals that "*working in small groups motivates the students to discuss solutions and learn from each other*" (Nørmark, Thomsen, & Torp, 2008, p. 241). Furthermore, Nørmark et al. (2008) observe that collaborative learning experiences provide an alternative to 'traditional' teaching experiences, by giving students the opportunity to direct their learning (Devenyi et al., 2018). In a Computer Science context, CS CPD programmes are incorporating collaborative, project-based methods to encourage teachers to their share experiences and to assist each other develop CS content knowledge for use in their teaching (e.g., Cutts et al., 2017; Decker, McGill, Ravitz, Snow, & Zarch, 2018; Ravitz, Stephenson, Parker, & Blazevski, 2017; Sentance et al., 2014).

Bridge21 is a team-based, technology-mediated learning pedagogical model designed to facilitate 21st Century learning experiences which aim to increase learner engagement and reduce learner dependency on the teacher (Lawlor et al., 2016). As part of a Certificate in 21st Century Teaching and Learning, Trinity College Dublin (TCD) provides CS CPD for in-service teachers, with the Bridge21 pedagogical model of 21st Century Teaching and Learning used in the delivery of Computational Thinking, Programming, and Hardware workshops. A typical Bridge21 workshop experience is designed to encourage collaborative working, where students in teams complete technology-mediated, project-based activities (Lawlor, Conneely, Oldham, Marshall, & Tangney, 2018). Bridge21 learning experiences also involve teachers switching role to a facilitator, who plays an active part in supporting students, working in teams, take collective responsibility for completing projects (Lawlor et al., 2016). The Bridge21 model provides teachers / facilitators with a sequence, which guides students through the design, implementation, and evaluation of their projects. The Bridge21 model has been adapted as a CS CPD method to meet the twin demands of preparing teachers with the content knowledge and methods to teach CS, and to give teachers practical experience in facilitating project-based, collaborative learning activities.

1.2 Research Questions

Sentance and Csizmadia (2017a) suggest that teachers need assistance in preparing to teach computing. The creation of a new primary school Coding curricula (NCCA, 2018c) in addition to a lower secondary short in Coding (NCCA, 2014a) and the roll out of an upper secondary advanced curricula in Computer Science (NCCA, 2018c) across Ireland, make it imperative to understand what supports teachers need to introduce and teach computing in schools. The adaptation of the Bridge21 model as a CS CPD method generates an opportunity to investigate what impact a collaborative, project-based, and student-centred approach to professional learning plays in equipping teachers with the content knowledge, confidence, and expertise to teach computing. Two research questions are explored in this dissertation. The first question explores teachers' perception of the CPD offering, including reactions to the content, teacher perceptions of their learning and intentions to use the content in teaching CS. The second question explores teachers' experiences of using what they learned in the form of the Bridge21 model and the CPD content to teach CS in schools (Table 1).

Table 1 Research Questions	
CPD Context	
Research Question	Sub Questions
Q 1: What are teachers' perceptions of the Bridge21 model as a method of CPD?	Q1.1: What are teachers' reactions to the CPD workshop content?
	Q1.2: What content knowledge did teachers learn?
	Q1.3 What strategies did teachers intend using for teaching computing?
School Context	
Research Question	Sub Questions
Q 2: What are teachers' experiences of using the Bridge21 model to teach computing?	Q2.1: What elements of the Bridge21 model did teachers identify as most relevant for teaching computing in practice?

1.2.1 Contributions

Three contributions emerged through answering the research questions. The first contribution emerged through constructing an evidence base from research conducted with a self-selecting sample of *N* = 1,215 in-service teachers over a five year period. The research results confirm that a the Bridge21 approach to professional learning acts as a method for equipping teachers with the computing content knowledge, the practical expertise, and the confidence to teach computing.

The second contribution emerged from research conducted with a self-selecting sub-sample of N = 385 teachers who attended the CPD on their experience in schools. The results provide evidence of teachers using the Bridge21 model elements of facilitation, collaborative learning, and contextualised learning tasks. The results also provide insights into barriers (lack of technical infrastructure and further assistance with lesson planning) as well as successes, with teachers self-reporting an observed increase in student engagement though applying the Bridge21 model.

Finally, a third contribution comes through adapting Kirkpatrick's (1994) work to create a theoretical framework to explore teacher reactions and perceptions of their learning and the impact that the CPD had on teaching computing in schools. Chapter 3 provides a theoretical overview of the Bridge21 CS CPD context and workshop design, with chapter 5 describing each of the instruments.

1.3 Bridge21 CS CPD Programme

Bridge21 is a social constructivist model of 21st Century Teaching and Learning developed in TCD (Lawlor, 2016), which is used by post-primary teachers to encourage collaborative, team-based, technology-mediated learning in their subject teaching (Lawlor et al., 2018). The Bridge21 pedagogical model contains eight elements (Table 2, p. 7), which combine to create collaborative, project-based, technology-mediated teaching and learning experiences. A Bridge21 learning experience seeks to encourage students to work collaboratively to complete tasks, with facilitators interacting with students, and asking questions, to nurture individual autonomy. Bridge21 learning experiences involve teachers switching role from that of leader, to the role of mentor and guide, with students encouraged to take individual responsibility for their learning (Lawlor et al., 2018).

The Bridge21 activity model (Table 3, p. 7) was developed in TCD, to structure the content of Bridge21 activities (Byrne, 2018). Design thinking theory (T. Brown & Wyatt, 2010) underpins the Bridge21 activity model. The activity model aims to develop critical thinking, problem solving, synthesis and lateral thinking skills, and content knowledge. The activity model also provides a seven step consecutive sequence covering the design, implementation, and evaluation of projects, with teachers acting as a facilitator / mentor, guiding students through each step of the process.

Table 2 Bridge21 Pedagogical Model (Lawlor et al., 2018)	
Eight Elements of the Bridge21 Pedagogical Model	
(1) Collaborative learning through teamwork.	(2) Skill development orientation.
(3) Social learning protocols are techniques designed to nurture self-confidence and develop individual autonomy, through encouraging sharing behaviours.	(4) Facilitator and/or Mentor(s) approach where teachers change role to guide, facilitate or mentor.
(5) Reflection to encourage individual and team reflection at the end of learning experiences.	(6) Learning space – teachers organise desks into groups, where possible, to facilitate teamwork.
(7) Project-based.	(8) Technology-mediated, where projects are accomplished through using technology.

Table 3 Bridge21 Activity Model (Byrne, Fisher, & Tangney, 2016)
Seven Step Consecutive Sequence of the Bridge21 Activity Model
(1) Set up phase, where games are used as 'icebreakers' to encourage bonding, networking.
(2) Warm-up activities, where divergent thinking exercises are completed to explore broad concepts.
(3) Investigation involves defining the research problem, research and refining a design.
(4) Project plan development including, assigning roles and tasks, project planning and schedules.
(5) Create phase with teams working on projects exploring the chosen topic.
(6) Presentation phase where teams present the outcome of their projects to peers.
(7) A reflection phase gives individuals and teams the opportunity to reflect on their learning.

The Bridge21 pedagogical and activity models combine to provide a structure that can assist teachers develop the confidence, content knowledge, and the practical expertise to teach computing. In this context, peer collaboration is perceived as crucial to successful learning, with Vygotsky's (1978) Zone of Proximal Development (ZPD) theory providing a lens through which to explore the role of collaboration in equipping teachers with the confidence, knowledge and practical expertise to teach computing. Vygotsky (1978) describes the ZPD as the "*distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers*" (1978, p. 87). Vygotsky's (1978) formulation of ZPD theory proposes that knowledge is socially constructed and is further strengthened through meaningful problem solving in collaboration with more knowledgeable peers. Bridge21 learning experiences are structured to support the social construction of knowledge through peer collaboration. The researcher draws from ZPD theory to explore the role of peer collaboration as a method for equipping teachers with the confidence, pedagogical content knowledge, and expertise to teach computing.

Trinity College Dublin (TCD) is using the Bridge21 pedagogical model in a CS CPD programme. This programme is designed for in-service primary and post-primary teachers planning to teach computing. Teachers can select computing workshop modules covering computing concepts including Computational Thinking, Programming, and Hardware. Each computing workshop uses the Bridge21 pedagogical model in its delivery, with teachers learning computing through working together to complete computing activities, collaborating on projects, and completing technology-mediated tasks supported by their peers (TCD, 2017).

1.4 Evaluation Theory and CPD Evaluation

Worthern (1968) argues that it is difficult to evaluate the impact that educational theories have on professional learning interventions in a CPD context. Stake (1983a) suggests that researchers should use models to help clarify the educational theory that is to be evaluated. Stufflebeam (1983) shares the view that models are essential for guiding evaluation designs that are used to evaluate the role that theory plays in social programmes. Cronback (1983a) further advises that researchers should use models to evaluate theory as it relates to programme performance, which is a view shared by Guskey (2000), who argues strongly for using models in CPD evaluations to create links between teacher experiences, teacher feedback and the development of programme outcomes.

The evaluation models proposed by Dick (1978); Gagné (1970); Bloom, Engelhart, Furst, Hill, and Krathwohl (1956); and Biggs (1999) involve the learner completing a series of tasks and the teacher evaluating learning outcomes. Harden (2002) defines learning outcomes as "*broad statements of what is achieved and assessed at the end of a course of study*" (p. 151). Evaluation models designed by Cronbach (1980), and Worthen and Sanders (1973) suggest using evaluations to explore the extent to which professional learning experiences transform practice. Further evaluation models created by Stake (1983b), Stufflebeam (1983) and Scriven (1991b) propose that evaluations provide an opportunity to explore the impact of CPD programmes on teacher professional practice over time. Guskey (2000) and Kirkpatrick (1994) provide evaluation models which explore links between CPD learning outcomes and 'on the job performance'. Having reviewed each of these models, the researcher shares the view proposed by Bernthal (1995), who argues that Kirkpatrick is distinct from other models in its simplicity and capacity to explore learning transfer from one context to another.

1.4.1 Adapting Kirkpatrick to Evaluate Bridge21 CS CPD Programme

Kirkpatrick (1994) provides a four level evaluation framework which links learner perception with their experience of applying training in context. Kirkpatrick (1994) describes his four level framework as "a *sequence of ways to evaluate programs. Each level is important. As you move from one level to the next, the process becomes more difficult and time consuming, but it also provides more valuable information. None of the levels should be bypassed simply to get to the level that the trainer considers the most important*" (p. 21). Table 4, p. 9 – 10, provides a description of the Kirkpatrick (1994) framework, and a definition of each level in the sequence in the order in which they are used. Implementing Kirkpatrick involves a mixed methods approach to data collection (Mosson et al., 2019; Wang & Chang, 2019), with Broad (1997) providing guidelines for exploring self-assessed or self-reported learning.

Table 4 Kirkpatrick (1994) Levels			
Training Environment Context			
Level	Description	Purpose	Kirkpatrick (1994) definition
1	Reactions	Participant reactions to the training.	"*Evaluation on this level measures how those who participate in the program react to it. I call it a measure of customer satisfaction*" (Kirkpatrick, 1994, p. 21).
2	Learning	Learning in terms of changes in attitudes, skills and knowledge.	Evaluation on this level explores the "*extent to which participants change attitudes, improve knowledge, and / or increase skill as a result of attending the programme*" (Kirkpatrick, 1994, p. 22).

Table 4 Kirkpatrick (1994) Levels continued			
Work Place Context			
Level	Description	Purpose	Kirkpatrick (1994) definition
3	Behaviours	Changes in behaviour as a result of attending the training.	Evaluation on this level explores "*the extent to which change in behaviour has occurred because the participant attending the training program*" (Kirkpatrick, 1994, p. 22).
4	Results	Impact of the training on job performance and outcomes.	"*Results can be defined as the final results that occurred because the participants attended the program*" (Kirkpatrick, 1994, p. 22).

Kirkpatrick's (1994) four level framework has been previously applied in a CPD context to explore teacher perceptions of their teaching performance (Naugle, Naugle, & Naugle, 2000), teacher perceptions of their learning outcomes (Coldwell & Simkins, 2011) and student perceptions of their learning outcomes (Lawless & Pellegrino, 2007). The Kirkpatrick (1994) level framework has also been used to evaluate CPD in General Medicine (Shen, Yufe, Saadatfard, Sockalingam, & Wiljer, 2017), Dentistry (Ratka-Krüger et al., 2018), Pharmacy (Kheir & Wilbur, 2018), and Surgery (Dort et al., 2018). Moreover, Smidt, Balandin, Sigafoos, and Reed (2009) argue that "*the Kirkpatrick model provides one technique for appraisal of the evidence for any reported training program and could be used to evaluate whether a training program is likely to meet the needs and requirements of both the organisation implementing the training and the staff who will participate*" (p. 266).

A perceived strength of Kirkpatrick (1994) is its ease of implementation, its ability to explore knowledge 'transfer 'and behavioural change and the impact of the training on practice as well as its capacity to link learning with performance. Broad (1997) further argues that Kirkpatrick provides the capacity to explore change in learning outcomes over time. The adaptation of the Bridge21 model as a CS CPD method generates an opportunity to adapt Kirkpatrick (1994) to explore what impact a collaborative approach to professional learning plays in equipping teachers with the content knowledge, the confidence and practical and professional expertise to teach computing over time.

The decision to adapt Kirkpatrick (1994) was based on the outcome of a SWOT analysis (Table 5, p. 11). SWOT analysis sought to determine perceived strengths and weaknesses of the model as well as highlight opportunities and perceived threats to implementation (Humphrey, 2005). Arguments for adapting the Kirkpatrick (1994) model rest with its perceived simplicity, with researchers able to adapt each level to organisational needs. Weaknesses include the difficulty in addressing causality between levels (Holton, 1996), accommodating summative and formative forms of assessment (Bates, 2004), and limited opportunities to adapt the model once implemented (Kaufman, Keller, & Watkins, 1996).

Table 5 SWOT Analysis applied to Kirkpatrick (1994)	
Strengths	What is "*good in the present*?" (Humphrey, 2005, p. 7)
	Guskey (2000) argues that a strength lies in its "*simplicity and practicality which has made it the foundation of training programme evaluations in business around the world*" (p. 55).
Weaknesses	What is "*bad in the present*?" (Humphrey, 2005, p. 7)
	Bates (2004) argues that a weakness is its inability "*to effectively address both the summative question (Was training effective?) and the formative question (How can training be modified in ways that increase its potential for effectiveness?)*" (p. 341).
Opportunity	What is "*good in the future*?" (Humphrey, 2005, p. 7)
	Coldwell and Simkins (2011) suggest that the model provides the capacity to explore 'knowledge transfer' from one context to another, with further 'levels' exploring the demonstration of learning in a practical and professional context.
Threats	What is "*bad in the future?"* (Humphrey, 2005, p. 7)
	Kaufman et al. (1996) caution that "*the threat, which flows from misuse, comes from the fear that performance data will be used for blaming and not for fixing or learning*" (p. 8).

The outcome of the SWOT analysis process revealed that Kirkpatrick (1994) provided a structure to explore links between satisfaction with professional learning experiences, self-reported perceptions of participant learning and perceived changes in workplace performance (Dahler-Larsen, 2001). Kirkpatrick also provides the capacity to explore participant experiences - accommodating the analysis of participants perceptions of new behaviours developed through attending the training as well as self-reported experiences of using the content in the work place (Naugle et al., 2000).

1.5 Methodological Approach

This section provides the theoretical approach underpinning the design of the research methodology, which will be applied to answer the research questions (Table 6). Crotty (1998) suggests that researchers consider using a framework to represent the choice making process used to formulate the research design. Crotty (1998) continues that "*the justification of our choice and particular use of methodology and methods is something that reaches into the assumptions about reality that we bring to our work. To ask about these assumptions is to ask about our theoretical perspective*" (p. 2). While criticised for forcing the researcher to commit early to making philosophical decisions (Barkway, 2001), such a framework proves useful for helping to clarify relationships between Epistemological Beliefs and the Theoretical Position, the Methodological Perspective, Research Methods, and Units of Data Analysis. What follows is an adaptation of Crotty's (1998) framework showing connections between the theories and methods applied in this thesis.

Table 6 Philosophical Approach adapted from Crotty (1998)

Description of how each is applied in this thesis.	
Epistemology	The researcher aligned with a subjectivist epistemology based on the assumption that knowledge is in part socially constructed (Berger & Luckmann, 1966).
Theoretical Position	The researchers' theoretical position is declared as interpretivist given that knowledge and understanding are shaped through social interaction (Mead, 1934).
Methodology	The researcher choose an embedded case study methodology (Yin, 2003) with Walsham (1995b) providing guidelines from the interpretivist paradigm.
Methods	The researcher used a mixed methods design (Creswell, 2005) with a hypothetico-deductive logical model guiding data analysis (LeCompte & Schensul, 1999).
Units of Data Analysis	Kirkpatrick's (1994) training programme evaluation framework was adapted as a logical model, with each of the four levels adapted as units of data analysis.

The philosophical approach informs the selection of the research methodology (Table 7). Having declared as an interpretivist, what follows is a summary of the justification for the use of an embedded case study methodology (Yin, 2003) applied from an interpretivist position (Walsham, 1995b). The rational for selecting this philosophical approach and associated methods is provided in section 4.1. An embedded case study methodology and underlying design proposed by Yin (2003) provided an overarching framework enabling the researcher to link learning theory to the educational context, the context with the case, and the case to units of data analysis.

Table 7 Methodological Approach Adapted from Yin (2003)	
Construct	Description
Learning Theory	The theory under investigation is peer collaboration in a 21st Century context, as expressed by Vygotsky (1978) as ZPD theory and applied in CPD and school contexts through the Bridge21 model.
Educational Context	The introduction of computing into schools in Ireland.
Embedded Case	The case refers to a cohort of professional teachers involved in attending Bridge21 computing CPD workshops.
Units of Data Analysis	The units of data analysis are mapped to the four constructs, of reactions, learning, behaviours and results, adapted from Kirkpatrick (1994).

1.5.1 Mixed Methods

Having set out the parameters of an embedded case study methodology, this section makes the case for using a mixed methods approach to data collection. A mixed methods approach to data collection facilitates the synthesis of mixed methods data sets (Creswell, 2005), with statistical results providing indicators which inform the development of qualitative instruments seeking depth (Ivankova, Creswell, & Stick, 2006). The researcher created two mixed method questionnaires to collect data from teachers attending the Bridge21 CS CPD workshops, and to collect data from teachers using the Bridge21 model to teach computing in schools. The inclusion of further contextual evidence from the analysis of N = 109 pages of field notes collected during the CPD workshops adds context to the CPD results (Appendix 9.10). Table 8, p. 14 provides the mapping used to link the research questions to each of the Kirkpatrick Levels, which were adapted as units of data analysis.

Table 8 Kirkpatrick (1994) Adapted for Bridge21 CS CPD Evaluation			
CPD Context			
Level	Description	Research Questions	Unit of Data Analysis
Q 1: What are teachers' perceptions of the Bridge21 model as a method of CPD?			
Mixed Methods Questionnaire 1			
Level 1	Reactions	Q1.1 What are teachers' reactions to the CPD workshop content?	This unit explores teachers' reactions to the computing workshops.
Level 2	Learning	Q1.2 What content knowledge did teachers learn?	This unit relates to knowledge developed by teachers.
Level 3	Behaviours	Q1.3 What strategies did teachers intend using for teaching computing?	This unit refers to strategies teachers intended using for teaching computing.
Field Notes			
Context		Quotes, observations, and artefact descriptions.	Teacher engagement with the CPD.
School Context			
Level	Description	Research Questions	Unit of Data Analysis
Q 2: What are teachers' experiences of using the Bridge21 to teach computing?			
Mixed Methods Questionnaire 2			
Level 4	Results	Q2.1 What elements of the Bridge21 model did teachers identify as most relevant for teaching computing in practice?	This unit explores the results in terms of teacher attitudes to using the Bridge21 model for teaching computing, and teachers' self-reported use of the model with students.

Each questionnaire was adapted from existing Kirkpatrick training programme instrumentation (Kirkpatrick, 2007). A self-selecting sample of *N* = 1,215 teachers attending *N* = 72 the CPD between January 2014 to June 2018, completed a pre and post-workshop questionnaires. A total of *N* = 293 teachers completed the pre-workshop questionnaire, with *N*

= 819 teachers completing a corresponding post-workshop questionnaire, generating *N* = 1,112 completions, achieving a workshop response rate of 92%. A further self-selecting sample of *N* = 385 teachers who attended at least one computing workshop between January 2014 and June 2018, and had given consent to be contacted, were invited to complete a follow-up questionnaire exploring teaching computing in schools. Questionnaire data from *N* = 64 teachers was processed, generating a 17% response rate.

1.5.2 CS CPD Workshop Questionnaire Variables

The first mixed methods questionnaire was completed by teachers at the start and at the end of each CPD workshop. The questionnaire contained Likert scales exploring demographics, and teacher reactions to the workshop content, teacher perception of the learning outcomes, and teacher intentions to use elements of the Bridge21 model. A further qualitative section asked teachers to report on key learnings taken from the workshop, perceived use of the workshop content in teaching, and intended changes to practice as a result of attending the CPD. Appendix 9.4, 9.5 and 9.6 provide the variable tables outlining question design and mapping to Kirkpatrick (1994).

1.5.3 Teaching Computing in Schools Questionnaire Variables

The second mixed methods questionnaire was completed by teachers, ranging between six months to five years since last attending a CPD computing workshop. Part one of the questionnaire contained ethics, background, and consent information as well as demographics questions, covering primary/secondary teachers attending the CPD and computing within the curriculum. The first section of the questionnaire explored Bridge21 elements used for teaching CS with subsequent sections exploring perceived barriers to using the Bridge21 model; other methods used for teaching CS, and areas of the Bridge21 model requiring further CPD. A second section explored teaching computing in schools covering examples from teachers reporting on use of the Bridge21 model and its impact on student engagement in a computing context. Appendix 9.11 provides the variable tables outlining the scales and question mapping to the Kirkpatrick (1994) model.

1.6 Analytical Approach

Having explored the methods used in data collection, the following section explores the process used to analyse quantitative and qualitative data sets (Schwandt, 2000). Hypothetico-deductive analytical models give interpretive researchers the flexibility to revisit, question, reconstruct, deconstruct and reassemble ideas over the duration of data analysis (Denzin, 1989). However, care must be taken not to 'over theorise' or to reduce the data to variables removed from their original context (Blumer, 1969). Hypothetico-deductive analytical models start from a position of theory (Lincoln & Guba, 1985) which in this case is the use of the Bridge21 model as a CS CPD method.

1.6.1 Processing Quantitative then Qualitative Data

LeCompte and Schensul (1999) suggest identifying categories to be addressed through analysis. Kirkpatrick's (1994) training programme evaluation framework levels were adapted as categories or units of data analysis. Having established categories to organise data collection, LeCompte and Schensul (1999) then suggest establishing procedures to process quantitative and qualitative data sets. In the case of quantitative results, the researcher calculated the sum, mean and percentages values from Likert scaled results (Oppenheim, 2000). Percentage values are given as representations or indicators which are used to gauge positive and negative responses against each Likert variable (Joshi, Kale, Chandel, & Pal, 2015). Cronbach (1951) Alpha Coefficient values were calculated for each Likert scale to provide an estimate of the reliability or consistency of scale items. Further statistical analysis using the Wilcoxon (1945) signed rank text was performed on pair matched pre and post learning outcome results to explore the 'significance' of the difference reported between medians, pre and post, and therefore confirming an increase or decrease between learning scores.

1.6.2 Comparative (Pattern) Coding of Quantitative Data

LeCompte and Schensul (1999) recommend using a comparative coding process known as pattern coding to reduce large qualitative data sets into meaningful clusters. Pattern coding is used to compare and merge concepts into themes. Having established initial themes, LeCompte and Schensul (2013) advise repeating coding within each theme (Stadler, 2004) to accommodate merging and to allow for the emergence of contrasting themes. Pattern coding enabled the researcher to add new codes, remove old codes, and merge existing codes given that "*codes will change and develop as field experience continues*" (Miles & Huberman, 1994, p. 61). The re-coding process also involved the re-categorisation of the data into new themes or for existing themes to be removed from the data set.

1.6.3 Validity Framework

Having defined the processes used to analyse the data, the researcher faces a further challenge in selecting a validity framework against which to authenticate the research findings (Lave, 1988). Yin's (2003) validity framework designed for case study research was used to verify the credibility of research results. Yin (2003) suggests building construct validity through establishing a chain of evidence. A chain of evidence was established through adapting

Kirkpatrick (1994), with levels mapped to the research questions, then units of data analysis mapped to each of the research instruments. Yin (2003) then suggests using logical models to establish internal validity.

The researcher followed a hypothetico-deductive approach to data analysis with percentages and themes used to demonstrate internal validity through providing the capacity for theoretical abstraction. Yin (2003) then recommends choosing a methodology, which can be replicated to establish external validity. The researcher applied Yin's (2003) embedded case study design as the methodology, and used Yin's (2003) case study protocol to structure reporting. Cronbach (1951) Alpha Coefficient values were calculated for each Likert scaled items which provides an estimate of the reliability and consistency of a scale, with pattern coding used to generate themes from text analysis (LeCompte & Schensul, 1999). The Wilcoxon (1945) signed rank test used to verify the statistical significance of the change in median values reported between pre and post-workshop learning outcome results.

1.7 Summary Findings

The adaptation of the Bridge21 model as a CS CPD method generates an opportunity to understand what impact a 21st Century approach to CS CPD (covering facilitator led, collaborative, project-based and technology-mediated learning) plays in equipping teachers with the content knowledge, confidence, and expertise to teach computing. Two research questions were designed to examine teacher perceptions and experiences of collaborative CS CPD (Table 1, p. 5).

The Bridge21 CPD workshops brought teachers together to learn computing and to develop practical expertise in planning programming activities for use in teaching. Encouraging teachers to learn as part of a team, motivated teachers to seek assistance from peers to clarify concepts or to demonstrate procedures, with facilitators on hand to moderate the social exchange of ideas, practices, and concepts and to ensure that teachers obtained answers to their questions. Working in a team also created an opportunity for teachers to share prior content knowledge as well as to ask for help from colleagues where content knowledge gaps emerged, with teachers also reacting positively to the workshops and leaving with a practical understanding of computing concepts.

The CPD results exploring teacher perceptions, reactions, learning and intentions demonstrate that teachers enjoyed the experience of learning computing content in a peer supported environment as well as the freedom to try out and explore computing concepts through practical work. Teachers enjoyed discussing ideas and concepts with colleagues and exploring how to adapt the CPD content to a classroom context. However, teachers also revealed needing more time to develop deeper content knowledge and more time to develop CS activities, with further research needed to explore the longer-term impact of using the Bridge21 model in schools.

Teachers who used Bridge21 to teach CS in their classrooms reported students being more confident in learning together and taking control of their learning. Teacher attitudes towards teaching computing were also reported as positive, with teachers empowered to teach computing and motivated to share computing expertise with colleagues. Teachers provided examples demonstrating use of the Bridge21 model as a method for increasing student engagement, with teachers facilitating students in teams demonstrating projects and confidently discussing the outcome of computing projects with peer groups. However, teachers indicated the need for further assistance with evaluating, planning, and implementing computing learning experiences and future activities.

To conclude, these findings show that a facilitator driven, collaborative, technology-mediated and project-based approach to CS CPD supports the process of equipping teachers with the content knowledge, the technical expertise, and the confidence to teach CS. The CPD results confirm that the Bridge21 pedagogical and activity models provide a scaffolding structure capable of assisting teachers develop the confidence, the content knowledge, and the practical expertise to teach computing. Teachers enjoyed the freedom to learn with colleagues and the opportunity to explore how computing content and teaching methods covered in a CPD context can be applied in a classroom context. Recommendations include more CPD activities covering core programming concepts, expanding modules to cover advanced topics, and investing in building CS communities, which support teachers in a classroom, school, and national capacity.

The above findings combine to produce three research contributions. The first contribution emerged through constructing an evidence base from research conducted with a self-selecting sample of *N* = 1,215 teachers attending Bridge21 CPD computing workshops over a five year period (from January 2014 until June 2018). These results confirm that a Bridge21 approach to professional learning acts as a process which can help to equip teachers with content knowledge, and the confidence to teach computing using a collaborative, and project-based approach.

The second contribution emerged from research conducted with a self-selecting sub-sample of *N* = 385 primary and post-primary and teachers who attended the CPD and are now involved in teaching computing in schools. These results provide evidence of teachers reporting on changes in classroom practice, with teachers using the Bridge21 model elements of facilitation, collaborative learning, and contextualised learning tasks to engage students in computing. These results provide insights into barriers (lack of technical infrastructure, time to practice programming and further assistance with lesson planning) as well as successes, with teachers self-reporting an observed increase in student engagement though applying the Bridge21 model.

A third contribution comes through adapting the Kirkpatrick (1994) training programme evaluation framework as a methodology for exploring teacher perceptions and their experiences of a facilitated, collaborative, technology-mediated, project-based approach to professional development in a computing context. Adapting the Kirkpatrick framework provided a structure for investigating teacher perceptions and experiences of 21st Century CS CPD in schools in Ireland.

1.8 Road Map

The chapters that follow describe the steps followed to review the literature and construct research questions, design a research intervention, apply methods of analysis and produce research findings.

Chapter 2: Literature Review

This chapter contains the methodology supporting the development of the literature review as well as a discussion of four themes, which emerged through evaluation and analysis. The first theme explores use of technology in education, investigating the applied use of technology in STEM and CS teaching. The second theme explores learning and teaching programming, with analysis examining emerging strategies for teaching programming. The third theme examines the role of professional development and its application in STEM and CS contexts. The fourth and final theme examines programme evaluation theory design and methods, both applied in a STEM professional development context and within CS professional leaning programmes.

Chapter 3: Bridge21 CS CPD Context

This chapter provides an overview of the research context including a description of Trinity College Dublin's Post Graduate Programme in 21st Century Teaching and Learning (TCD, 2018), a summary of the Bridge21 model (Lawlor et al., 2018) and an outline of the Bridge21 activity model as it was applied in CS CPD computing workshop context (Byrne et al., 2016).

Chapter 4: Methodological Approach

This chapter summarises the philosophical approach structured according to Crotty's (1998) framework, which links theories of reality and knowledge to field work. An embedded case study methodology was used to structure fieldwork (Yin, 2003), with a mixed method design guiding data collection (Creswell, 2005) and Kirkpatrick's (1994) four level evaluation framework adapted as units of data analysis linked to research instruments. A description of the embedded case study methodology is provided, including Yin's (2003) protocols which were followed to collect data from CPD and school contexts. The remaining sections cover instrument design, data collection including ethics, and the validity framework applied to data.

Chapter 5: Analytical Framework

This chapter covers data analysis procedures and provides the logical model and the data processing techniques used to analyse quantitative and qualitative data sets. The first three sections cover theory and processes used to analyse quantitative and qualitative data sets. Having explored general theories, the particular strategies used for processing quantitative and qualitative data sets in this thesis are discussed, covering the mapping used to link instruments and data sets to the Kirkpatrick (1994) model as well as the algorithms that were used to process each of the data sets collected in CPD and school contexts.

Chapter 6: Findings

This chapter presents the research findings in full, placing them in context with the problem statement and organised under each of Kirkpatrick's (1994) four evaluation levels which were adapted as units of data analysis. The first part explores the research findings, which were gathered from self-selecting samples of teachers attending Bridge21 CS CPD computing workshops. The second part examines findings, which were gathered from self-selecting samples of teachers reporting on their experience of using the Bridge21 model to teach CS in a school context, with teachers providing examples of the models impact on student learning.

Chapter 7: Discussion

This chapter explores the findings in context, revisiting the research questions, the research contributions, recommendations for further research and limitations. The discussion is organised according to the research questions, and starts by revisiting the problem statement and rational for teaching computing. The aims and objectives of the Bridge21 CS CPD programme are then revisited, before the research findings are discussed in relation to the research questions. The research limitations and conclusions then follow. This chapter ends an overview of the research contributions and proposals for further research.

2 Literature Review

This literature review aims to understand what issues teachers face in preparing to teach computing. Sentance and Csizmadia (2017a) identify computing as a difficult topic to teach and highlight two professional challenges impacting teachers in their computing preparation. Thus the first challenge concerns understanding what assistance teachers need to develop computing content knowledge (M. Webb et al., 2017) and the second challenge relates to helping teachers develop the confidence to use relevant methods for teaching computing topics including programming (Caspersen, 2018). With teachers in Ireland facing the challenge of preparing to teach CS, this review implements a methodology designed to identify what supports teachers need to prepare for teaching CS.

This chapter provides a description of the process followed to develop the theoretical rational for designing research questions and methods used to explore teachers perceptions and experiences of a particular model of 21st Century CS CPD. The first section covers the methodology used to source and analyse literature included for evaluation and review. Four themes emerged through the literature evaluation process. The first theme explores the use of technology in an educational context and implications for teachers integrating technology into STEM and CS lessons. The second theme examines barriers to learning and teaching programming, and explores proposals for teaching strategies to make programming content and processes accessible to learners. The third theme explores the role of professional development and its function in preparing teachers to teach computing and programming content .The fourth theme examines programme evaluation theory, with the aim of identifying models that are used for evaluating CDP programmes in computing.

2.1 Literature Review Methodology

A literature review presents an analysis of research in order to extend our understanding of a particular topic. It involves analysing theories to reveal a lack of theory or to explain the use of theory in a different context (Booth, Sutton, & Papaioannou, 2016). Literature review methodologies help the researcher build a strong theoretical base for designing research (Rowe, 2014). It is against the content of the literature review that the researcher confirms or rejects theories based on their relevance (Littell, Corcoran, & Pillai, 2008). Thus care must be taken to select a literature review methodology which helps the researcher develop their theoretical claims (Cooper, 2010).

According to Grant and Booth (2009) there are at least fourteen methodologies that the researcher can use. Each methodology can be sorted into one of three categories. The first category uses statistical frameworks to compare and aggregate quantitative results (Paré, Trudel, Jaana, & Kitsiou, 2015). The second category uses thematic frameworks to code and cluster related topics into general themes (Onwuegbuzie & Weinbaum, 2017). The third category combines statistical and thematic analysis (Arrowsmith, Lau-Walker, Norman, & Maben, 2016). Each methodology aims to help the researcher make sense of concepts through a systematic process, with meta-synthesis providing a framework to explore statistical and thematic content (Whittemore & Knafl, 2005).

What differentiates meta-synthesis from other literature review methodologies is the capacity to combine qualitative and quantitative results (Coorey, Neubeck, Mulley, & Redfern, 2018). The meta-synthesis process involves using comparison to merge concepts into themes (Noblit & Hare, 1988). Originally established as a technique for generating theory (Phillips, Koehler, Rosenberg, & Zunica, 2017), Stern and Harris (1985) adapted meta-synthesis as a process for developing explanatory models (Teague & Roe, 2008). Examples include using meta-synthesis to generate metaphors (Tom, 2015) and themes (Melcer & Isbister, 2018) as well as new theoretical models (Melcer & Isbister, 2018) with Tondeur et al. (2012) arguing for expanding qualitative meta-synthesis to include statistical analysis. The researcher adapted a meta-synthesis methodology based on the rational provided by Tondeur et al. (2012) and Coorey et al. (2018) who call for analysing statistics and themes to provide a balanced view.

2.1.1 Meta-Synthesis Methodology

The rational for choosing a meta-synthesis methodology was based on the need to use a framework capable of merging quantitative and qualitative results. Traditionally, literature reviews are constructed to analyse qualitative or quantitative results, following particular procedures, which govern the syntheses of themes or the aggregation of statistical results. Given the requirement to explore themes and statistics together, the researcher adapted an existing literature review methodological approach (qualitative meta-synthesis) which provides the capacity to facilitate the retrieval, evaluation, and the 'meta' synthesis of qualitative themes and quantitative results.

Having defined the research problem as the need to explore what supports teachers need to prepare for teaching CS, the next step involved selecting a meta-synthesis framework to guide the searching and analysis of concepts emerging from the literature. The researcher adopted the four stage approach developed by Tondeur et al. (2012) who adapted the meta-synthesis methodology to explore concepts examining the instructional use of technology in a teaching and learning context. The first phase of the process involved designing a search strategy and then developing criteria to retrieve research papers. The second phase covered literature screening and evaluation procedures applied to retrieved content. The third phase involved meta-synthesis, consisting of coding and theming retrieved results, and the fourth involved the construction of a conceptual model.

The first step involved developing a search strategy (Table 9). The search strategy was constructed through deconstructing keywords in the problem statement into Boolean operators (AND, OR, NOT). Boolean terms were entered across educational and computing databases (TCD online library, ERIC, ACM, IEEE, Sage Journals Online, Science Direct, Taylor and Francis, and JSTOR), with saved searches and alerts activated to capture new publications.

Table 9 Search Strategy Applied to Databases
Educational (AND) Technology (AND) Information, Communication, Technologies (AND), In-Service, Post-Primary, Teaching, 21st Century Teaching and Learning; (AND) Teaching (AND) Learning (AND) Computer Science (AND) Programming; (AND) Collaboration (AND) Continuing Professional Development (AND) Programme Evaluation (AND) Theory (AND) CPD Evaluation.

The researcher followed what Bujaki and Richardson (1997) call a 'citation trail', with search results returning initial citations linking to further citations. Literature from peer reviewed journals and conference papers as well as monographs in English were included, with non-peer reviewed literature excluded from further review. The search parameters for full text papers was set to capture citations published between 2000 and 2018, with the timeline extended to 1900 for monographs. Chen (2017) confirms that a twenty-year period provides suitable scope for exploring changes in theory over time. The search strategy returned *N* = 1,973 items. Cooper (2010) defines literature reviews as providing exhaustive, exhaustive with selective citation, representative, central or pivotal reports. The researcher defines this review as providing 'representative' coverage of computing education themes, which includes a 'sample that typifies larger groups of articles' (Cooper, 2010, p. 110).

The second phase involved applying abstract screening to retrieved content. Abstract screening helps determine paper relevance and screening provides a filter through which to accept or reject papers based on their theoretical content (Papadakis, 2018). The researcher adapted abstract screening process used by Houghton et al. (2017) as the methodology to select articles for inclusion in the analysis (for further examples of this methodology see Lundorff, Holmgren, Zachariae, Farver-Vestergaard, & O'Connor, 2017; Paras, Pal, & Ekwall, 2017). The abstract screening process proposed by Houghton et al. (2017) involved reading the title and abstract of each retrieved paper as well as noting key authors and prominent theories. The researcher followed this process for five years, corresponding with time spent in the field. This process generated *N* = 1,324 items.

The researcher then adapted evaluation criteria used by Atkins et al. (2008) to explore theoretical links between retrieved papers. Using the markers of '*Yes*', '*No*' or '*Unclear*', helped the researcher further reduce the data set to *N* = 612 items, with the researcher then using visual mapping software to cluster papers sharing the same theoretical approach into groupings. Appendix 9.15 provides a series of visual maps which show themes which emerged from analysing the literature. These themes became headings which were used to structure the literature review. Acknowledging bias in the coding and theming process (Coombs, 1994), the researcher re-evaluated papers during literature searching, with more relevant papers replacing less relevant papers (Lincoln, 2002). Concept mapping also enabled the researcher to create a visual representation of themes, which emerged through analysis (Appendix 9.15).

Four themes emerged through using the meta-synthesis methodology (Table 10). The first theme examined the educational use of technology and strategies for using technology to enrich STEM and CS teaching. The second theme explored perceived barriers to learning and teaching computing, and strategies for teaching programming. The third theme examined professional development in general and in a STEM and CS context. The fourth theme investigated evaluation theory, covering frameworks used in CPD programme evaluation, including STEM and CS CPD. These themes formed the structure of the literature review.

Table 10 Meta-Synthesis Model and Procedure

Section	Sub-Themes	Section	Themes
2.2.1	Technology Enhanced STEM Teaching	2.2	Theme 1 - Educational Technology Context
2.2.2	Factors Influencing Teaching Computer Science		
2.3.1	Current Strategies for Teaching Programming	2.3	Theme 2 - Learning and Teaching Programming
2.3.2	Emerging Strategies for Teaching Programming		
2.4.1	STEM Professional Development	2.4	Theme 3 - Professional Development in STEM and CS
2.4.2	Computer Science Professional Development		
2.5.1	Evaluating STEM Professional Development	2.5	Theme 4 - Programme Evaluation Design and Methods
2.5.2	Computer Science Professional Development Evaluation		

2.1.2 Limitations

Structuring the literature review using meta-synthesis, helped the researcher to obtain good understanding of broad issues related to the problem area (Cooper, 2010). Using this methodology, the researcher was able to explore broad concepts in the quantitative and qualitative literature, before proceeding to take a more in-depth look at specific issues. This approach proved crucial in so far that the researcher had qualifications in librarianship, information analysis, and technology enhanced learning as well as twenty-five years professional experience as a technology analysist, but lacked the practical experience of teaching computing in schools. In summary, a 'meta-synthesis' approach to coding and theming concepts in the literature, familiarised the researcher with domain-based issues before starting work on the design of the study methodology (Noblit & Hare, 1988).

2.2 Educational Technology Context

Teaching in the 21st Century is perceived as exciting and challenging. Teachers face the initial challenge of integrating technology in lessons and the subsequent challenge of using methods which involve students taking more responsibility for their learning (Henriksen & Mishra, 2018). Significant developments in learning theory have been made since Skinner (1968) initially experimented with using technology to assist students learn procedural knowledge and further proposed using computers as teacher replacements for presenting content to students (Skinner, 1961). A new paradox has emerged, with teachers now spending more time supervising students using technology, while teachers are struggling to maintain the balance between teaching, and using technology to enhance student learning (Comi, Argentin, Gui, Origo, & Pagani, 2017).

Advancements in educational technologies have created new opportunities for teachers to design learning activities, which involve students taking more responsibility for their learning (Admiraal et al., 2017; Avidov-Ungar & Forkosh-Baruch, 2018). One example can be seen through teachers using projects to encourage students to use technology to bring together information from different sources around a theme (Atherton, 2018). Integrating technology into learning activities is complex (i.e., Boulton, 2017; J. Harris, Phillips, Koehler, & Rosenberg, 2017; Kale, 2018) and one of the difficulties teachers may encounter is a lack of confidence (Pareja Roblin et al., 2018).

An advantage of integrating technology into learning experiences, is that it enables teachers to create tasks which encourage students to make decisions about what they want to learn (Knowles, 1988, p. 18). Some examples include using technology to complete online research, game play, and puzzles with each task giving students the opportunity to practice making choices (B. Kim, Park, & Baek, 2009). Thus teachers are looking for help to integrate technology into teaching in ways which enable students to lead in learning activities (Esteve-Faubel, Martin, & Junda, 2018). Teachers are seeking help to design tasks, which enable students to master skills, which enable them to learn independently, and exercise choice making (Angeli, 2013; Laurillard, 2013; Little, 1995).

There are a number of educational frameworks, which can help teachers integrate technology into teaching. Some of the most well-known are 21st Century Learning frameworks such as the SAMR model (Puentedura, 2006), the P21 Framework (2017) or the UNESCO ICT Competency Framework for Teachers (2011). The SAMR model provides guidelines for teachers seeking to enhance teaching through using technology-mediated tasks (Puentedura, 2006). While the P21 Framework (2017) includes lesson plans for teachers intending to develop activities which involve students using technology to complete tasks such a searching for information or creating technology artefacts. The UNESCO ICT Competency Framework for Teachers (2011) provides overarching policy direction for governments and their educational systems seeking to integrate technology into teaching and learning. Both the SAMR, and P21 Framework in particular, as well as the other 21st Century learning frameworks evaluated by Dede (2010) support the integration of technology into teaching and provide teachers with a way to combine facilitation, projects and teamwork into lessons, with the aim of equipping students with the skills and confidence to learn independently.

Over the past forty years, the educational use of technology has evolved considerably (Mishra & Henriksen, 2018). For example, students are perceived as confident users of digital technologies, with students using mobile technologies and internet resources to communicate with peers and interact with a technology-mediated world (Zawacki-Richter & Latchem, 2018). However, in contrast to this, teachers are struggling with integrating technology into their teaching (Henriksen, Richardson, & Mehta, 2017). In other work contexts, the integration of technology into everyday jobs ranges from improving productivity to disrupting the traditional paradigm (Bevan, Brinkley, Cooper, & Bajorek, 2018). However the same cannot be said for teaching, with Gil-Flores, Rodríguez-Santero, and Torres-Gordillo (2017) asking why are teachers still struggling with integrating technology into their teaching and Olofsson, Lindberg, and Fransson (2017) calling for more research to understand what supports teachers need to become confident users of technology and empowered in the classroom use of digital tools.

Successful technology implementations are dependent on teachers who are confident in their use of technology and are strong in their belief that technology adds value to teaching (Pareja Roblin et al., 2018). Research examining teachers' beliefs and the educational use of technology is not a new phenomenon (N. Davis et al., 1997; Duval, Sharples, & Sutherland, 2017; Passey, 2006). Twenty years ago (1997) predicted that teachers would use technology to reduce the time that was spent presenting content to their students through *"additional or alternative source of knowledge and information"* (p. 15). Ten years later Passey (2006) cautions that there is a "*clear need for teachers to know how each form of ICT (Information Communication Technology) supports precise aspects of learning in each subject area, topic and activity*" (p. 139). Today teachers are still exploring how to integrate technology into teaching and are still deciding how to maximise technology for instruction.

Tondeur, van Braak, Ertmer, and Ottenbreit-Leftwich (2017) acknowledge that "*achieving technology integration is still a complex process of educational change*" (p. 555). For example, providing students access to technology gives students the opportunity to learn through sharing resources and collaborating with peers (Sergis, Sampson, & Pelliccione, 2018). However, facilitating students using technology increases the amount of time that teachers spend on checking the accuracy, and the credibility of sources (Hatlevik & Hatlevik, 2018).

One technology integration framework, which is designed to help teachers understand the use of technology in their subject teaching, is TPACK (Technological, Pedagogical, Content Knowledge). TPACK is a framework which seeks to align teachers content knowledge, knowledge of methodological practices used in subject teaching and technical expertise (Koehler & Mishra, 2009). The process of transforming subject knowledge into 'rich forms of instruction' is complex and occurs as a result of combining content knowledge, teaching methods and expertise which is developed through practice (De Miranda, 2018). The TPACK framework provides one way to explore the integration of technology into teaching (C. J. Lee & Kim, 2017). However, much has changed since Mishra and Koehler (2006) developed TPACK to explore the "*phenomenon of teachers integrating technology into their pedagogy*" (p. 1017).

Finally, Richardson and Mishra (2018) argue that more research is needed to explore what impact technology-mediated learning has on teaching, so that supports can be put in place to help teachers prepare for future changes in the use of technology in a classroom context. Further recommendations are proposed by Phillips et al. (2017) who seek further clarity of the term 'teacher content knowledge,' while Koehler, Greenhalgh, Rosenberg, and Keenan (2017) remain concerned at the lack of frameworks that are available to help teachers to enhance their 'technology knowledge.' It is argued that 21st Century learning frameworks provide an alternative to textbook and lecture led teaching (Fischer et al., 2018). Indeed, 21st Century learning frameworks with their emphasis on skills based learning, help students develop the confidence to direct their learning (Dede, 2010). Examples of these skills include ‘critical thinking,’ ‘communication,’ ‘collaboration,’ and ‘creativity’ in addition to ‘life and career skills,’ and ‘information, media, and technology skills’ (P21, 2017).

To conclude, Robinson (2017) states, “*we are living in a world that is changing faster than ever and facing challenges that are unprecedented. How the complexities of the future will play out in practice is all but unknowable*” (p. 1). Thus, it is imperative that teachers are provided with frameworks, which enable them to develop the potential of technology in their teaching. The increase in use of digital tools and applications across society and within education makes it possible for teachers to become more involved in designing learning experiences, which involve students using technology to direct their learning. However integrating technology into teaching is complex, with teachers facing the challenge of developing content knowledge as well as pedagogical methods that are suitable for technology-mediated experiences. Integrating technology into teaching is underpinned by educational frameworks, which view technology-mediated learning experiences as essential for enabling students to develop key skills for future learning and working. However, teachers face significant challenges in moving from textbook and lecture based teaching methods, to a 21st Century approach that involves students learning problem solving, critical thinking, and communication skills as well as mastering technology. Changing methods to support students learning with technology is complex; however, the benefit to be gained from moving to a collaborative, technology-mediated teaching approach is the potential to engage students.

2.2.1 Technology Enhanced STEM Teaching

The demand for students with science and technology expertise across western society is increasing (Sauberschwarz & Weiss, 2018). In consequence, students are being encouraged to study Science Technology Engineering and Mathematics (STEM) subjects (Miller, Sonnert, & Sadler, 2018). As the demand for students with STEM skills grows, Barak and Assal (2018) call for research to identify what content knowledge, and professional expertise, as well as what technical skills teachers need to teach STEM. Horvath, Goodell, and Kosteas (2018) argue that more needs to be done to understand the teachers’ experience; with Thibaut, Knipprath, Dehaene, and Depaepe (2018) calling for research to uncover what methods are used in teaching STEM, and Schuck, Aubusson, Burden, and Brindley (2018) seeking evidence of examples which are used to help students understand core concepts.

There are a number of strategies, which teachers can use to teach STEM. At the core of a STEM learning experience is the need to help students develop a practical knowledge of science (Galili, 2018). In contrast to both Popper (1972), and Kuhn’s (1970) view of scientific knowledge, Latour (2017) insists that scientific knowledge is neither procedural nor objective, but "*is more complex and messy*" (p. 1). Thus, according to Latour, STEM teachers should be encouraged to adopt strategies, which offer a more 'creative' approach to teaching. Jonassen (1999) suggests using strategies which encourage critical thinking (Jonassen, 2000), problem solving (Jonassen, Howland, Moore, & Marra, 2002), and decision making (Howland, Jonassen, & Marra, 2014). Latour further argues that critical thinking and problem solving tasks offer the potential "*to present science as science in action"* (2017, p. 1), making science relevant and contextual.

There are a number of ways that STEM teachers can use technology to enhance their teaching. Atherton (2018) suggests no less than fifty ways that teachers can enhance learning using technology, including online game play, virtual reality, interactive assessment, social media , audio-visual simulations, and collaborative and groupware tools as well as data analytics. Thus, STEM teachers face a twofold challenge in designing STEM tasks. First, STEM teachers face the challenge of designing learning experiences which are "*active, constructive, cooperative, authentic and intentional*" (Howland et al., 2014, p. 3). Second, STEM teachers face the challenge of using strategies that encourage students to "*use technology to represent what they know rather than reproducing what teachers and text books tell them*" (Howland et al., 2014, p. 6).

STEM teachers are publishing lesson plans and examples, which demonstrate the ways that technology can enrich STEM teaching in general, and with each individual subject. For example in Science, computer models play an important role in helping students visualise concepts, enabling students to replay simulations and interrogate concepts which generate different outcomes (Riga, Winterbottom, Harris, & Newby, 2017). Further examples can be found in the Technology curriculum, with teachers using learning experiences to expose students to technical theories and concepts with students also encouraged to design, plan and create technological artefacts (Avramides, Hunter, Oliver, & Luckin, 2015). While within Engineering, computer programming is used to expose students to algorithmic thinking (Thompson, 2017). Finally in Mathematics, teachers are using online game play to give students the opportunity to exercise critical thinking, analytical and problem solving skills (Stanford, Wiburg, Chamberlin, Trujillo, & Parra, 2016).

Finally, the above examples demonstrate 'science in action' with students given the autonomy to use technology to enrich their understanding of STEM. Ntemngwa and Oliver (2018) suggest that integrating technology into STEM activities offers the potential to "*expose students to the connections among and across these concepts and/or practices, and supports learning and/or application of the concepts simultaneously or in isolation. In this way, students* *learn to apply the synthesised concepts in authentic real life problems while using 21st century analytical skills*" (p. 12). Banks and Barlex (2014) argue that teaching STEM content is complex, as it involves teachers learning pedagogical and technological content knowledge. On the one hand, STEM content gives teachers the opportunity to combine concepts from different subjects to create learning experiences, which bring real world examples into the classroom. On the other hand, STEM teachers face the challenge of upskilling in technology to make science accessible and compelling.

To conclude, students are being encouraged to specialise in STEM. As a consequence of this, teachers are looking for new ways to create 'realistic' learning experiences, which engage students in learning science and give students the opportunity to learn through doing and exploring concepts. STEM teachers face the initial challenge of developing teaching strategies for delivering science lessons, which expose students to some of the ways that science is applied in real world contexts. STEM teachers face the subsequent challenge of developing technical expertise, to support students using technology to learn, through exploring science. While STEM activities create valuable opportunities for students to develop and apply critical thinking, problem solving, and analytical skills, a new approach to teaching is required which supports students interacting with Science.

2.2.2 Factors Influencing Teaching Computer Science (CS)

Teachers preparing to teach Computer Science (CS) face the challenge in learning "*appropriate pedagogies for delivering a new subject, particularly in those aspects of computer science that relate to algorithms, programming and the development of computational thinking skills"* (Sentance & Csizmadia, 2017a, p. 469). Barak and Assal (2018) advise that success in teaching CS topics including programming, and computational thinking depends upon the careful "*design of the course methodology and especially the students' assignments"* (p. 121). However Passey (2017) cautions that CS students need assistance from more knowledgeable peers to master "*technical, operational and application skills and competencies"* (p. 427).

Early research exploring the difficulties that novices face in learning computing, show that programming can be disorientating for those with no prior experience in the domain (Du Boulay, 1986). Learners can become confused with the understanding the nature of programming, or lack contextual understanding of the role that programs play in every day contexts. Learners can also fail to grasp the relationship between programs and hardware. A further difficulty may rest with failing to understand computing structures, including the role of algorithms, or the pragmatics of solving problems. Table 11, p. 31 provides an overview of computer programming difficulties, which learners can experience in learning programming.

Table 11 Computer Programming Difficulties (Du Boulay, 1986)		
Orientation		
What is programming?	*"Finding out what programming is for, what kinds of problems can be tackled and what the eventual advantages might be in expending effort in learning the skill"* (Du Boulay, 1986, p. 57).	(Anderson, 1993; Grover & Pea, 2013; Pea & Kurland, 1984; Stiller, 2009; Ulloa, 1980)
Notional Machine		
Understanding Hardware	*"There are difficulties in with understanding the general properties of the machine that one is learning to control, the notional machine, and realizing how the behaviour of the physical machine relates to this notional machine "* (Du Boulay, 1986, p. 57).	(Dennis, 2013; Halfacree & Upton, 2012; Michie & Johnston, 1985)
Notation		
Language, syntax and semantics	*"There are problems with the notation of the various formal languages that have to be learned, both mastering the syntax, and the underlying semantics"* (Du Boulay, 1986, p. 57).	(Andersen, 1992; Dagdilelis, Satratzemi, & Evangelidis, 2004; Eco, 1984; Lavonen, Meisalo, Lattu, & Sutinen, 2003)
Structures		
Achieving programming goals	*"Associated with notation are the difficulties of acquiring structures, clichés or plans that can be used to achieve small scale goals"* (Du Boulay, 1986, p. 58).	(Meerbaum-Salant, Armoni, & Ben-Ari, 2011)
Pragmatics		
Skills to specify, develop, test and debug a program	*"There is the issue of mastering the pragmatics of programming, where a student needs help to learn the skill of how to specify, develop, test and debug a program using whatever tools are available"* (Du Boulay, 1986, p. 58).	(Brito & de Sá-Soares, 2014; Clear, 2004; Ko, Myers, & Aung, 2004; Martinovic et al., 2014)
Pedagogy[1]		
Approach used to construct meaning	Consideration needs to be given to methods used in teaching to help students draw meaning and understanding from programming.	(Ben-Ari, 2001, 2004; Hazzan et al., 2014; Papert, 1993; Skinner, 1961)

[1] *The author added an additional field of 'Pedagogy' to Du Boulay's model.*

Over the past forty years, educators have strived to make programming accessible to students. Initial research by Bukoski and Korotkin (1976) observed teachers using problem solving as a strategy to encourage students to learn computer programming. Further research by Clark (1985) explored the impact of teaching methods on shaping students experiences of learning programming. Almost ten years later, Veen (1993) observed that teachers belief in new methods and practices are the key to successful learning experiences and student outcomes. In the last ten years, research by Paraskeva, Bouta, and Papagianni (2008) subsequently found that personal factors including "*computer self-efficacy, self-concept, attitudes, motivation and needs are considered crucial to the integration and development of modern technologies in education*" (p. 1084). These examples show that making programming content engaging to students remains complex, with teachers' personal beliefs, preferences, expertise, and technical confidence shaping learning experiences.

Today, Papadakis (2018) reflects that "*in the last 30 years, the scientific community has not stopped looking for new pedagogical approaches and teaching techniques in introductory computer programming courses*" (p. 1). Papadakis continues that the need to search for new ways to teach computing and programming content may be in part linked to the concern that *"traditional teaching approaches are unable to contribute substantially to the development of the necessary cognitive models by the students, producing high rates of failure and dropout in introductory programming courses"* (2018, p. 1). What this means is traditional approaches to teaching (based around laboratory sessions and problem solving) are perceived as somewhat incompatible with teaching programming. Papadakis's comments indicate that educators are still looking for alternative ways to teaching which provide meaningful ways to help students understand programming.

Mishra and Henriksen (2018) view programming as a 'creative process' and argue that teachers should consider integrating team based projects into computing lessons as a way to help students develop "*solutions far greater than would be possible with simply a human being working alone*"(p. 73). However Mishra and Henriksen (2018) were not the first to argue for using team based projects in a computing context. Ben-Ari (2001) suggests that organising students into teams for project work created opportunities for students to explore solutions and practice implementing concepts, supported by their peers. Zendler (2018) argues for using teamwork to create opportunities for students to take the lead in their learning, with projects used to give student different experiences and perspectives of computing. However, Hazzan et al. (2014) advise that team based project work is time consuming, and further recommends that teachers provide a structure which can help students develop new knowledge gradually, with teachers advised to encourage feedback and use tasks all students can complete. Furthermore, Lingard and Barkataki (2011) call for strategies which engage all students, as group learning *"takes a great deal of faculty time, effort and energy to guide groups of students in doing effective teamwork"* (p. 1).

Finally, both Mishra and Henriksen (2018), and Ben-Ari (2004) as well as Zendler (2018) have argued that incorporating teamwork into computing lessons creates opportunities for students to share technical knowledge and expertise, however orchestrating teamwork involves teachers spending more time interacting with students and facilitating problems solving to ensure that students receive the answer to their questions. Hazzan et al. (2014) caution that incorporating team based projects into computing lessons can provide a framework to help students explore and experience different roles involved in creating computing artefacts. In response to Hazzan et al. (2014), Yadav et al. (2017) call for teachers to design learning lessons which enable students to develop skills in lateral and critical thinking as well as project management, communication and design skills. While the option to learn computing at school gives student the advantage of developing computing skills as part of formal schooling, teachers face the challenge of preparing to teach a subject, which difficult to teach. To address this problem, educators are moving to use team-based, project work in teaching, with teachers encouraging students to share their experiences.

To conclude, the rationale for integrating computing into formal education comes from the European Commission as well as from educational systems who view computing and programming as important skills for supporting economic growth and supporting digital economies. While the option to learning computing at school gives student the advantage of developing computing skills as part of formal schooling, teachers face the challenge of preparing to teach a subject, which is perceived as difficult to teach. Malcom Knowles (1988) warns that in order to support students reach their learning potential as independent thinkers teachers need to 'follow the flow' and create learning experience which create opportunities for discussion and exploration. To address this problem, educators are moving to include team, and project-based methods in computing, with teachers playing a more interactive role encouraging students to share their ideas and creativity.

2.2.3 Summary

This section explored the impact that technology has on teaching, with examples showing the benefits of integrating technology into teaching, and an analysis demonstrating the use of technology in STEM and CS contexts. The biggest challenge that teachers face in integrating technology into classroom teaching relates to developing generic technical skills and the confidence to replace existing teaching methods with students learning though collaborating on projects, with students using technology to create content, which demonstrates their thinking and encapsulates their ideas. The integration of technology in STEM teaching is more specialised, with teachers not only facing the challenge of developing technical exercises, which simulate authentic work contexts. Teachers face the methodological challenge of adopting an integrated approach to STEM teaching, which means designing activities, which enable students to explore themes through different disciplines. Teachers face further difficulties in developing computing learning experiences, using technology to master concepts that are used in programming. However, there are solutions available to help teachers not only master strategies for integrating technology into teaching, and developing the technical expertise to supervise students using technology to master programming.

2.2.4 Gaps

To conclude, this section of the literature review revealed the paradox that teachers are now spending more time learning to use technology so that they can supervise students using more technology in their learning (Comi et al., 2017). Moreover, this section also shows that there are at least more than fifty different ways to use technology to enhance learning experiences (Atherton, 2018). The gap which emerged through this section of the review is that STEM teachers face the challenge of choosing the right strategy to teach STEM content, with the lack of an overarching pedagogy making it difficult for teachers to design lessons (De Vrieze, 2017). Du Boulay (1986) further identifies barriers to learning programming, with Mishra and Henriksen (2018) proposing that collaborative and project-based teaching methods play an important role in helping students develop a meaningful understanding of computing. Thus, further analysis is need to explore strategies used for learning and teaching programming, to uncover what approaches teachers use in the classroom, and which methods engage students.

2.3 Learning and Teaching Programming

At its simplest level, the process of writing a computer program involves writing and organising codes into sequences, which are processed by machines (Tenenberg et al., 2018). One of the benefits of learning programming is that learners develop the skills to structure their work, thinking in a sequence and developing the practical skills to identify and solve problems (Kalelioğlu, 2015). However, a negative aspect associated with learning programming is that some students may lack the capacity to bring together the individual elements of a program in the correct sequence (Kay et al., 2000). While learning computer programming creates opportunities for students to move into technology careers, some students perceive programming as just too difficult to learn. Thus, teachers are seeking examples of strategies, which can be adapted into activities that teachers feel confident in using to help students learn how to program.

A barrier that students experience when learning how to program is that of applying concepts from one context to another (Pea, 1987). The process of applying one concept in another context is called 'transfer', and it is a problem that students experience when learning computer programming (Salomon & Perkins, 1987). Campbell-Kelly and Asprey (2018) explain that students find it difficult in understanding the relationship between code, programs, and processes run on computers, and attributes this difficulty to the problem that "*the computer itself continues to evolve and acquire new meanings"* (p. xiv). Indeed Mayer (2008) calls for more research to clarify "*how students learn or what students learn from programming experiences*" (p. 2). While Lye and Koh (2014) urge educators to consider designing programming experiences which include problem solving as a way to expose students to different contexts, perspective and practices.

Du Boulay (1986) advises that there are three common programming mistakes that teachers "*should look out for"* (p. 58). These mistakes are described as (1) misapplication of analogy, (2) overgeneralisations, and (3) interactions (Table 12, p. 35 -36). Discussing each problem in turn, the first problem – the misapplication of analogy - involves making conceptual connections between phenomena, which are not related. The second problem relates to making overgeneralisations or claims without understanding basic concepts. In addition, the third problem means that students can experience difficulties in understand how different parts of a 'program' interact with each other.

Table 12 Common Programming Mistakes (Du Boulay, 1986)		
Mistake	Definition	Example
(1) Misapplication of Analogy	Learners make conceptual connections between phenomena, which are not related.	For example, since "*students often believe that since a variable is like a 'box 'it can hold more than a single value*" (Du Boulay, 1986, p. 58).
(2) Overgeneralizations	Learners make overgeneralisations without understanding basic concepts.	"*An example here might be the student surrounding an REM statement in Basic with quotes because the text following a PRINT statement is quoted*" (Du Boulay, 1986, p. 58).

Table 12 Common Programming Mistakes (Du Boulay, 1986) continued		
Mistake	Definition	Example
(3) Interactions	Learners are unable to handle complexity in general and interactions in particular.	For example "*we may find different sub-parts of a program improperly interleaved, or the perpetual shape of a program on the screen interfering with a correct appreciation of what its text actually denotes*" (Du Boulay, 1986, p. 58).

Du Boulay (1986) offers two suggestions for teachers that can help learners understand programming. The first suggestion is that teachers: *"need to present the beginner with some model of description of the machine she or he is about to operate via the given programming language. It is then possible to relate some of the troublesome hidden side-effects to events happening in this model, as it is these hidden, and usually unmarked, actions which often cause problems for beginners*" (Du Boulay, 1986, p. 72). The second suggestion "*concerns the way that learners form a view of how the programming language works and what is going on inside the computer. Very often they form quite reasonable theories of how the system works, given their limited experience, except that their theories are incorrect*" (Du Boulay, 1986, p. 72).

There are further barriers to learning programming that students can encounter (Table 13, p. 37). For example Grover, Basu, and Schank (2018) suggest that learners can struggle with understanding the meaning of 'code' which is a view shared by Campbell-Kelly and Aspray (2018). Samurcay (2013) reports that some learners can experience difficulties in mastering problem solving skills, however Lister (2011, 2016) suggests that problem solving skills are a core part of learning programming. Furthermore, not all students may understand the design process that is used to develop a computer program (Ko et al., 2004), which is connected to the point raised by Pea (1987) who states that students can struggle with 'transferring' one set of concepts into a different context.

Theorists are working to develop strategies to help students learn programming. Sentance (2018c) points to the work of Lister (2011, 2016) who argues that learning programming involves cycles of 'trial and error.' Piaget (1950) suggests that making mistakes motivates the learner to develop new strategies for solving problems. However Luckin and Du Boulay (1999) argue that group work can help play an important role in helping individuals develop strategies for solving problems and making connections between what the learner perceives are unrelated concepts. Sentance (2018c) further adds that both approaches, and others, play an important role in helping learner engage with computing, with teachers most in need in learning strategies to assist students.

Table 13 Barriers to Learning Programming	
Author	Contribution
Grover et al. (2018)	Learners do not understand the meaning of codes that are used in programming languages.
Samurcay (2013)	Learners struggle with developing the problem solving methods that are used to solve programming problems.
Ko et al. (2004)	Design issues, concept selection, co-ordination, use of programming languages, general problems with understanding, and information processing are barriers.
Du Boulay (1986)	Learners misunderstand the nature of programming; lack the content knowledge to understand relationships between code, programs, and machines; are unable to write code and organise codes into meaningful structures, and the failure to master sufficient skills to correct errors.
Pea (1987)	Learners struggle with learning how to transfer one series of concepts from one context to another.
Campbell-Kelly and Aspray (2018)	Students find it difficult understanding the relationship between code, programmes, and processes run on computers.
Lister (2011, 2016)	Learners develop a concrete understanding of programming through computing activities, which involve trial and error.

A pedagogical solution is proposed through using contextual and project-based learning experiences, which encourage divergent thinking but also give students the opportunity to explore concepts and apply their thinking in a practical setting. A past example using the seminal programming language LOGO[2], involved students in the planning, design, testing and implementation of programming projects, making programming accessible to learners (Clements & Gullo, 1984). A more recent example, involves students completing

[2] *Papert, S. (1993) Mindstorms: Children, Computers, and Powerful Ideas. New York, NY: Basic Books*

educational games as a learning methodology which encourages students to explore and develop links between programming concepts (Ma, Shang, & Xiao, 2017).

The above examples and others (e.g., Cutts et al., 2018; Williamson, 2016; Yildiz Durak, 2018) emphasise the need for learning experiences which provide students with the opportunity to strengthen thinking, planning and design skills. Programming languages designed to make coding concepts accessible to students include the visual programming language ‘Scratch’ (Resnick et al., 2009) and SNAP! (https://snap.berkeley.edu/). These languages are ‘block’ based with an interface, which enables students to connect blocks together to create a program. Examples in text-based programming include Processing.org (https://processing.org/) and Python (https://www.python.org/). Furthermore, the Raspberry Pi and its operating system (https://www.raspberrypi.org/) offer an entry point into hardware and physical computing. On-line communities connected to each of these languages and hardware systems aim to support learners create computing projects, providing support and advise which may help to change the conversation from ‘why is programming so difficult to learn’ (Bosse & Gerosa, 2017) to what can we ‘learn about students learning from using open-ended programming projects’ (Grover et al., 2018).

Finally, Resnick and Robinson (2017) suggest that students need not be deterred from learning programming, and argue that programming activities can be engaging and foster creativity. Resnick and Robinson (2017) propose that designing activities around projects and problem solving offer the potential to give students the freedom to design, explore, and create, their own artefacts. Mishra and Henriksen (2018) argue that programming is a creative process, with students learning the skills to manage the design process that is used to generate a finished artefact. Indeed, programmers working in a professional capacity need to have the skills to discuss ideas, manage projects, and the confidence to communicate their designs to development teams (Beecher, 2018). Thus, educators involved in teaching programming need to adopt strategies, which help students develop teamwork and project management skills as well as the essentials of programming.

To conclude, programming is perceived as a difficult topic to learn, with students finding difficulty in understanding the relationships between codes, programs, and functions performed on machines. Educators are designing contextualised learning experiences involving online and offline tasks as well as games and divergent thinking activities and projects to help students make connections between programming concepts. Moreover, educators are designing programming activities, which encourage learners to take the lead in their learning, though setting problem solving tasks, which engage students in the process of working out, proposing solutions, and retesting outcomes. These skills are perceived as essential for solving computational problems, with educators looking for new ways to help students deepen their problem solving skills. However, educators also require assistance with applying interactive approaches to teaching.

2.3.1 Current Strategies for Teaching Programming

Teaching programming involves designing tasks which involve students in the process of problem solving, planning, designing and testing computer programs (Sentance, Barendsen, & Schulte, 2018). However, there remains indecision as to the best way to teach programming. Mayer (2008) asks "*how can we teach children to use computers productively and what effect will learning to program computers have on them*" (p. xv). While Milne and Rowe (2002) observe that programming is perceived as difficult to teach "*because of the student's inability to comprehend what is happening to their program*" (p. 55).

Robins et al. (2003) identify four barriers to teaching programming (Table 14). The first relates to the development of relevant content knowledge and the second barrier concerns designing authentic or contextual learning experiences, which enable students to make connections between programming concepts. The third barrier involves developing the relevant methodological expertise to supervise students collaborating on projects, while the fourth relates to developing the confidence to help students solve programming problems. What Robins et al. (2003) propose are using learning experiences which are contextual, collaborative, and project and problem based.

Table 14 Barriers to Teaching Programming (Robins et al., 2003)
(1) Developing relevant content knowledge.
(2) Designing authentic or contextual learning experiences.
(3) Developing sufficient methodological expertise to supervise students collaborating in programming projects.
(4) Developing the confidence to help students solve their programming problems.

Major et al. (2012) take a different view and suggest a further three teaching barriers which relate to the complex nature of the subject, the negative stereotypes associated with programming and that 'introductory programming courses often fail to encourage student understanding' (p. 502). Major et al. (2012) further suggest using more guidance and facilitation to help students explore the meaning of concepts and links to other contexts and content, which students understand.

Both Robins et al. (2003) and Major et al. (2012) explore perceived barriers to developing content knowledge that teachers teach to their students. In a programming context, teachers need to develop a special type of content knowledge, which involves understanding computing and programming concepts alongside the technical expertise to teach the content to their students (Saeli, Perrenet, Jochems, & Zwaneveld, 2011). Qian, Hambrusch, Yadav, and Gretter (2018) argue that teachers new to computing and programming need the most assistance, given that they may lack the expertise and the confidence to use problem solving methods in their teaching.

Finally, there are different methods that teachers can use for teaching programming, but little consensus on which one to use (Sentance, Barendsen, et al., 2018). A sample includes teaching students programming through completing programming tasks individually, in computing laboratories (Chamillard & Braun, 2000); organising students to complete programming assignments in pairs (Dybå, Arisholm, Sjøberg, Hannay, & Shull, 2007); designing tasks which involve students completing programming projects in small groups (N. M. Webb, Ender, & Lewis, 1986); and using lectures or 'chalk and talk' methods to communicate concepts to large student cohorts (Matthíasdóttir & Arnalds, 2015). There also remains some dispute over which teaching methods are best at helping students learn programming (Kirschner, Sweller, Kirschner, & Zambrano R, 2018). Sentance, Sinclair, Simmons, and Csizmadia (2018) recommend that CS CPD programmes should use strategies which help teachers "*learn a range of new skills in terms of being able to plan and execute a small-scale research project, and their data shows that they gained an understanding of how children learn computing and ways in which deeper learning of computing can be facilitated*" (p. 23).

To conclude, just as there are methods for helping students learn programming, there is a need for CPD programmes, which equip teachers with the confidence and the expertise to guide students through tasks, mediate disputes between students, orchestrate group discussions, and use techniques to communicate ideas and content to large cohorts (Crook, 2018). Teachers need help with developing teaching strategies which enable them to address general concept related questions and equip teachers with the confidence to offer practical solutions to programming tasks (Haduong & Brennan, 2018). Computer programming is perceived as difficult to teach, with teachers facing the challenge of learning how to program. Teaching computing and programming involves problem solving, with teachers required to develop sufficient content knowledge and confidence to help students solve problems in the context of their work. Teachers have the option of using a number of methods for teaching programming, ranging from lectures to individual instruction. Thus, teachers need assistance in developing strategies, which enable them to teach programming, and empower them to help students solve problems.

2.3.2 Emerging Strategies for Teaching Programming

In recent years, teachers started to embrace an approach for teaching programming, which encourages students to work together or collaborate to solve problems. Supporters of a collaborative approach are Ben Ari (1998) and Sentance and Csizmadia (2017a) as well as Voogt et al. (2015) who agree that learning together helps students develop a practical understanding of programming. Haduong and Brennan (2018) observe, "*many novice programmers, the process of finding and fixing errors in code can be frustrating. Debugging is rarely explicitly taught in introductory programming courses, perhaps because best practices of teaching debugging are largely undefined. In K-12, teachers new to teaching CS may also experience trepidation about supporting student-directed work in languages and environments unfamiliar to them"* (p. 1092). Haduong and Brennan (2018) further suggest using 'debugging' as a collaborative strategy, to help students develop communication skills, confidence and programming ability.

Choosing a strategy to teach programming is perceived as difficult today, as it was in the past. Papadakis (2018) further observes that, "*traditional teaching approaches are unable to contribute substantially to the development of the necessary cognitive models by the students, producing high rates of failure and dropout in introductory programming courses. In the last 30 years, the scientific community has not stopped looking for new pedagogical approaches and teaching techniques in introductory computer programming courses*" (p. 1). Underlying a choice of strategy, is the need to provide students with access to content which enables students to formulate and construct new knowledge (Phillips et al., 2017). In a programming context, this means using teaching strategies, which give students opportunities to discuss and analyse what they have learned. Indeed Papadakis (2018) and Phillips et al. (2017) as well as Crook (2018) agree that traditional teaching strategies, in the form of lectures and text books are insufficient for teaching students programming. Table 15, p. 42 provides examples of collaborative, project-based approaches to teaching CS, with a summary provided of the key arguments for supporting a collaborative learning strategy in a CS context.

Educators are re-evaluating 'traditional' learning theory in an attempt to develop new strategies which to construct learning experiences which engage students in learning programming. Examples can be seen in the work of Lister (2016) who references the Piagetian theory of 'cognitive constructivism' to develop programming activities which support the 'accommodation and assimilation' of concepts (i.e., Gluga, Kay, Lister, & Teague, 2012; Lister, 2011; Lister et al., 2007). A further example can be seen in research on tutoring systems by Luckin and Du Boulay (1999) who adapted the Vygotskian theory of the 'Zone of Proximal Development' which they perceive as "*an appealing and persuasive idea for those concerned with how best to help learners learn. In essence, the ZPD requires collaboration or assistance for a learner from another more able partner. The need for this more able learning partner arises from the belief that the activities which form a part of the child's education must be beyond the range of her independent ability. The learning partner must provide appropriately challenging activities and the right quantity and quality of assistance"* (p. 1560). One can argue that Piaget and Vygotsky follow 'parallel paths to constructivism' (Pass, 2004), with each supporting the view that learning is a social process and shaped through interaction (Chalkin, 2003). However unique to Vygotskian theory is the proposal that 'meaningful' learning demands collaboration, with Luckin (2010) further observing that ZPD theory provides a lens to explore professional peer-learning in a CS context.

Table 15 Rational for Collaborative, Project-Based Teaching in CS	
Papadakis (2018)	Traditional teaching approaches are unable to contribute substantially to the development of the necessary cognitive models by the students.
Phillips et al. (2017)	Need for teachers to develop alternative teaching approaches, which help students develop cognitive models.
Crook (2018)	Collaborative activities, give students the opportunities to discuss and analyse what they have learned.
Haduong and Brennan (2018)	Using 'debugging' as a collaborative activity to introduce students to domain based skills such as solving skills.
Teague and Roe (2008)	Tangible collaborative activities are essential in helping students develop the confidence and practical expertise to write computer programmes.
Luckin and Du Boulay (1999)	Encouraging more able others to help colleagues explore computing concepts, assisting with developing strategies and meaning making.

Luckin and Du Boulay (1999) make the case for adapting collaboration theory as a strategy for teaching and learning computing, Vygotsky (1978) argues that knowledge is constructed through social exchange, with more able others helping learners identify and address gaps in their knowledge. This theoretical perspective presents an opportunity to develop strategies, which supports peer learning through teamwork, project-based learning, and practical activities. Vygotsky (1978) further suggests that peers play a crucial role in assisting with problem solving, using their knowledge to help others extend their knowledge. This suggests the need for a strategy that uses a facilitator / mentor driven approach, with individuals encouraged to assist team members. Finally, Vygotsky (1978) encourages complex forms of problem solving in the form of contextualised activities, to encourage learner autonomy and strengthen thinking. Problem solving skills are perceived as essential in programming, with Samurcay (2013) calling for collaborative strategies that use technology-mediated activities to encourage lateral thinking and project-based learning.

The benefit of adapting collaboration theory for teaching and learning programming, are the opportunities that are afforded to students to ask questions, demonstrate ideas, discuss problems and share knowledge (Blumenfeld et al., 1991). Indeed, Teague and Roe (2008) suggest that collaborative activities are essential in helping students develop confidence in mastering the practical expertise to write programs. Teague and Roe further observe that "*as students' progress through a first programming unit, they enjoy it less, find it more difficult than they expected, and have less confidence in being able to successfully complete it. The students also believed that collaborative learning would have a beneficial impact on their learning outcomes and make studying programming more engaging, interactive, and fun*" (2008, p. 147). Thus there is an opportunity to use facilitation and construct collaborative, project-based activities as a way to help students overcome perceived emotional issues ranging from 'anxiety and fear to boredom' (Tom, 2015). Melcer and Isbister (2018) suggest that practical tasks involving making "*tangibles have a greater positive impact on learning, situational interest, enjoyment, and programming self-beliefs. We also found collaborative play helps further reduce programming anxiety over individual play*" (p. 1).

An alternative approach to collaborative learning is proposed by Mayer (2004) and Kirschner, Sweller, and Clark (2006) who argue that students need strong guidance and explicit instructions to formulate knowledge. Furthermore, Lieberman (2001) argues that the best way to teach programming is through teacher led examples, rather than through student driven design. While Lewis (2011) argues that learning how to program is best experienced as an individual, with individuals controlling the pace at which they want to learn rather than at a pace set by others. Lingard (2010); Lingard and Barkataki (2011) argues that while there are benefits to be gained from setting individual programming tasks, teachers are looking for more social and collaborative ways to teach programming which include all learners, which may involve teachers using games, setting problems and configuring tasks which enable students to express their creativity.

Finally, computer science educators are leading the way in designing innovative, collaborative exercises in an attempt to encourage students to express their creativity and engage with programming. Educators have used magic tricks to teach computational thinking (J. Black et al., 2013; Curzon, 2007; Goode, Flapan, & Margolis, 2018); fairy tales to deconstruct problems (Kubica, 2012, 2013); and quests using algorithms to solve mysteries (Kubica, 2016). Vizcaíno, Contreras, Favela, and Prieto (2000) continue that 'creative' teaching strategies offer the potential to engage learners, adding that teaching methods which involve "*collaborative learning has many advantages such as the interchange of ideas among the students, or an increase of the motivation to learn*" (p. 263). Sentance and Csizmadia (2017a) further suggest, *"the use of collaborative work, peer mentoring, pair programming and other strategies is helping teachers to establish computational thinking skills in young students. What is clear is that there is a change for teachers"* (p. 489).

To conclude, programming is perceived as a difficult subject to teach, with students struggling with understanding general principles involved in writing computer programs. Thus, teachers are looking for new ways to teach the content, which make it easier for students to understand computing concepts. Traditional approaches to teaching programming (involving lectures, laboratory sessions, and problem solving) are perceived as inadequate in equipping students with the concepts that they need to engage with programming. Thus, teachers are looking for alternatives. Collaborative learning experiences are perceived to give students the opportunity to demonstrate concepts, with activities involving storytelling, magic tricks, and quests, helping students construct mental models (Table 15, p. 42). A collaborative approach to teaching can help students develop the practical expertise to program but implementing collaborative learning experiences gives students more time to explore computing concepts and share their own ideas.

2.3.3 Summary

This section explored barriers to teaching and learning computer programming and examined strategies which have been developed to help teachers and students overcome these barriers, with a collaborative approach to learning and teaching programme offered as a potential solution. The biggest challenge that learners face in developing programming knowledge and expertise and mastering strategies to resolve errors or correct problems that emerge when learners change or move around the contents of a computer program. The main challenge that teachers face in preparing to teach programming is in developing the confidence to apply content knowledge and problems solving skills to help learners understand programming errors. Both learners and teachers can benefit from a facilitator led, collaborative, project-based approach to programming, with practical activities providing learners with access to support networks to ask questions and demonstrate solutions, and teachers with a methodology, which enable them to consult with teams and assist with problem solving as and when problems emerge. The perceived educational success of a peer-led and project-based approach to teaching and learning programming has encouraged professional development designers to integrate social learning theories into professional learning.

2.3.4 Gaps

To conclude, this section of the literature review uncovered common mistakes experienced by the novice programmer setting out to engage with computing, and learn programming (Du Boulay, 1986). A sample of the problems that the novice programmer can encounter include misunderstanding codes (Grover et al., 2018), failure to master problem solving methods (Samurcay, 2013), and difficulties with program design (Ko et al., 2004), as well as problems with transferring concepts from one context to another (Pea, 1987). Furthermore, this section also revealed that teachers are using moving towards integrating collaborative and contextual activities into the computing classroom to give students the opportunities to discuss and critique what they have learned (Crook, 2018). This section of the literature review identified the gap that there is a lack of an overarching pedagogical model to help teachers implement collaborative programming experiences in schools. Moreover, Teague and Roe (2008) report that collaborative teaching and learning strategies provide the capacity to increase confidence enabling learners to take control over the programming process, however further analysis is required to understand what supports teachers need to prepare for using methods and for teaching content using collaboration.

2.4 Professional Development in STEM and CS

CPD programmes play a pivotal role in assisting teachers enhance their pedagogical content knowledge (Blömeke & König, 2012) and to enhance their teaching methods (Jarvis, 2012). CPD programmes also provide teachers with the opportunity to revisit beliefs about theories which underpin the construction of content knowledge (Krolak-Schwerdt, Glock, & Böhmer, 2014). One of the reasons that teachers use CPD is to explore enhancing their content knowledge to help students engage with their subjects (Harland & Kinder, 2014). CPD programmes also provide teachers with the opportunity to re-evaluate teaching established methods to help student achieve their learning goals (Scales, Pickering, & Senior, 2011). Teachers who attend CPD can use the experience to meet with other professionals and discuss planned changes to methods used in subject teaching, in an environment supported by peers and designed to provide educational guidance (Dudley, 2014).

The literature classifies CPD programmes as either following a 'traditional' (Martin, Kragler, Quatroche, Bauserman, & Hargreaves, 2014) or 'modern' approach (Hardy, 2012). Traditional CPD programmes are perceived as using '*knowledge transfer models'* such as lectures and other direct teaching methods to convey content knowledge (Dikilitas, 2015). While modern CPD programmes are perceived to use '*knowledge construction models*', which means that teachers learn through engaging in tasks which encourage teachers, as learners to construct content knowledge (Olsen, 2015). Both CPD approaches have educational merit (McInerney, 2013). 'Traditional' models are perceived as useful in helping teachers listen to and explore fundamental, to core domain concepts (McInerney, 2013; Moon, Butcher, & Bird, 2000). While 'modern' CPD models are perceived as useful in helping learners construct knowledge and give teachers the opportunity to experience the methods that are used to teach the content (Beijaard, Meijer, Morine-Dershimer, & Harm, 2005).

Both traditional and modern CPD programmes are designed to help teachers adjust teaching methods to enhance student learning, which is reported as the main reason why teachers seek access to CPD programmes (Scales et al., 2011). However, recent changes in educational policy have given teachers further reason to attend CPD, and that is to prepare for supporting collaborative, project-based and technology-mediated learning (Weimer, 2013). Some argue that changing teaching to a more 'student-centred' approach requires strong professional guidance, especially during the initial phases of preparation (Nelson, Spence-Thomas, & Taylor, 2015). Thus there is a need for CPD to immerse teachers in activities which involve planning, implementation, and the evaluation of technology-mediated, collaborative, project-based experiences (Desimone, Porter, Garet, Yoon, & Birman, 2002). CPD programmes are emerging which emphasise a collaborative approach to teaching, so that teachers can experience creating lessons using methods which support collaborative, technology-mediated project-based learning (Bolam & McMahon, 2005).

One might argue that what makes teaching in the 21st Century unique, is the requirement for teachers to integrate technology use into teaching (Amory, 2018). Teachers are spending more time developing methods for supporting students using technology to complete classroom tasks (Atherton, 2018). The increased use of technology in learning experiences has prompted teachers source CPD programmes to learn how to construct technology-mediated activities which enhance learning and support students leaning with technology (Compton & Almpanis, 2018). Integrating technology into subject teaching is reported to bring many instructional benefits (see section 2.2) however King (2002) advises that using technology in a teaching context can also be *"intimidating or frustrating"* (p. 283).

While some teachers are perceived as confident technology users (Baird & Clark, 2018), others are somewhat less confident with Borko, Whitcomb, and Liston (2009) calling for professional development which gives all teachers the time to practice using technology.

Garet, Porter, Desimone, Birman, and Yoon (2001) further suggest that successful professional development programmes should aim to have a 'positive effect' on teacher learning. Garet et al. (2001) continue that there are three factors which contribute to positive CPD experiences – these are a: "*focus on content knowledge; opportunities for active learning; and coherence with other learning activities"* (p. 915). However, there are other factors that contribute to success in a professional learning context. For example, it is important that CPD provides teachers the opportunity to reflect upon the impact that changing methods has on student learning outcomes (e.g., Guskey, 2000; Van Driel, Meirink, Van Veen, & Zwart, 2012). It is also important that professional learning activities give teachers the opportunity to explore using new skills and also give teachers the time to evaluate their impact (Sachs, 2011).

Finally, the measure of successful professional development are programmes which align with teachers pedagogical beliefs (D. Clarke & Hollingsworth, 2002). Baird and Clark (2018) suggest that professional development programmes which are perceived as having the most impact, are those which help teachers 'transform their teaching' and empower teachers "*to take ownership to identity and solve problems to impact their teaching and outcomes for their students*" (p. 327). Collaborative professional development programmes are designed to encourage teachers to share theoretical beliefs, express opinions, discuss practical expertise and evaluate the implications of adapting new theories of teaching and learning into their subject teaching (Hargreaves & Dawe, 1990). Collaborative programmes also aim to support teachers tailor prior content knowledge, expertise, beliefs and practices to new approaches. Kennedy (2016) agrees, and has observed that "*collaborative professional learning has grown enormously in popularity over the past decade or so, with common acceptance of the establishment of groups called, variously, teacher learning communities, communities of learning, professional learning communities and so forth"* (p. 667).

To conclude, changing teaching methods challenges teachers to think about their core beliefs about the process of teaching (A. King, 1994; Scapp, 2003; Von Glasersfeld, 1989). Furthermore, just as there are different methods that teachers choose to use to enhance their subject teaching, there are also different approaches to professional development which are designed to respond to emerging trends in education (Feldman, Altrichter, Posch, & Somekh, 2018). One of these trends concerns the integration of technology into teaching (Jones & Dexter, 2018) covering the preparation of teachers who are seeking to specialise in computing

(Maiorana et al., 2017). A second is CPD programmes which encourage collaborative learning, to provide teachers with an opportunity to learn from each other (Hargreaves & O'Connor, 2018). However Rogers (1995) cautions that in *"getting a new idea adopted, even when it has obvious advantages, is often very difficult*" (p. 1). The advantage of collaborative, and team based programmes is that they provide experiences *"where peers rely on the expertise and support of one another to adopt innovative practices"* (Glazer & Hannafin, 2006, p. 179). Collaborative programmes also encourage learning *"by collaborating with other teachers; by looking closely at students and their work; and by sharing what they see. This kind of learning enables teachers to make the leap from theory to accomplished practice"* (Darling-Hammond & McLaughlin, 1995, p. 598).

2.4.1 STEM Professional Development

One of the paradoxes of teaching in the 21st Century is that teachers are using less in direct teaching methods, in favour of methods which encourage students to take the lead in their learning (Jarvis, 2012). This shift in the teaching and learning paradigm has meant that teachers are presenting less content and supervising students constructing more of their own content (S. P. Marshall, 2009). With students being encouraged to take the lead in their learning, STEM teachers face the particular challenge of designing science based activities which enable students to grasp complex concepts (Kitts, 2009). Professional development programmes for STEM teachers therefore need to equip teachers with the content knowledge and expertise to design innovative learning experiences which motivate and equip students with the skills to explore science (Mayorova, Grishko, & Leonov, 2018).

The demand for STEM graduates has prompting some teachers to complete professional learning programmes which specialise in STEM (Varadharajan, Buchanan, & Schuck, 2018). STEM teachers need professional development which use activities which are project-based, technology focused and are constructed to encourage students to apply higher order thinking skills (Schuck et al., 2018). STEM programmes also need to provide learning experiences which enable teachers, both new to STEM and experienced in teaching the content, to collaborate, share professional and subject expertise and learn from each other's experiences (Hobbs, Clark, & Plant, 2018).

STEM learning activities are designed to involve students in "*problem-centred, inquiry-based, design-based, and cooperative learning*" (Thibaut et al., 2018, p. 190). A typical STEM learning experiences can be described supporting the open collaboration and sharing of ideas and concepts between students and teachers, who both participate in shaping the design and implementation of scientific ideas (Barak & Assal, 2018). Behaviours observed in STEM learning experiences include problem solving, where students are encouraged to explore new concerts and demonstrate their ability to rework designs (Burke & Burke, 2018). Working in this way requires teachers to adopt a facilitator led approach to teaching, which involves teachers interacting with students to mentor, guide, and assist students with solving problems they may encounter through completing their work.

STEM professional development programmes therefore need to mirror classroom practices (S. Kim, Song, Lockee, & Burton, 2018). STEM teachers need to the opportunity and the freedom to ask questions, seek clarifications and direction, as to how to plan, design, and implement STEM based projects. STEM teachers also need additional supports in planning and facilitating students researching STEM based work, especially given the general *"lack of guidance and direction from policy for STEM education"* (Montgomery & Fernández-Cárdenas, 2018, p. 4). In response, Barak and Assal (2018) propose that STEM CPD should provide a road map, and at a minimum cover *"(1) practice—basic closed-ended tasks and exercises; (2) problem solving—small-scale open-ended assignments in which the learner can choose the solution method or arrive at different answers; and (3) project-based learning—open-ended challenging tasks*" (p. 121). This road map, and others proposed by Zeidler (2016); Kelley and Knowles (2016); and Bybee (2010) and Kilpatrick and Fraser (2018) agree that peer learning is crucial for helping teachers develop confidence.

Finally, STEM teaching involves helping students explore and bring together concepts from other scientific domains, thus it is recommended that STEM programmes "*are mediated by STEM teachers who are responsible for organising, implementing and evaluating the activities with a view to promoting STEM subjects"* (Aslam, Adefila, & Bagiya, 2018, p. 58). Teacher collaboration is perceived as essential in helping to equip teachers with the confidence and the content knowledge for teaching STEM. Indeed Lambert, Cioc, Cioc, and Sandt (2018) argue that one of the "*greatest strengths of the (STEM professional development) program were the STEM connections that teachers began making; the changes in teachers instructional practices; improved attitudes, beliefs, and confidence in teaching; increased comfort with using technology; and the enthusiasm that students exhibited during a STEM lesson"* (p. 3). However Kilpatrick and Fraser (2018) advise that as STEM education continues to evolve, it is essential that STEM teachers are connected with professional networks to help teachers plan for new innovations that emerge within STEM and across Science.

To conclude, teachers opting to teach STEM face the challenge of teaching content, which involves students in the active construction ideas and the synthesis of concepts. STEM teaching therefore requires a different approach to teaching and learning which involves teachers working with students to develop skills and strategies, which enable them to bring together concepts and develop ideas. A shortage of STEM teachers in post-primary education has created new opportunities for teaching professionals from across the curricula to qualify as STEM teachers, with teachers taking advantage of this shortfall as an opportunity to deepen their experience and develop collaborative, technology, mediated, project-based methods. Thus professional development programmes are required which are interactive and use problem solving and project-based tasks in an attempt to equip teachers with a practical understanding of STEM.

2.4.2 Computer Science (CS) Professional Development

Governments, and local educational systems are active in designing curricula to introduce computing into schools (Barr & Stephenson, 2011; Caspersen, Gal-Ezer, Nardelli, Vahrenhold, & Westermeier, 2018; Hubwieser, Armoni, Giannakos, & Mittermeir, 2014; Yadav et al., 2017). The requirement to teach computing in schools has highlighted a lack of teaching capacity (Neutens & Wyffels, 2018) which has created a need for teachers qualified to teach computing (E. Roberts, 2018). Teachers are taking advantage of new professional development programmes to develop the content knowledge, computing skills and methodological expertise to teach new curricula in computing. However providing computing professional development programmes does not necessarily ensure that teachers are prepared to teach CS (Hamlen, Sridhar, Bievenue, Jackson, & Lalwani, 2018).

Traditional professional development programmes are perceived as teacher-centric, using lectures, text books, and problem solving to convey content to learners (Heaysman & Tubin, 2019). A didactic form of knowledge transfer is perceived as a 'tried and tested' teaching method and teachers use this method to impart core concepts to large cohorts of students (Smerdon, Burkam, & Lee, 1999). Didactic teaching methods also focus on the dissemination of information, using demonstrations and discussion to cover new concepts or processes (Smerdon et al., 1999). A limitation with a didactic methodology is that it is perceived to limit teachers in their ability to delve deeper into questions and limits the amount of time that teachers can spend on solving problems raised by students (Hamilton, 2018). Teaching can include lectures (Kaasbøll, 1998), but the sole use of lectures limits the time that students can learn through direct engagement with computing.

Collaborative, and project-based teaching methods involve students in the construction of knowledge, with teachers using tasks which involve students in developing and applying computing concepts and skills (Kaasbøll, 1998). A further benefit of a collaborative and project-based approach to teaching is that students learn a number of other skills used in the computing domain, including leadership and project management, design skills as well as communication and presentation skills. Furthermore, Abernethy and Treu (2009) agree that students need to learn more than just the technical skills, and advises that "*while teaching technical skills to students of computer science and information technology is critical, it has become increasingly clear that computing professionals must also excel at the "soft" skills of communication and interpersonal interactions"* (p. 178). Social skills (such as listening to others, talking about ideas and sharing expertise) are also important in helping students develop the confidence to explain designs and approaches to peers (Bell & Newton, 2013).

There are a number of reported problems in using a collaborative, project-based approach to teaching computing content. First, teachers may lack the practical expertise in facilitating collaborative, project-based learning (Stronge, 2018). Second, teachers may feel that they have a less than *"adequate computer science background"* and therefore lack the pedagogical content knowledge to facilitate students and support teams working collaboratively (Yadav et al., 2017, p. 235). Thirdly, teachers may be somewhat reticent *"to bring coding into classrooms"* given the expectation that teachers are experts when they themselves are at the start of learning and developing methods for teaching computing (Kong & Wong, 2017, p. 377). In contrast, research by Hamlen et al. (2018) explored the use of collaborative learning theory in a professional learning context. Hamlen et al. (2018) found that the "*key goals of the program were to develop ability and confidence in programming skills among teachers and students, and to train and encourage teachers to use peer instruction, allowing for a great deal of interaction among students and engagement with the content facilitating the development of expertise among students"* (p. 741). The results further *"showed that teachers improved in both knowledge and confidence after taking the workshop, and the gains were evident for their students as well"* (Hamlen et al., 2018, p. 741).

Computing educators are exploring the impact that a collaborative and project-based approach to teaching computing has in a CPD context. A further example is provided by Sentance et al. (2013; 2012; 2014; 2016) who cite Kennedy's (2005, 2014) professional development model as the inspiration for a programme designed for computing teachers. Sentance and Csizmadia (2017b) argue that there is need to expand professional development programmes to equip teachers with the knowledge and methods that they need to teach computing, and to connect teachers with communities of practice to help teachers make lasting changes in their teaching. Moreover, Quille, Faherty, Bergin, and Becker (2018) call for closer collaboration within the CS community, to increase opportunities to share content knowledge and professional expertise for implementing computing and programming lessons.

The CPD model that Sentance and Csizmadia (2017b) propose includes collaboration in a six tier process covering peer mentoring and access to online sources as well as academic support to evaluate lessons and links to computing communities (Table 16).

Table 16 Collaborative CS CPD Model for Teachers (Sentance & Csizmadia, 2017b)
(1) Communities of practice – teachers are encouraged to work together towards a common goal such as implementing a new strategy, share experiences and explore practices.
(2) Training – training is focused on exploring pedagogical issues to support collaborative work with other teachers.
(3) Mentoring / coaching – peer coaching supports knowledge sharing between teachers of equal knowledge, while mentoring supports knowledge sharing from expert to novice.
(4) Accreditation – ongoing accreditation, of professional recognition of professional learning status demonstrating growth of specialist knowledge, expertise in school and peer contexts.
(5) Teacher research – providing an infrastructure for teachers to design and implement methods, which provide metrics on the performance and the impact of changes to classroom practice.
(6) Cascading good practice / knowledge - teachers feel confident, supported and encouraged to share worked examples, perspectives, and pedagogical beliefs with peers and colleagues.

In the model described above by Sentance and Csizmadia (2017b), collaboration theory underpins professional learning experiences, with the CPD programme structured to encourage connections with communities of practice external to the CPD as well as nurturing collaboration between teachers during learning experiences. Sentance and Csizmadia also proposes using a facilitator led approach to teaching, with teachers encouraged to share knowledge and expertise as well as contribute their understanding of computing phenomena. A further strength in the model proposed by Sentance and Csizmadia (2017b), is an emphasis on developing teaching methods over time, with teachers also encouraged to document their teaching experience and report on the outcome of using particular methods and strategies. Sentance and Csizmadia provide a powerful and integrated model which provides an opportunity for teachers to share professional and subject expertise as well as giving teachers access to 'experts' and mentors who can assist with shaping and developing learning experiences which teachers plan to use with students in schools.

The accreditation element of Sentance and Csizmadia (2017b) model is covered by a BCS Certificate in Computer Science Teaching which is available to primary and post-primary teachers across England. The first stage of this two-year programme involves teachers engaging with computing materials, including online tutorials and attending professional meetings linked to computing which demonstrate engagement with professional development which surpass minimum threshold of 20 professional development hours. The next part of the programme involves teachers working on projects covering a computing topic, with teachers encouraged to design tasks, which give students a 'contextual' understanding of computing. Sentance and Csizmadia (2017b) give the example of a 'computer science knowledge quiz' developed in Python. The final part of the certificate programme encourages teachers to explore the computing curriculum, with teachers given the option to choose an 'aspect of computer science pedagogy', implement the methodology, and evaluate the results. Assessment is completed and accreditation is achieved through the submission of evidence (covering projects, the design of materials used in classrooms), which is validated by assessors who provide formative feedback through each stage of the certificate.

Another example of a similar accredited CS CPD programme comes from Hazzan et al. (2014).This programme is designed to support teachers cover a computing curriculum which also includes research activities, with teachers completing modules that are linked to a national centre for compuing (Hazzan, et al., 2014, p. 237). The programme provides workshops for teachers as well as opportunities to implement workshop content in schools. Teachers are encouraged to use online reources and collaborate with peers to develop lesson plans. A combination of self-directed and group activites are required to qualify for a 'license' to teach CS.

Both examples aim to help teachers prepare to use a collaborative, project-based apprpach to teaching CS. In later work, Sentance (2018a, 2018b) report that teachers need access to CPD which makes the content accessible, while Passey (2017) calls for CPD which involves teachers in designing, implementing, evaluating the implementation of CPD content. Essentially, what is needed is professional development which provides computing tasks which *"lie within the learner's 'zone of proximal development' and provides enough support to allow the learner to succeed"* (Ridgway & Passey, 1991, p. 6). Collaborative activities incorporating projects, teamwork, and facilitation play an important role in helping teachers master theory and processes that are used to teach computing concepts (Guzdial, 2016). The above examples and others (see Condamines, 2011; Flatland et al., 2018; Herawati, 2018; Warner, Fletcher, Monroe, & Garbrecht, 2018) support a form of CPD, which is collaborative, project-based and designed for building confidence in CS.

Finally, as the field of computing professional development continues to grow, governments and their educational systems need to ensure that teachers are equipped with the confidence, knowledge and expertise to teach computing (Ekmekci, Parr, & Fisher, 2018). Research exploring collaborative, project-based approaches argue that CPD designers need to place "*more effort in creating a sustainable community of practice so knowledge and experiences can still be shared even after the program has finished*" (Neutens & Wyffels, 2018, p. 840). However Ott, Ureel, and Wallace (2018) caution that *"maintaining a Community of Practice for CS teachers, however, can be challenging. Demands on teacher time, lack of institutional buy-in, physical isolation, and lack of appropriate peer institutions are some confounding factors"* (p. 1067).

To conclude, the concept of collaborative, project-based and facilitator driven professional development is gaining in popularity as teachers seek out programmes, which help them develop knowledge and expertise for teaching curricula, which involve students taking more ownership and responsibility for their learning. The benefit of attending student-centred professional development programmes is that they use teamwork and encourage teachers to share experiences, practices, and knowledge as well as use tasks, which support learning by doing. STEM teaching involves experiments, field trips, projects and practical work, thus STEM teachers require a CPD offering, which enables teachers to develop the practical expertise for supervising students completing practical work. Computer Science teachers are seeking similar CPD offerings to develop strategies for supervising students. Thus, CPD designers are moving to incorporate teamwork, facilitation and projects based models into CPD programmes which in turn has led to the need develop evaluation frameworks to determine their impact on teaching and learning.

2.4.3 Summary

This section explored current constraints with professional development programmes and examined the rationale for developing collaborative learning programmes, which aim to equip teachers with confidence and expertise to teach contextual, team-based, learning experiences. The move to collaborative, professional learning programmes comes from the evaluation of current methods of professional development, which are failing to equip teachers with the practical expertise to use teaching methods, which incorporate technology into subject teaching, and support a facilitator driven, student centred approach to project-based learning. The development of collaborative professional learning programmes for STEM teachers, which immerse teachers in authentic learning experiences and use teamwork and technology-mediated activities, give teachers the practical experience of using techniques, which place responsibly for learning with students. Collaborative professional development programmes in computing aim to use a collaborative approach to teaching and learning to empower teachers in the use of project-based methods and problem solving techniques, which are used for assisting students overcome barriers to learning programming. The rapid development of collaborative professional learning programmes has raised the need to establish what role collaborative theory plays in equipping teachers with the expertise to implement collaborative learning experiences in their subject areas, thus educators are calling for evaluation metrics to be applied to collaborative programmes to assess impact on teaching practice.

2.4.4 Gaps

To conclude, research by Montgomery and Fernández-Cárdenas (2018) identifies a lack of guidance and direction in STEM educational policy, with Aslam et al. (2018) confirming that to fill this 'gap' STEM teachers are taking the lead in creating, implementing and evaluating learning experiences. Sentance and Csizmadia (2017b) and Hazzan et al. (2014) as well as Passey (2017) and M. Webb et al. (2017), and others (e.g., Condamines, 2011; Flatland et al., 2018; Herawati, 2018; Warner et al., 2018), identify the need for more research to explore collaborative, project-based CPD programmes used in the capacity to help teachers build confidence needed to teach CS. In response to this need, Sentance and Csizmadia (2017b) further propose a 'transformative model' of CPD, drawn from professional learning theory (e.g., Kennedy, 2005, 2014) which combines collaboration with project-based activities and self-directed learning. In response to the need explore CS teacher CPD requirements and impact, the following section explores evaluation and design in a CPD context.

2.5 Programme Evaluation Design and Methods

An evaluation can be described as a systematic form of measurement (Weiss, 1998). Evaluations are used to examine the performance and impact of phenomena, which may occur within systems, or social context (Rossi, Lipsey, & Freeman, 2004). What distinguishes evaluations from other forms of social research, is the underlying requirement to identify factors which impact upon social phenomena (Van Den Akker, Gravemeijer, McKenney, & Nieveen, 2006). Wholey (1987) adds further clarity, reporting that evaluations used in social contexts are useful for measuring *"program performance (resource expenditures, program activities, and program outcomes) and the testing of causal assumptions linking program resources, activities, and outcomes"* (p. 77). Evaluations contain measurement criteria that are designed to explore, test, analyse, and validate programme outcomes (Heshusius, 1990). A limitation in applying measurement criteria is that the criteria are viewed as 'subjective', in so far that they are created then administered to produce a particular series of results which may be perceived as 'biased' (Fetterman, 2005). A further limitation in applying measurement criteria is that the process may generate research results which don't match the social programmes founding aims and objectives (Chelimsky & Shadish, 1997). Thus, there is a need to build 'objective' measures into the measurement criteria to ensure that evaluation results correspond with the programme outcomes and that the findings are reflective of the programmes aims.

Educational evaluations are used across a number of contexts to perform "*a wide array of activities, including student assessment, measurement, testing, program evaluation, school personnel evaluation, school accreditation, and curriculum evaluation. It occurs at all levels of educational system, from the individual student evaluations carried out by classroom teachers, to evaluations of schools and districts, to district-wide programme evaluations, to national assessments, to cross-national, comparisons of student achievement*" (Kellaghan, Stufflebeam, & Wingate, 2012, p. 1). However Cronbach, Rajaratnam, and Gleser (1963) argue that a limitation of evaluation research is that different frameworks are perceived to generate different outcomes. Thus, an essential component of evaluation design is the need to clearly define the outcomes that require measurement (Guskey, 2000). For example measurement metrics can help educators determine if teachers are prepared with the expertise, and confidence to apply professional learning outcomes. However, evaluating learning outcomes is perceived as complex with Shaha, Lewis, O'Donnell, and Brown (2004) recommending that "*evaluating the efficacy of professional development offerings, and validating their impact, requires a multi-dimensional approach*" (p. 2).

There are number of steps, which can be put in place to add validity to the measurement of social programmes (A. M. Black & Earnest, 2009; Posavac, 2016; Schalock & Bonham, 2003). McLaughlin and Jordan (2010) suggest organising a planning meeting with programme stakeholders to clarify learning outcomes. While Newcomer, Hatry, and Wholey (2010) advise using an established framework to structure the evaluation process. However, different frameworks place different emphasis on measurement criteria (Shadish, Cook, & Leviton, 1991). This variance in design means that the same measurement criteria can be applied inconsistently (Chelimsky & Shadish, 1997). Furthermore, the lack of centralised framework and validity criteria (Cronbach, 1983a) means, as Scriven (1991a) suggests, adapting existing frameworks which contain measures for moderating data collection and measures for validating the credibility of the evaluation results.

Adapting an existing evaluation framework can be problematic. However Rossi et al. (2004) recommend integrating formal research methods into evaluation frameworks to guide the systematic collection and analysis of data. Furthermore, professional associations such as Joint Committee on Standards for Educational Evaluation (1994) and the European Evaluation Society (2018) provide community support, publications and materials that evaluators can adapt to their interventions. However Stake (1967) further cautions that *"educators differ among themselves as to both the essence and worth of an educational program."* Stake continues that evaluation design is a subjective process in that "*neither a strict preordination design or a responsive design can be fixed upon an educational evaluation; rather, choice of design should be governed by how far the evaluator wishes to go beyond values and standards*" (1967, p. 287).

One area of consistency within evaluations is the measurement of learning outcomes (Jonassen, 1991). However, there are those who are critical of measuring learning outcomes which are perceived as giving a limited view of complex social phenomena (e.g., Roessger, 2015; Thurlings & den Brok, 2017). For example, Mezirow (1996) challenges evaluators to rethinking the treatment of the learning outcome paradigm, and to consider what other factors can be measured. While Van Merriënboer and Kirschner (2017) continue that measuring learning outcomes means that evaluations present results which "*are dealt with one at a time…….thus, the learner is taught only one or a very limited number of constituent skills at the same time. New constituent skills are gradually added, and it is not until the end of the instruction – or at all – that the learner has the opportunity to practice the whole complex skill"* (p. 6). Evaluators, including Fetterman, Kaftarian, and Wandersman (2015) recognise this constraint, providing alternative measures which explore factors including confidence, collaborative learning, empowerment, and practice. Harden (2002) supports the exploration of learning outcomes, as they "*represent what is achieved and assessed at the end of a course of study and not only the aspirations or what is intended to be achieved*" (p. 151).

Finally, there are a number of published frameworks, which provide the capacity to evaluate learners' perceptions of their learning as well as their experience. For example: Kirkpatrick (1956, 1994); Cronbach et al. (1963); Stufflebeam (1966) and Fetterman et al. (1996) examine learning outcomes in addition to attitudes, satisfaction, behavioural change, confidence. However, there is disagreement on the order followed and the methods used to support 'knowledge transfer' claims (Weiss, 1998). Baird and Clarke (2018) add that "*rarely do we measure the impact of professional development on teacher learning, implementation and student outcomes"* (p. 326). In response Biggs (1999) states that evaluations play an important role in generating results which explore change in learning and development giving the example that: "*objectives express the kinds of understanding that we want from students, the teaching context encourages students to undertake the learning activities likely to achieve those understandings, and the assessment tasks tell students what activities are required of them, and tell us how well the objectives have been met"* (p. 57).

To conclude, evaluations are a research methodology, which use measurement criteria to investigate social and educational programmes, to verify that programmes meet their objectives. Evaluators are advised to clarify programme outcomes, adapt existing frameworks and use established research methods to guide implementations. Constraints with the measurement of the learning objective paradigm, has prompted evaluators to explore alternative measures, exploring learning outcomes to capture data on satisfaction, confidence, perceptions of behavioural change and practice in context (Harden, 2002). Within education, there is still some dispute on which methods to use to provide visibility of the impact of professional learning programmes; however, there is also a case to be made for focusing on the teacher experience, and the issues that teachers face in implementing new theories and practices.

2.5.1 Evaluating STEM Professional Development

Growth in the demand for students with STEM skills had meant that evaluators are exploring professional development performance to understand how to equip STEM teachers for teaching the curricula. Evaluation designers are seeking to gain a better understanding of "*teachers views of STEM activities, how they understand their role as primary facilitators and the impact of their STEM engagement on their professional development*" (Aslam et al., 2018, p. 58). Aslam et al. (2018) and others (e.g., Kilpatrick & Fraser, 2018; Lambert et al., 2018) have identified the need for evaluation which explores knowledge 'transfer' between professional development and classroom practice. Davis, Garcia, and Stephenson (2018) seek to better understand teacher experiences, with others looking to identify what professional supports teachers need to teach computing (i.e., Cutts et al., 2017; Decker et al., 2018; Menekse, 2015; Qian et al., 2018; Sentance et al., 2014).

In STEM, as with other domains, professional development programmes play a critical role in helping teachers respond to policy change which impact upon teaching practice (Aslam et al., 2018). Nadelson, Seifert, Moll, and Coats (2012) further stress that we use evaluation frameworks to better understand how to support STEM teachers, and argue that "*in service teacher professional development is critical to achieving the goal of enhancing student knowledge of STEM*" (p. 69). Teachers can attend a number of different professional development approaches to strengthen their STEM content knowledge, including demonstrations (Cousins & Brooke, 2018), lesson study (Lampley, Gardner, & Barlow, 2018), action research (Hazzan, Heyd-Metzuyanim, Even-Zahav, Tal, & Dori, 2018), case study analysis (Yadav & Beckerman, 2018). Further examples include, one-to-one, and group coaching (Newton, 2018), teamwork including professional networking (Tytler, Symington, Williams, & White, 2018) and online professional development (Loucks-Horsley, Stiles, Mundry, Love, & Hewson, 2009). Kilpatrick and Fraser (2018) argue that 'effective' CPD should help teachers address barriers, and empower teachers to make the changes they need to their teaching.

There are a number of different ways that researchers can collect evaluation data which explore teachers experiences of professional learning programmes (Caracelli & Greene, 1993). A first step may involve establishing what types of research methods that are compatible with programme evaluation. In STEM professional development programmes "*there is no prescription for which designs are right for which situation – no "paint by numbers kit" for professional development"* (Loucks-Horsley et al., 2009, p. 25). The lack of a centralised approach to STEM programme evaluation makes it difficult to decide which type of framework to use and what type of methods are best suited to exploring STEM professional learning outcomes.

Finally, evaluators can use qualitative, quantitative, mixed methods and case study approaches to explore STEM professional development programmes (Bybee, 2010). Asghar, Ellington, Rice, Johnson, and Prime (2012) have applied qualitative methods to investigate *"teachers understanding and perceptions of problem-based learning (PBL) as an approach to interdisciplinary STEM education as well as their perceptions of the personal and systemic challenges in implementing such an approach in their professional practice"* (p. 85). While Saxton et al. (2014) have applied quantitative methods to investigate nine constructs examining "*student learning to teacher practice to professional development to school-level variables"* (p. 18). In contrast Allen, Webb, and Matthews (2016) have used a case study to evaluate the claim "*that teachers who possess a well-developed STEM pedagogical content knowledge, a constructivist paradigm of teaching and learning, and an ability to draw on a vision while reflecting on and during teaching to help negotiate challenges are well positioned to engage in the process of adaptive teaching"* (p. 217). These examples, and others (e.g., Awad & Barak, 2018; Ebert-May et al., 2015; Nadelson et al., 2013; Vennix, den Brok, & Taconis, 2017) recommend using research methods, which give results that cover preparation, and planning as well as the impact of implementations.

To conclude, professional development programmes play different and important roles in helping STEM teachers acquire the knowledge and professional expertise they need for teaching STEM content. Researchers face the challenge of adapting evaluation methods to investigate professional learning programmes to ensure that programmes meet their aims and learning outcomes. The evaluation process that researchers use includes adapting formal methods (such as qualitative, quantitative and case study strategies) to guide the design, evaluation, and analysis of data collected from professional development contexts. There are number of different formats of professional development that teachers can attend to help them identify gaps, with evaluations useful in exploring what content and methods are used in a CPD contexts and impact on practice.

2.5.2 Computer Science Professional Development Evaluation

Researchers working in Computer Science CPD design are adapting evaluation theory to investigate the extent to which professional learning programmes are meeting their goals (Warner et al., 2018). Research is underway to explore the extent to which CPD programmes are providing quality learning experiences (S. Davis, Garcia, et al., 2018) and are successful in supporting teachers implement computing lessons (Flatland et al., 2018). The rational for adapting evaluation theory to explore CS CPD programme performance comes from the need to establish what impact educational theories and models have on teacher preparation and the processes used to teach computing. Guzdial (2014) adds that it is essential that we explore teacher experiences of computing professional learning programmes as "*computing teachers need pedagogical content knowledge, which includes awareness of common misconceptions, methods for diagnosing those misconceptions, and interventions to help students develop more robust conceptions"* (p. 1).

With universities now offering professional learning programmes for computing teachers, evaluation theory and models provide the capacity to explore the link between university programme outcomes and the practicalities that teachers face in teaching computing in schools (Reding & Dorn, 2017). Reading and Dorn (2017) continue that it is essential that we evaluate professional learning programmes to ensure that teachers are prepared to 'independently teach' computing. Further research suggests that it is important to confirm that teachers are confident to engage with the rigours of high quality computing lessons (Flatland et al., 2018, p. 958).

Moreover, researchers are exploring the potential of evaluation theory to examine the outcome and impact of professional development programmes on teaching computing in schools (Cutts et al., 2017; Decker et al., 2018; Sentance, Barendsen, et al., 2018). University

CPD programmes, are designed to empower teachers with theory, practical skills and the confidence to enhance their teaching (R. E. Lee, 2018; Turner, Christensen, Kackar-Cam, Fulmer, & Trucano, 2018). The emergence of university designed professional development programmes in computing and computational thinking has generated a need to verify that teachers are equipped to teach computing and are also confident in their ability to plan future lessons and activities (Neutens & Wyffels, 2018). Evaluation theory offers the potential to explore organisational outcomes and learning objectives, and has been considered by Sentance et al. (2014), Cutts et al. (2017), and Ravitz et al. (2017) to assess CPD outcomes and impact on empowering teachers to teach CS.

Researchers are playing a more active role in facilitating CPD earning experiences and are designing measures which assess performance for programmes that are *"taught by university faculty"* (Cabrera, Morreale, & Li, 2018, p. 141). For example, Perry and Boylan (2018) have adapted Clarke and Hollingsworth's (2002) interconnected model of professional growth to explore teacher learning needs and examine the impact of CPD on practice. While Sentance et al. (2014) draws from Guskey (2000) to put in place measures to explore CS CPD programme performance.

Finally, university professional learning programmes give teachers the opportunity to attend out of school workshops in third level institutions. These workshops enable teachers to implement and review the outcome of changes made to professional practice under the guidance of colleagues and academic peers (Patton, 2011). University programmes also aim to help teachers identify knowledge, and expertise gaps, and to provide a platform for teachers to explore and discuss the implications of changing professional practices to incorporate new theories, methods and content. However, universities want to benchmark CPD programme performance to ensure that theories, activities, and projects help teachers make the changes they want to make to their practice.

To conclude, Pollock et al. (2017) report that *"professional development (PD) programs continue to be an essential mechanism for preparing in-service teachers who have little formal background in CS content, skills, and teaching pedagogy"* (p. 477). Roberts, Prottsman, and Gray (2018) argue that it is essential that professional development providers put in place metrics to track participation, scope and impact. Martinez, Gomez, Moresi, and Benotti (2016) suggest that more analysis is needed to understand why teachers are more "*likely to replicate the same activities they experienced during PD workshops in their classrooms than to produce their own"* (p. 77). Understanding what content teachers teach and what methods teachers use informs CPD theory and design which CPD providers need to consider including in their programmes (Carl, 2009).

2.5.2.1 Comparison of Guskey and Kirkpatrick Evaluation Models

An essential part of the CPD evaluation process is ensuring that teachers voices are heard, documented and analysed (Carl, 2009). Rubin (2018) continues that for professional learning programmes to have meaning, evaluations need to include the analysis of teacher perceptions and their experiences. Level models of evaluation are suggested as providing a bridge between the exploration of teacher perceptions and experiences. This usefulness of level models in CPD evaluation is supported by Coldwell and Simkins (2011) which report that "*continuing professional development (CPD) evaluation in education has been heavily influenced by 'level models', deriving from the work of Kirkpatrick and Guskey in particular, which attempt to trace the processes through which CPD interventions achieve outcomes"* (p. 143). The five level framework proposed by Guskey (2000) as well as the four level model proposed by Kirkpatrick (1994) provide the capacity to explore the learners experience of the training and analyse the impact on workplace practice.

Guskey's (2000) five level model (Table 17, p. 63) links teacher perceptions with teacher experiences. Guskey is widely used in education to evaluate CPD (e.g., Kelchtermans, 2004; Lydon & King, 2009; Muijs & Lindsay, 2008; Roesken-Winter, Schüler, Stahnke, & Blömeke, 2015; Sugrue & Mertkan, 2017). Guskey's model starts by exploring '*level 1 - participant's reactions; level 2 - participant's learning; level 3 – organisational support and change; level 4 – participants use of new knowledge and skills; and level 5 student learning outcomes*' (Guskey, 2002, pp. 46-49). Furthermore, Muijs, Day, Harris, and Lindsay (2004) describe Guskey as a design which provides a clear structure which can help evaluators "*think about gauging impact at different levels, and may be related directly to different orientations and intended outcomes*" (p. 299).

Educational researchers have adapted Guskey's (2000) model to explore factors influencing teachers use of technology in teaching (Bouslama, Lansari, Al-Rawi, & Abonamah, 2003; Persico, Manca, & Pozzi, 2014). Guskey has also been used to explore science teachers experience of using technical tasks and activities or enhancing science teaching (Zambak, Alston, Marshall, & Tyminski, 2017). Further use of Guskey can be seen in exploring computer science professional development (i.e., Cutts et al., 2017; Sentance et al., 2014), with Sentance further reporting that Guskey provides a framework for exploring teacher confidence and preparedness to teach CS. Moreover, Guskey provides a pathway and clear linkage between CPD outcomes and student learning outcomes.

A perceived limitation with Guskey's (2000) model is the reported lack of a process of follow up with teachers to review what impact professional learning experiences had on changing practices (D. Clarke & Hollingsworth, 2002). Slavin (1987) raises the concern that the model lacks pre and post-tests, to explore change over time. While Coldwell and Simkins (2011) caution level models are inadequate for exploring the links between programme outcomes and CPD performance. Boylan, Coldwell, Maxwell, and Jordan (2018) add that single pathway CPD evaluation models, which link training to classroom context, such as Guskey and Kirkpatrick, are difficult to implement if researchers do not have access to schools, and classroom access to teachers and their students.

Table 17 Comparison between Guskey (2000) and Kirkpatrick Model (1956)

Guskey's (2000) five level model		Kirkpatrick's (1956) four level model	
Level 1	'Participant's reaction's'	Level 1	Reactions - 'the degree to which participants find the training favourable, engaging, and relevant to their jobs.'
Level 2	'Participant's learning.'	Level 2	Learning - 'the degree to which participants acquire the intended knowledge, skills, attitude, confidence and commitment based on their participation in the training.'
Level 3	'Organisational support and change.'	Level 3	Behaviour – 'the degree to which participants apply what they learned during training when they are back on the job.'
Level 4	'Participants use of new knowledge and skills.'	Level 4	Results – 'the degree to which targeted outcomes occur as a result of the training.'
Level 5	'Student learning outcomes.'		

In comparison, Kirkpatrick's (1956) initial four level model is used to evaluate training and professional learning interventions. The Kirkpatrick model evaluates "*Level 1 – reactions – the degree to which participants find the training favourable, engaging and relevant to their jobs; Level 2 – learning - the degree to which participants acquire the intended knowledge, skills, attitude, confidence and commitment based on their participation in the training; Level 3*

– behaviour - the degree to which participants apply what they learned during training when they are back on the job; and Level 4 – results - the degree to which targeted outcomes occur as a result of the training" (Kirkpatrick Partners, 2018, p. 1). Further implementations include using the Kirkpatrick the four level framework to explore change in the knowledge transfer process (Aluko & Shonubi, 2014); and to determine professional development 'effectiveness' (Merchie, Tuytens, Devos, & Vanderlinde, 2018; Praslova, 2010).

As well as being used to evaluate CPD programmes, Kirkpatrick (1994) has also been adapted as an instructional design model (Dick & Johnson, 2007; Weston, McAlpine, & Bordonaro, 1995); and to explore community engagement (Watkins, Leigh, Foshay, & Kaufman, 1998). Further adaptations include using Kirkpatrick as a method to evaluate web services (Moller, Foshay, & Huett, 2008); and online training (Davidson-Shivers, Rasmussen, & Lowenthal, 2018); game design (Landers & Armstrong, 2017); interactive online tutorials (Turnbow & Roth, 2017); and to evaluate the outcomes of a digital technology professional development programme (O'Neil & Pegrum, 2018). Millwood, Strong, Bresnihan, and Cowan (2016) have adapted Kirkpatrick to explore teacher perceptions of pair programming learning experiences for teachers working in different jurisdictions.

The continued adaptation and implementation of the Kirkpatrick model across domains and professions over the past fifty years demonstrates its flexibility and versatility in evaluating training programmes (see Faerman & Ban, 1993; Lefkowitz, 1972; Moffie, Calhoon, & O'Brien, 1964; Noe & Schmitt, 1986; Praslova, 2010; Ruiz & Snoeck, 2018; Tamkin, Yarnall, & Kerrin, 2002). Kirkpatrick offers the potential to bring clarity in the absence of an overarching framework, and provides the flexibility to explore teacher perceptions of their learning in a workshop context and in a workplace context. However, there are limitations in adapting Kirkpatrick. In the first instance, there are *"three problematic assumptions of the model may be identified: (1) the levels are arranged in ascending order of information provided. (2) The levels are causally linked. (3) The levels are positively inter-correlated"* (Alliger & Janak, 1989, p. 331). Managing the construct of causality within level based or sequential frameworks, which seek 'transfer', remains problematic, with researchers having to declare the strategies that are used to manage subjectivity in data collection and analysis process.

Further limitations in adapting the Kirkpatrick (1994) model are that *"even successful training programs cannot guarantee that newly learned knowledge and skills will be transferred to the workplace. This has led to researchers' interests in understanding the transfer process. Notwithstanding that transfer issues have been studied for several decades, the recent emphasis on 'workplace learning', especially the so-called 'situated learning'* *approach, suggests that conventional training transfer research may be inadequate to understand the dynamics of performance improvement through training"* (Cheng & Hampson, 2008, p. 327). In response to this, the main strength of the Kirkpatrick model is the ability to link programme learning outcomes to workplace performance. However in opting to implement Kirkpatrick, evaluators still face "*the difficulty in, implementing all four levels"* (Reio, Rocco, Smith, & Chang, 2017, p. 35).

The most ardent of Kirkpatrick's critics, calls the four level model 'flawed' (Holton, 1996), countering that "*the four-level system of training evaluation is really a taxonomy of outcomes and is flawed as an evaluation model…such a model needs to specify outcomes correctly, account for the effects of intervening variables that affect outcomes, and indicate causal relationships"* (p. 5). Further criticism comes from Bates (2004) who argues that level models in general, and Kirkpatrick in particular need *"to specify outcomes correctly, account for the effects of intervening variables that affect outcomes, and indicate causal relationships"* (p. 341). Bates (2004) raises the issue that setting out research to explore four areas, which promote the transfer of learning from workshop to work settings is difficult to measure, and even more difficult to prove. While academics argue that Kirkpatrick is two rigid, making it difficult to address issues, which emerge through implementation (see: Kaufman et al., 1996; Kaufman & Keller, 1994; Watkins et al., 1998).

There are academics that argue that there are theoretical flaws in designing evaluations, which transcended evaluation model design, and are anchored to the learning outcome paradigm. A concern raised by Malan (2000) is that if there is "*uncertainty about the desired learning outcomes and failure to assess outcomes properly (this) could end in a situation where learners only attained pseudo knowledge, pseudo-skills, pseudo-attitudes and pseudo-values*" (p. 22). A further problem with basing evaluations on learning outcomes is that the results are subject to bias, given that outcomes are constructed to test for instances of particular phenomena (Prøitz, 2010). Indeed Murtonen, Gruber, and Lehtinen (2017) content that "*well-defined objectives in terms of learning outcomes can be useful for students and help those who are responsible for developing and evaluating study programmes. (However) there is a danger, however, that if the theoretical background of the "learning outcome" concept is not considered or not known, the use of learning outcomes can lead to unintended consequences*" (p. 115).

Both Kirkpatrick (1994) and Guskey (2000) are anchored to learning outcomes, with both seeking evidence of change emerging through shifts in learning outcomes prior to and after professional learning interventions. While Guskey (2000) ends with the measurement of student learning outcomes, Kirkpatrick (1994) stops short, putting in place measures which explore links between the participant, the training context, workplace performance, and change in practice.

Finally, the above studies raise the concern that using a predetermined structure to explore CPD programme performance and learning outcomes may give a somewhat restricted view of complex social phenomena. In response to this Kirkpatrick (1994) suggests using the four level model as a way in which to organise an evaluation process, rather than as a sequential structure. Rather, Kirkpatrick provides the flexibility to capture initial teacher perceptions of professional learning experiences and their experiences of using new content, producing results which not only provide an insight into the challenges that teachers face in teaching new content and methods, but also the difficulties which CPD designers can use to revise their programmes in response to changes.

Moreover, level models of evaluation, such as those proposed by Guskey and Kirkpatrick provide a pre-determined structure, which confines the researcher to following a particular evaluation pathway. However, given the need to understand the teacher experience of preparing to teach computing, there is a case to be made for focusing on the teachers experience as a learner (Darling-Hammond & Bransford, 2005). Furthermore, designing evaluations to focus on the teacher experience, provides an opportunity to explore teachers perceptions of CPD and to assess whether the CPD is meeting their learning outcomes (Lawless & Pellegrino, 2007). A teacher centric evaluation process gives teachers a voice, and captures the teacher perspective (Adelson, 2017).

To conclude, the need to evaluate computing professional learning programmes is just one research stream that is emerging in computer science, with others reported as "*computing education as technological training, as training for software development, as a central element for the field's academic recognition, and as training for computational problem-solving in any domain of knowledge"* (Tedre, Simon, & Malmi, 2018, p. 1). Each of these themes demonstrates that computing education is changing. While there are limitations in adapting evaluation theory, particularly level models, Kirkpatrick provides a structure, which spans workshop and workplace contexts, giving evaluators the opportunity to examine learning and implementation issues, and focus on the teachers' experience of professional learning programmes in CS. While there is a need to explore the impact of professional programmes on student performances, there is a somewhat greater need to focus on the teacher experience, concentrating on giving teachers voice, and putting in place measures and supports that give teachers control other the curriculum and their learning.

2.5.3 Summary

This section explored the concept of evaluation theory and its use as a form of measurement used in educational programmes to explore participants' perceptions of their learning and to examine 'performativity', or the impact of professional learning interventions on workplace performance. There are a number of frameworks that are used to evaluate professional development programmes, with researchers in education adapting evaluation theory to explore what impact professional learning experiences have on learning outcomes, with the results seeking evidence of the impact that collaborative learning activities have on teachers achieving their learning outcomes. The development of professional learning programmes in CS has given rise to university-school partnerships, which enable teachers to attend professional development, to learn methods, content, and expertise in designing, implementing, and reflecting upon the outcome of computing learning experiences, which are taught in schools. However, divergence in the evaluation community has led to severe criticism of level based models of evaluation, with transference between levels and the capacity to implement all levels in equal depth perceived as limitations. The literature shows that 'level models', such as those proposed by Guskey (2000) and Kirkpatrick (1994) provide clear links and a structure to explore concepts that are developed and enacted in workshop environments in school contexts. Researchers are adapting level models to gather evidence and produce metrics which not only verify which elements of professional learning experiences teachers perceived as 'effective' but also provide the capacity to explore teacher self-reported experiences of implementing professional development content, over time, giving a longitudinal view of change. However both Guskey (2000) and Kirkpatrick (1994) are linked to learning outcomes, which may limit exploration, but also provide the opportunity to focus further on particular learning events.

2.5.4 Gaps

To conclude, this section explored evaluation theory, and its application as a methodology in STEM and CS contexts. Scriven (1991a) advises adapting frameworks to guide the evaluation process however Stake (1967) cautions that framework selection is a subjective process. Further analysis reveals that learning outcomes provide a consistent measurement in programme evaluations (Mezirow, 1996). However Van Merriënboer and Kirschner (2017) caution that building evaluations around the learning objective paradigm is subjective, while Fetterman et al. (1996) argue for metrics which assess confidence, practice, and empowerment. The lack of an overarching and centralised framework overseeing the standardisation of validity criteria for conducting evaluations makes assessment problematic

(Cronbach, 1983a), with the literature providing examples of academics tailoring frameworks to meet research and organisational needs. Furthermore the rational for selecting an evaluation framework, is linked to access to the research context, as well as the researchers ability to plan research, which links learning outcomes to professional practice. Guskey and Kirkpatrick provide frameworks, linking the CPD context to work place performance, with Guskey providing further reach into the student experience. A limitation with implementing Guskey and Kirkpatrick is access to work place contexts. The advantage to be gained from using Kirkpatrick is the capacity to focus on teachers' professional practice as it relates to CPD and professional practice, using robust methods to overcome the problem of 'inadequate explanatory power' which Guskey (2016) argues has resulted in the models' somewhat limited use in education.

2.6 Limitations

This literature review provides a 'representative view' of the literature, with the themes arranged to explore the implications of using technology in education; complications associated with teaching and learning computing; the role that CPD plays in assisting teachers preparing to teach STEM and CS and the role that evaluation theory plays in assisting researchers build metrics to explore STEM and CS CPD. In contrast to systematic and meta-analysis approaches, this review, compiled using an adapted version of a meta-synthesis methodology has attempted to construct a rational supporting the evaluation of professional development services for CS teachers to find out what impact, collaborative methods have on learning and teacher preparation for teaching computing. Furthermore, the themes explored in the literature review highlight a gap in the literature emerging around the need to understand what role collaborative methods play in teaching and learning computing. To address this gap, a research opportunity has emerged in Ireland, with Trinity College Dublin adapting a collaborative, project-based, and technology-mediated approach to teaching and learning for use in a professional development programme for computer science teachers. The next chapter provides the theoretical background and context to the Bridge21 CS CPD programme, with further sections describing the pedagogical approach and activities used within the programme.

3 Bridge21 CS CPD Context

The literature review covered the analysis of international research, which highlighted the need to explore what supports teachers need to develop computing content knowledge (M. Webb et al., 2017) and appropriate methods that can be used for teaching computing (Caspersen, 2018). These challenges also impact upon teachers in Ireland who are preparing to teach new curricula in coding and Computer Science.

This chapter begins by exploring computing in an Irish context covering the design of new curricula in Coding (for lower secondary level), Computer Science (as upper secondary subject option) and programming (for teaching in primary school). A description of Bridge21 is then provided covering pedagogical and activity models and their use in subject teaching. A summary of Trinity College Dublin's Post Graduate Certificate in 21st Century Teaching and Learning follows which includes an overview of the computing modules, which make up the Bridge21 CS CPD programme. Having provided a general overview of the Certificate programme, the content of each of the Bridge21 CS CPD computing workshops are then described. The final section provides a chapter summary.

3.1 Computing and the Irish Context

The Irish educational system consists of pre-school, primary and post-primary or secondary schools. Children between the age of 3 to 5 can avail of free tuition at designated pre-school facilities while primary school children complete eight years of state education covering years 5 to 12. Post-primary education comprises of two cycles, with the first covering years 1 to 3 (also referred as lower secondary) and the later covering years 4 to 6 (referred to as upper secondary). All students from the age of 12/13 cover core curricula in Irish, English and Mathematics over the first three years of their secondary education, with students given the option to complete certificates in other subjects (including Music, Art, Science) which the remaining time table spaces filled by supplementary short courses (DES, 2018c). Students from the age of 15/16 cover a reduced syllabus for the remaining three years, with the latter two years spent studying subjects (at ordinary or higher level) for state examination, three of which must include the mandatory subject options of Irish, English and Mathematics (DES, 2018c).

3.1.1 NCCA Short Course in Coding

The current lower level short course in Coding covers three strands including an introduction to computer science, computing systems and coding (NCCA, 2018b). Teachers design leaning activities which enable students to harness skills which are compatible with coding, including implementing and expressing ideas, thinking creatively, goal setting, working with others and taking ownership of their learning (NCCA, 2018d). Teachers need to ensure that computing activities enable students to master skills which comply with four statements of learning which are relevant to Coding (see NCCA, 2015 for full generic list). Thus teachers need to plan lessons which engage students in devising strategies and reasoning skills to solve problems; that involve students in analysing and interpreting patterns within data; that also encourage students to use appropriate technologies to complete a design task, and enable students to apply their technical ideas (NCCA, 2018d). Assessment and reporting procedures are open to adaptation, with teachers involved in designing criteria.

3.1.2 NCCA Leaving Certificate in CS

A new upper secondary Computer Science leaving certificate subject provides practical and in-depth exposure to computing, with students completing an end-of-course examination (worth 70%) with the remaining 30% awarded for the submission of project work incorporating concepts from three strands or topic areas (NCCA, 2018d). This curriculum is designed to support differentiation, using a student-centred approach to teaching. The curriculum follows a modular design to encourage students to explore different elements of computing, with the curricula content covering computational thinking, computers and society, and design and development. There is explicit emphasis on project work, giving students the opportunity to apply problem solving skills and incorporate date analytics, data modelling and embedded systems concepts (NCCA, 2018d). Teachers are encouraged to use methods which support students in brainstorming solutions, creating digital products, and applying communication and problem solving skills as part of the computing design process (NCCA, 2018d). The NCCA (2019) are working with N = 40 pilot post-primary schools across Ireland to implement the new leaving certificate in Computer Science, with the Irish Government (2018b) reporting that the first cohort of students will sit exams in 2020.

3.1.3 NCCA Coding Programme for Primary Schools

Further work is underway in the primary sector, with the NCCA involved in the process of developing a coding programme for teaching at primary level, through the Coding in Primary Schools Initiative. This involves the NCCA working with schools who opt to teach coding as part of the curriculum (NCCA, 2018c). The NCCA are working with an initial N = 15 Irish primary schools to implement the Coding syllabus (NCCA, 2018c). The NCCA are planning to work *"with more schools in the coming months, in particular, schools which have done little or no work on coding before. As part of this, schools will be offered on-going professional development and support, and have the opportunity to work with and learn from teachers in other schools as* *they get involved in hands-on, project-based approaches to teaching coding and computational thinking in the classroom. This work will help NCCA to tease out and clarify how, to what extent, for what purpose, and where computational thinking and coding could be integrated in the primary curriculum"* (2018c, p. 1). The development of the primary coding programme, in combination with the national implementation of the lower secondary level Coding short course, and the further piloting of the upper secondary Computer Science leaving certificate in *N* = 40 post-primary schools nationwide, gives teachers new options and fresh opportunities to change path and specialise in CS. Having explored the computing context across primary and post-primary sectors, the following section covers the Bridge21 model, its design and implementation, as well as its use across subject teaching within the Irish education system.

3.2 Bridge21

Bridge21 is a model of 21st Century Teaching and Learning developed in TCD (Lawlor, 2016), which is used by post-primary teachers to encourage collaborative, team-based, technology-mediated learning in their subject teaching across Ireland (Lawlor et al., 2018). The Bridge21 pedagogical model contains eight elements (Table 2, p. 7), which combine to create collaborative, project-based, technology-mediated teaching, and learning experiences. A Bridge21 learning experience seeks to encourage students to work collaboratively to complete tasks, with facilitators using techniques including open ended questioning (Lawlor et al., 2016), which aim to nurture learner autonomy. Bridge21 learning experiences involve teachers switching role from that of leader, to the role of facilitator, with students encouraged to take individual responsibility for their learning. The Bridge21 activity model, developed in TCD by Byrne (2018), is used to structure Bridge21 activities (Table 3, p. 7). The Bridge21 activity model provides a seven step consecutive sequence covering the design, implementation, and evaluation of projects, with teachers acting as a facilitator / mentor, guiding students through each step of the design process. A description of both models now follows.

3.2.1 Bridge21 Pedagogical Model

This section provides a summary of each of the eight elements of the Bridge21 pedagogical model, which contribute to student-centred, project-based, and collaborative learning experiences.

(1) Teamwork

Collaboration lies at the core of the Bridge21 pedagogical model and involves the teacher organising students into small groups or teams (no less than 3). The teacher uses a sorting criteria (based on knowledge and experience of interacting with their students) to sort individuals into functioning teams. The model for organising students into teams is influenced by the patrol system of learning espoused by the World Organisation of the Scout Movement. In this context, each team should contain ‘scouts’ of different ages and genders as well as different levels of experience, and abilities.

(2) Skills development orientation

Skills orientation means that learning experiences are designed to focus on the learning by doing paradigm, with teachers encouraged to design tasks which are skills centric, thus students are encouraged to research, present, build, construct, think about, analyse and reflect upon the process of constructing content and leading the design process.

(3) Social learning protocols

Social learning protocols are learned behaviours. Lawlor et al (2018) provide examples which include using facilitation and group work as a context to help students developing the confidence to liaise with peers, taking more responsibility for their actions, and the skills to act as a team member.

(4) Facilitator and/or Mentor(s)

Through changing role to that of a facilitator or mentor, teachers are given the opportunity to engage with groups, and encourage individuals to contribute to team tasks, and ensure equal contributions from all group members participating in activities. Teachers use the role of facilitator to ask questions and challenge learners, prompting students to give explanations, or give demonstrations, with groups and individuals encouraged to share knowledge and expertise.

(5) Reflection

Reflection forms a core complement of learning experiences, with teachers using student reflections at the end of a lesson or workshop to encourage students in teams to reflect on what they have learned, which includes giving individuals the opportunity to discuss issues or raise concerns with group work and learning in a team. Facilitator/mentors use reflection to prompt for team and individual contributions as well as to facilitate broader discussion between individuals and teams, as well as between teams.

(6) Technology-mediated tasks

Technology-mediated tasks are integrated into learning experiences and can include using computers to search the internet for resources, using tools to create content or programming. Teachers design the task and format of the technology. Tasks can also include offline activities.

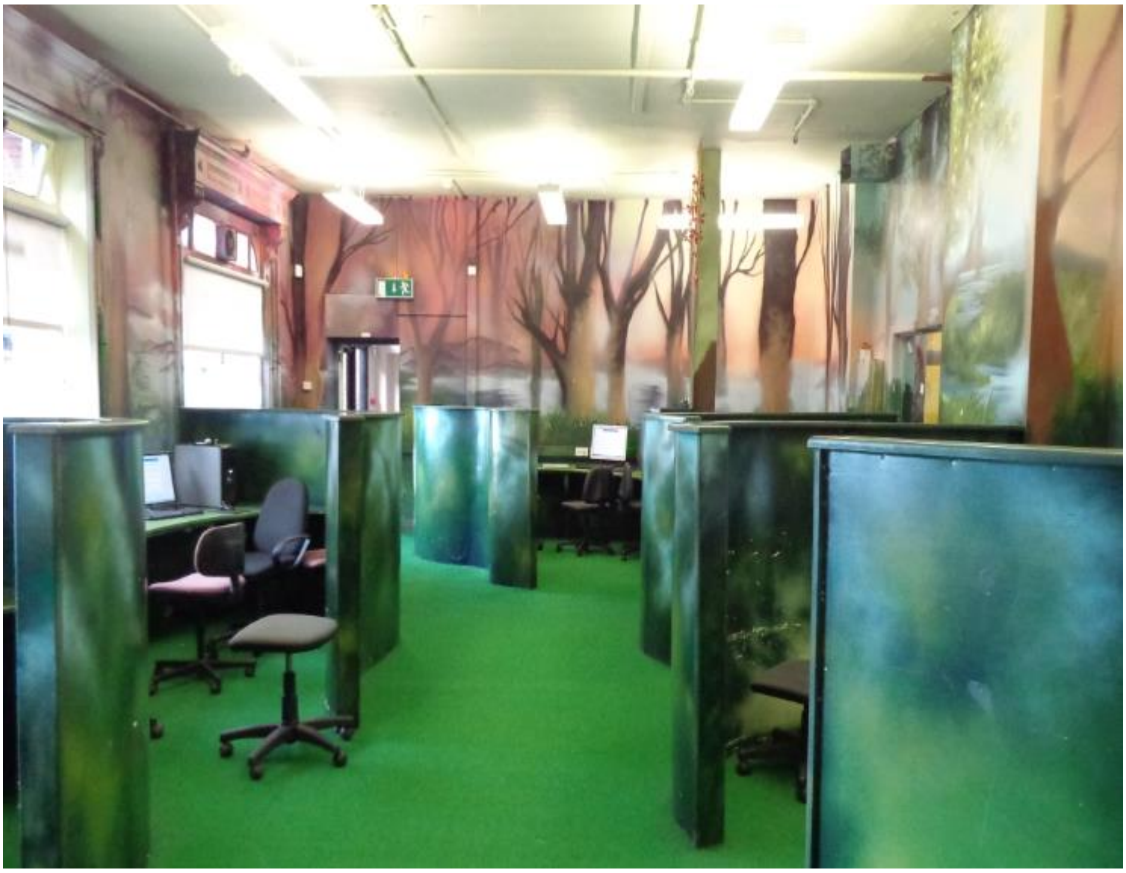
(7) Project-based learning

Project-based learning is used to bring together planning work, assigning roles, managing time lines, scheduling tasks, using technology, and monitoring the completion of projects to coincide with teacher assigned deadlines. For example, teams are encouraged to assign roles to each individual, Individuals can opt for a particular role, or an elected team leader can nominate particular roles.

(8) Learning space

The learning space describes the configuration of desks into clusters, to facilitate students working on projects. Figure 3–1 provides an image of the 'pod' space where teams work on projects.

Figure 3-1 Bridge21 Learning Pod Space (Reprinted with Permission, CPD Teacher, 2013)



3.2.2 Bridge21 Activity Model

Underpinning each Bridge21 learning experience is the activity model developed by Byrne (2018), which is a structure that is followed while teaching subject content. The activity model is inspired by design thinking (T. Brown & Wyatt, 2010). What follows is a description of the seven phases of the activity model. Each phase is completed in sequence with facilitators spending more or less time on a particular element of the sequence depending on the content and nature of the associated tasks.

(1) Set-Up Phase

The 'Set up' phase involves an 'Ice Breaker' for participants to get to know each other. The teacher orchestrates *'Team formation'*. A 'Team Name and Charter' task follows which involves deciding on a team name and charter which lists ways of working the team will adhere to, e.g. giving all members a chance to talk, listen to all suggestions, with teams committing to 'having fun'.

(2) Warm Up

Warm up activities follow. A typical warm up activity involves teams discussing and brain storming a topic related to the activity that follows. This activity is used to encourage discussion and start the process of interacting with peers on a particular content area. The facilitator consults with each group, asking questions around a central theme or topic they have introduced, with groups then invited to share their findings, one by one, as the teacher asks teams to report on their work.

(3) Investigation

The investigation moves from exploring general topics and themes to more specific content tasks, and the facilitator uses this part of the learning experience to introduce core concepts and materials. During this phase the facilitator can use a number of tasks or activities which are designed to examine the subject matter in more detail. Figure 3–2 on p. 75 captures brain storming around the topic 'computers in everyday life', where each team member is encouraged to write down comments.

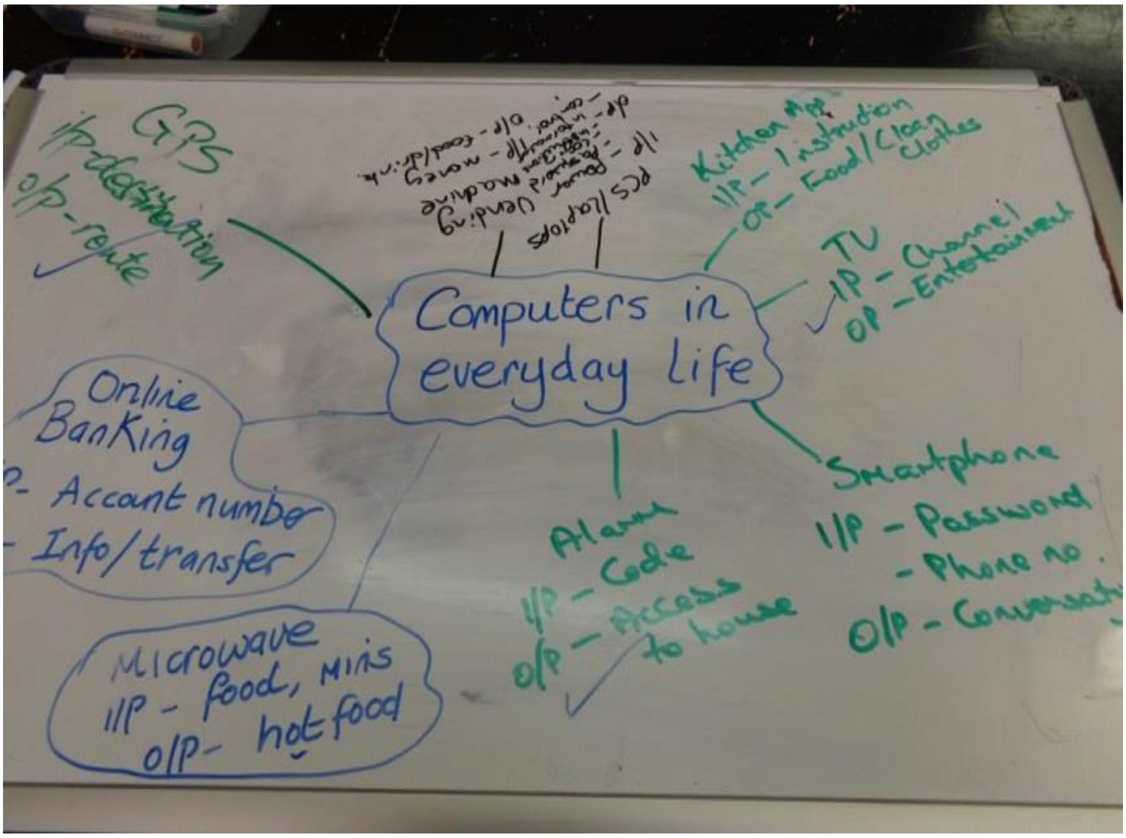
(4) Project planning

Project planning involves teams working together to divide work and assigning roles and tasks, engage in project planning and define schedules to ensure that projects are completed within the specified time frame. During this exercise, teachers set out the parameters of the project, which is to be completed and gives supplemental instruction including videos, demonstrations, as well as facilitating discussion on concepts that may be new to students or require clarification.

(5) Create

The create phase involves teams working through creating their artefact following a cyclical process. Students create, test, review, and retest their artefacts. Teachers play an advisory mentoring roll and are on standby to assist with technical issues, problem solving and give suggestions as well as encouragement. Teachers check in with teams regularly, and ask team leaders to report on group updates, communicate implementation problems or feedback on timelines with projects. Teachers end the create phase by giving teams a deadline to compile projects, finish tasks and prepare artefacts or projects ready for the group presentation in front of their peers.

Figure 3-2 Bridge21 Brainstorming Computers in Every Day Life Activity Example



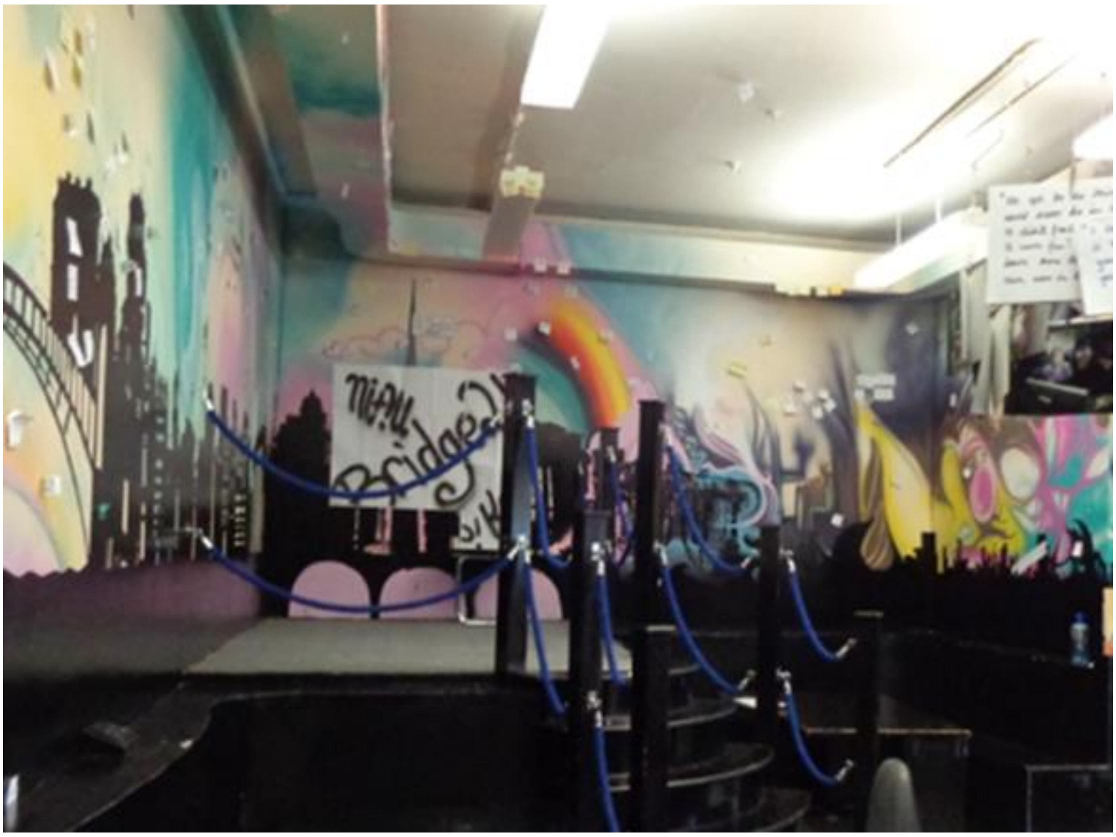
(6) Presentation

Presentation phase involves teams presenting the outcome of their projects to peers. This phase represents the 'finale' or the end of a Bridge 21 learning experience. Here, teams present their work (Figure 3–3, p. 76), and discuss what they had learned both as a team, and individually, with other teams encouraged to ask questions or engage with the games or the created artefacts made by peers.

(7) Reflection

A reflection phase comes at the end of the create phase and gives individuals and teams the opportunity to demonstrate their learning. Teams discuss the outcome of the create phase and prepare content for presentation. During this phase, individual students can record and report on elements of the create phase or the learning experience as a whole that was satisfactory or report on elements which were problematic. The reflection phase also gives students time to think about what impact the learning experience had their learning and talk about ideas, suggestions, and recommendations for improving future learning experiences. Reflection phase enables individual teams to meet with colleagues working in other teams to share their learning, and experiences.

Figure 3-3 Bridge21 Auditorium Space (Reprinted with Permission, CPD Teacher, 2013)



3.2.3 Bridge21 and Subject Teaching

The Bridge21 model is used in the informal learning space on the TCD campus (Oriel House), and has been adopted by teachers in primary and post-primary schools across Ireland. Evidence of this can be seen in Table 18, p. 77 which provides an overview of research into the use of the Bridge21 approach. Analysis of the outcome of this research suggests that students enjoy the experience of working and learning in teams, with students self-reporting that Bridge21 learning experiences create contexts which give students the freedom to direct their learning, explore and develop ideas, create artefacts – helping students develop the skills and the confidence to share and discuss gaps in their learning. Further analysis suggests that while the planning and set up of Bridge21 learning experiences is time intensive, teachers enjoy the experience of changing role to that of facilitator.

Table 18 Bridge21 Pedagogical / Activity Models Applied in a Subject Context	
Subject area	Publications
Student key skills	(Johnston, Conneely, Murchan, & Tangney, 2015)
Peer teaching	(Sullivan, Marshall, & Tangney, 2015)
New Literacies	(Kearney, 2018)
German, Language Acquisition	(Bauer, Devitt, & Tangney, 2015)
History, Contextual Inquiry	(O'Donovan, 2014, 2015; O'Donovan & Kearney, 2015)
STEM	(Bray, 2015; Bray & Tangney, 2013, 2017; Tangney, Boran, Knox, & Bray, 2018; Wickham, Girvan, & Tangney, 2016)
Robotics, Programming, Internet of Things, Hackathons	(Byrne, O'Sullivan, & Sullivan, 2017; Byrne, Sullivan, & O'Sullivan, 2018; Tangney, Oldham, Conneely, Barrett, & Lawlor, 2010)
CPD	(Conneely, 2018; Girvan, Conneely, & Tangney, 2016)

3.3 TCD Post Graduate Certificate in 21st Century Teaching and Learning

The TCD Post Graduate Certificate in 21st Century Teaching and Learning provides professional development to in-service teachers seeking to implement team-based, technology-mediated, cross-curricula projects in Computer Science. The TCD course aims to "*equip in-service teachers with the requisite knowledge, skills and competence to support the development of an innovative learning culture within schools, which is team-based, technology-mediated, project-focused and cross curricular. The course modules aim to enhance the expertise of participant teachers in new models of teaching and learning with particular emphasis on Science Technology Engineering Maths/Computer Science. They also aim to address complex challenges related to developing an inclusive educational environment and preparing all school students for higher academic aspiration and progression, through a focus on whole school culture, leadership and change. It is intended that participant teachers will learn how to develop and lead a 'cultural change process' within the classroom and the wider school community*" (TCD, 2018, p. 1).

This one year certificate programme consists of twelve module options which educators can attend during evenings and weekends (TCD, 2018). Half of the workshop modules cover educational content including leadership and change management, inclusive education, school/classroom based research, with the remainder covering the CS topics of Computational Thinking, Animation and Game Design using the Scratch programming language, Text Based programming using Python, and hardware using the Raspberry Pi (B21, 2018). Table 19, p. 79 provides a full list of all modules.

First offered in September 2014 (TCD, 2014), the part-time course is entering its fifth year of operation and runs over one academic year. The course was designed to assist teachers prepare for teaching the NCCA (2014b) short course in 'Coding' and Digital Media Literacy (NCCA, 2014b), and covers most of the content for the new leaving certificate subject in Computer Science (NCCA, 2018c). All certificate teachers complete one compulsory module in Digital Media (TA21-Mod-1) which introduces teachers to the Bridge21 model and engages teachers in digital tasks, where teachers gain hands on experience of teamwork and constructing digital artefacts. Modules one to six follow the Bridge21 methodology, with each workshop covering one day which enables teachers to experience the full Bridge21 methodology. Attendances of any Modules from TA21-Mod-2 through to TA21-Mod-6 are referred to as the Bridge21 CS CPD programme.

The Bridge21 CS CPD programme is a sub-set of TCD Pg. Cert in 21st Century Teaching and Learning modules. Teachers who are not enrolled can attend computing modules TA21-01 through to TA21-06 on a non-assessed basis. Non-cert teachers experience the same curricula as well as have the opportunity to work with TCD Pg. Cert 21st Century teaching and learning course participants, with both cohorts encouraged to share expertise and collaborate on projects.

3.3.1 Bridge21 CS CPD Programme Details

The Bridge21 CS CPD programme consists of five computing workshops, which teachers are advised to follow in sequence to maximise their learning. Teachers can opt to complete all five computing modules, with each consisting of a workshop, a corresponding assignment support session, an implementation phase and then the construction of an academic report which evaluates the implementation. Teachers can also opt to register to attend, just the workshop component, which gives teachers, the option to collaborate on computing and programming projects. Each workshop covers a different but interlinked content area. Teachers with prior computing expertise or teachers with more advanced computing skills can opt to attend workshops listed later in the sequence after discussion with the workshop co-ordinators. What follows is a description of the computing workshop content, covering their aims and associated learning outcomes for each content area.

Table 19 TCD Pg. Cert 21st Century Teaching and Learning Module List (TCD, 2018)		
Students must select 6 modules from the available suite of twelve module options. Modules comprise of workshops, lectures, online materials and all students must complete TA21-Mod-1: Digital Media. Each module equates to 5 ECTS credits, which includes 100 student effort hours, and covers attendance of workshops, lectures and seminars, pre-module reading and preparation, in course reading, practical implementations in schools and assignments. Face-to-face contact time for each module is 8 hours.		
Digital Media	TA21-Mod-1: Digital Media Literacy and 21st Century Learning	Compulsory
Computer Science	TA21-Mod-2: Problem Solving in the 21st Century (Computational Thinking)	*Optional*
Computer Science	TA21-Mod-3: Introduction to programming (Scratch 1: Introduction & Animation)	*Optional*
Computer Science	TA21-Mod-4: Intermedia programming (Scratch 2: Game design)	*Optional*
Computer Science	TA21-Mod-5: Exploring Computer Systems (Raspberry Pi 1: Introduction)	*Optional*
Computer Science	TA21-Mod-6: Text-based Programming (Python 1: Introduction)	*Optional*
STEM	TA21-Mod-7: Contextualised Mathematics	Optional
STEM	TA21-Mod-8: Science, Technology, Engineering & Mathematics Pedagogy	Optional
Education	TA21-Mod-9: Bridge21 Advanced Methodology: Teacher as Co Researcher	Optional
Education	TA21-Mod-10: Inclusive Education: issues related to equality, diversity and disadvantage in educational settings	Optional
Education	TA21-Mod-11: Leadership & Change Management in Education	Optional
Education	TA21-Mod-12: Information Literacy through Contextualised Inquiry	Optional

3.3.2 TA21-Mod-2: Problem Solving in the 21st Century (Computational Thinking)

Also referred to as Computational Thinking, this first module introduces teachers to practical problem solving strategies as they relate to coding and programming, without the need to use a computer. This module also exposes teachers to concepts of algorithms and algorithmic thinking, and uses offline and online activities to give teachers practice experience in solving problems, which emerge through interacting with computing systems. This module aims to give teachers the building blocks to understand computational processes before moving onto projects, which involve using programming languages. The syllabus covers 'Problem Solving and Computational thinking: Problem solving strategies,' 'Algorithms,' 'Problem solving and computational thinking skills and competencies,' and 'Practical approaches for implementing computational thinking and problem based activities' (TCD, 2017). The following learning outcomes accompany this module. On completion of this module, teachers should be able to: '(1) identify and describe some problem solving strategies; (2) describe and explain some algorithms and algorithmic thinking; (3) solve problems which have more than one possible solution; (4) plan a 21st Century learning experience which incorporates algorithmic thinking & problem solving activities, and (5) critically reflect upon and evaluate the planned learning experience' (TCD, 2017). The Problem Solving in the 21st Century workshop was designed to give teachers a strong grounding in computing terms and problems.

3.3.3 TA21-Mod-3: Introduction to programming (Scratch 1: Introduction & Animation)

This module seeks to introduce teachers to basic programming concepts as well as help teachers to build a practical and foundational understanding of computing and programming content before moving onto more complex programming modules. This module aims to develop teacher's practical programming skills, through working in teams to help teachers develop the confidence to use applications and methods which assist in the development of basic programming skills and apply application of programming skills in a project. The syllabus covers 'basic programming concepts (loops and initialisations), practical introductory technical skills, including basic proficiency with tools such as Scratch for animation and practical approaches for implementing an introductory programming-based learning activity' (TCD, 2017). On completion of this module, teachers should be able to: '(1) plan and implement introductory programming learning activities according to the Bridge21 model of 21st Century learning; (2) identify and illustrate ways in which programming will enliven and enrich their classroom teaching; (3)illustrate an understanding of basic programming concepts such as loops and initialisation; (4) relate basic programming concepts to basic animation actions, and (5) critically reflect upon the planned learning experience'(TCD, 2017). Scratch 1 exposes teachers to visual programming where teams create their own animations (Figure 3-4).

Figure 3-4 Scratch Visual Programming Language Example

3.3.4 TA21-Mod-4: Intermedia programming (Scratch 2: Game design)

This module revisits core concepts examined in Scratch 1 then introduces new and more advanced programming concepts, which teachers will apply in their projects. In this module, teachers develop more advanced computing skills, which are incorporated into the design of games, which support concurrent game play of two or more players. The syllabus covers the intermediate programming concepts including 'variables, events, concurrency, inputs', and 'technical skills, including intermediate proficiency with tools such as Scratch.' This module emphasises a 'practical approach for implementing a learning activity based on game design' (TCD, 2017). On completion of this module teachers should be able to: '(1) plan and implement game design learning activities according to the Bridge21 model of 21st Century learning; (2) Identify and illustrate ways in which programming will enliven and enrich their classroom teaching; (3) create a technical artefact that demonstrates the use of intermediate programming concepts such as variables, and concurrency; (4) relate intermediate programming concepts to basic game design elements; and (5) critically reflect upon the planned learning experience' (TCD, 2017).

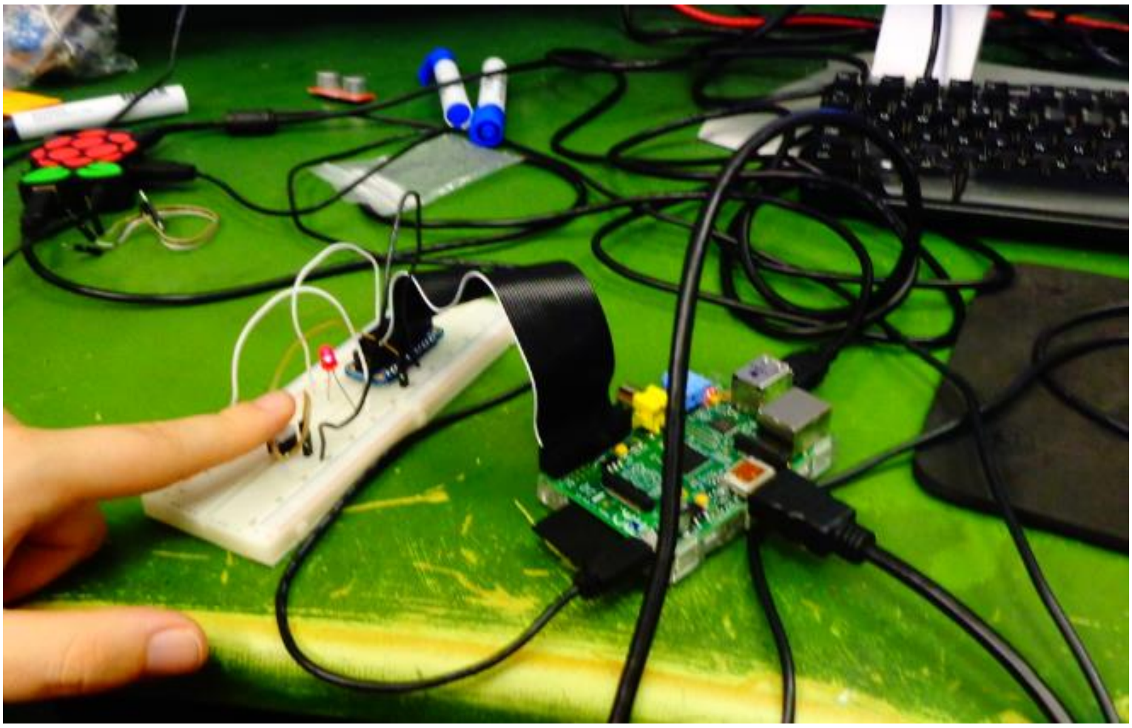
3.3.5 TA21-Mod-5: Exploring Computer Systems (Raspberry Pi 1: Introduction)

This module builds on programming skills explored in Scratch 1 and 2, and extends these concepts into physical computing. This module introduces teachers to the concepts inputs and outputs as they relate to computing and the systems that we use in everyday life. This module also aims to help teachers build a practical understanding of 'embedded' systems, using problem solving and brain storming techniques to encourage the deconstruction and exploration of systems to understand their use in related applications. This particular syllabus takes a deeper look at 'computing in everyday life, with a focus on inputs and outputs; intermediate embedded system skills and competencies (electronics and programming) and practical approaches for implementing learning activities involving embedded systems' (TCD, 2017). On completion of this module teachers should be able to: '(1) plan and implement an embedded systems learning activity according to the Bridge21 model of 21st Century learning; (2) construct basic electronics circuits and code to interface with electronic components; (3) relate their use of embedded systems to real world applications; (4) identify the set up and support requirements of an embedded systems 21st Century learning activity, and (5) critically reflect upon the planned learning experience' (TCD, 2017).

3.3.6 TA21-Mod-6: Text-based Programming (Python 1: Introduction)

The fifth and final workshop in the computing module series is Text-based Programming (Python 1: Introduction). This module combines programming theories from earlier modules (computational thinking, scratch programming, and Raspberry pi) into a project using the Python programming language to activate a switch (Figure 3-5, p. 83). The aim of this module is to introduce teachers to text-based programming, and their use in computing tasks, one of which includes programming a light switch. On completion of the module teachers should be able to: '(1) plan and implement a learning activity using a text based programming language in the context of the Bridge21 model of 21st Century learning; (2) identify and illustrate ways in which programming will enliven and enrich their classroom teaching; (3) create a technical artefact that demonstrates proficiency in the use of a text based programming language; (4) relate programming tasks to real world applications, and (5) critically reflect upon the planned learning experience' (TCD, 2017). This syllabus covers the advanced programming 'concepts including syntax and debugging, advanced technical skills, including proficiency with programming languages such as Python, and practical approaches for implementing a learning activity using a text based programming language' (TCD, 2017).

Figure 3-5 Using the Python programming language to activate a switch



3.4 Summary

Across Ireland, third level universities are putting in place professional development programmes which are designed to assist teachers plan for and respond to the twin challenges of including 21st Century skills into subject teaching and preparing to teach computing courses. Each new computing course (both the short course for lower secondary students and the leaving certificate for teaching at upper secondary) are skills focused, project-based and designed to encourage students to engage in the design, production and construction of computing artefacts. One professional development programme in particular, Trinity College's Dublin Post Graduate Certificate in 21st Century Teaching and Learning offers teachers the combined opportunity to experience a skills focused, team-based, project orientated and technology-mediated approach to teaching which has been adapted for computer science. Based on the patrol system, where students of a similar age work in teams to help each other, lead, and complete tasks, the Bridge21 model provides both a pedagogical process and an activity structure that is designed to help teachers implement a 21st Century approach to teaching. The Bridge21 model has been adapted for teaching and learning computing using computing activities that are constructed to support teachers, as learners, working in teams, learning, creating, developing, and producing computing artefacts.

4 Methodological Approach

The previous chapter covered the background and set up of the Bridge21 CS CPD programme, and set out the learning outcomes for each computing workshop. Each computing module used the Bridge21 model in its delivery, with teachers encouraged to learn computing through collaborating with peers to complete projects and share computing outcomes. The process of teachers reporting on their experiences of collaborative learning in a professional development context provides valuable information highlighting perceived barriers and successes. However, further analysis was required to explore teacher reactions to the workshop content, teacher perceptions of their learning and intentions to use the workshop content, as well as teacher experiences of using the workshop content in school and the results in terms of the impact on teaching. The methodological approach covers the theories, and protocols used to design, organise, and collect data in these contexts.

This chapter describes the theories and methods, which the researcher evaluated and then combined to collect data exploring teacher perceptions and experiences of the Bridge21 CS CPD programme. This chapter also covers the design of instrumentation used to explore teacher perceptions of the CPD, and experiences of using the CPD content in teaching computing in schools. The first section provides a description of the philosophical framework underpinning the research design. The second section introduces the research questions and the third section describes the methodological approach adapted for this research and covers the design of the instrumentation. The remaining three sections describing the protocols that were used to collect data from two contexts and the last section in concludes with limitations.

4.1 Philosophical Framework

Lincoln and Guba (1985) ask "*how can an inquirer persuade his or her audiences that the findings of an inquiry are worth paying attention to, worth talking account of*" (p. 290). The answer lies in the declaration of our philosophical assumptions about the nature of truth, reality and the construction of knowledge. Stating these assumptions helps us to help our audience understand what 'contextual framework' has been used to support the exploration of research claims (W. V. Harris, 2000, p. 110). Defining our perceptions of 'reality, knowledge and the formation of truth statements' also help us to align our world view with the views of others, and our methods and philosophical arguments with broader discussions, in an attempt to 'establish confidence in the truth of our findings' (Thietart & Wauchope, 2001, p. 25).

The following philosophical approach consolidates the researchers' thinking on the construction of reality (ontology) and knowledge (epistemology) as well as the processes that influence the construction of truth claims. Crotty's (1998) framework provided a structure through which to explore relationships between Epistemological Beliefs, Theoretical Position, Methodological Perspective, Methods, Units of Data Analysis and Learning Theory. The rational underpinning the treatment of each construct in the context of this research is covered in the following sections.

4.1.1 Ontology

Cho Choosing an ontological perspective binds the researcher to a particular paradigm (Hesse-Biber & Leavy, 2003). Thus, the researchers' ontological perspective needs to be made explicit, so that subsequent foundational claims of validity can be argued as 'trustworthy' (Denzin, 1997). Once the researcher has developed their ontological perspective on 'reality', this shapes how we then perceive the construction of reality (Berger & Luckmann, 1966). There are a number of ontological paradigms that researchers can chose to explore the social construction of reality (Table 20).

Table 20 Ontological Paradigms Adapted from Guba and Lincoln (2000)

Positivism	Positivists argue that there is an 'objective' reality, which can be 'bounded,' and then measured as an object detached from its context (Hassard, 1993). An 'objective' understanding of reality 'is' perceived as apprehendable and can be reached through constructing hypotheses (Acton, 1951).
Post-Positivism	Post-positivists propose that there is an 'objective' reality but that it is less easily bounded, with measures providing approximations (Guba & Lincoln, 2000).
Critical Theory	Critical theorists suggest that there is no 'objective reality' but that there are subjective reconstructions of reality, and influenced by political, social, cultural, economic, gender values (Cruickshank, 2007).
Interpretivism	Interpretivists adopt the position there is no 'object reality' but that that there are different ways to reconstructing accounts of social reality (Walsham, 1995b).

Lincoln and Guba (2000) suggest that there are four common paradigms, which place different emphasis on the 'treatment' of reality in social research - there are positivism, post-positivism, critical theorists, and interpretivists. Positivists argue that 'reality' is an object which can be studied, explored and explained (Trauth & Jessup, 2000). While post-positivists seek to extend positivist thinking, acknowledging that gathering research evidence is an 'intersubjective' process (Lincoln & Guba, 1985). In contrast, critical theorists, view reality as shaped by a multitude of lenses, which give a distorted and biased view of reality (Geuss, 1981). Similarity, interpretivists propose that there is no objective reality, but rather our perception of social reality is shaped through our experiences and interactions with others in the world (Guba & Lincoln, 2000). The researcher chose an interpretivist perspective which means that "*users of this paradigm are orientated to the production of reconstructed understandings of the social word*" (Guba & Lincoln, 2000, p. 85).

4.1.2 Epistemology

Developing an epistemological perspective means agreeing with objective, subjective or inter-subjective theories which each propose a path to knowledge construction (Davidson, 2001). Mead (1934) continues that learning and the construction of knowledge and meaning is a social process, shaped through our own actions and behaviours, and through our social interactions with others in the world. In keeping with the interpretivist tradition, Walsham (1993) suggests that "*interpretivism is thus an epistemological position, concerned with approaches to the understanding of reality and asserting that all such knowledge is necessarily a social construction and thus subjective*" (p. 6). What this means is that the construction of knowledge is a social process, where "*everyday life presents itself as a reality interpreted by men (and women) and subjectively meaningful to them as a coherent world*" (Berger & Luckmann, 1966, p. 33). The researcher aligns with a subjective view of the social construction of knowledge, which implies that facts are value laden constructs that have been shaped over time and accepted into common use by society (Lynch, 1996).

4.1.3 Methodology

The researcher chose an embedded case study methodology (Yin, 2003), given that it provides a structure to link theory to the research context, the context to a case and the case to units of data analysis. Walsham (1993) cautions that "*from an interpretive position, the validity of an extrapolation from an individual case or cases depends not on the representativeness of such cases in a statistical sense, but on the plausibility and cogency of the logical reasoning used in describing the results from the cases, and in drawing conclusions from them"* (p. 15). However Walsham (1995b) also provides implementation guidelines for applying case study research from within the interpretivist paradigm. Geertz (1988) advises that the motivation for applying methodologies is to aim toward "*making that information available to the professional community in practical form*" (p. 1). The researcher thus chose to implement an embedded case study methodology from the interpretivist tradition, given the need explore the impact that collaborative learning theory had on participants, as well as participants' experience of applying the same theory in their own contexts.

4.1.4 Methods

The researcher chose to implement a mixed method design (Creswell, 2005). A mixed methods design gathers data in two phases; using quantitative instrumentation to gather data to build a more general picture of the problem statement which is complemented by qualitative methods to add definition and context that is used to explain phenomena (Creswell, 2005). An advantage in using this design is that it provides a process for collecting quantitative and qualitative data; however a limitation rests with deciding which data to use in reporting. The researcher addressed this limitation through using a logical model linked to variable tables (Appendix 9.4, 9.5, 9.6, and 9.11).

4.1.5 Units of Data Analysis

Having defined a process for structuring data analysis, units of data analysis were then required to assist with clustering emergent results into themes and statistical outcomes. While working with predefined units of data analysis can be perceived to limit the capacity for generating theory (Glaser & Strauss, 2012), Krippendorff and Block (2009) argue that units can play a core role in helping the researcher cluster the data into themes which are meaningful to the researcher, and guide coding processes. Goetz and LeCompte (1981) argue that clustering data according to units of data analysis, facilitates analytic induction, or a comparative method of reducing data sets into meaningful themes. Thus the researcher adapted each of Kirkpatrick's (1994) four evaluation levels as reaction, learning, behaviours and results units of analysis to support theorisation in CPD and school contexts.

4.1.6 Summary

Underpinning this research, and the following sections, which cover, methods, analysis and the reconstruction of research findings, is an interpretive approach to research. Interpretivist beliefs influence the claims that can be made about the formation of social reality, the reconstruction of knowledge, and the processes that are used to gather and reconstruct phenomena in the world. Interpretivist beliefs also inform the use of models and frameworks that are used to that provide new evidence and insight into social processes. Having declared as an interpretivist, the researcher draws from relevant literature to develop methods and instruments that will be used to gather data in order to attempt to reconstruct participants' perceptions of the world. Further underpinning the administration of the research instruments and analytical methods is Vygotskian (1978) theory. Vygotskian (1978) theory provides a lens through which to explore peer collaboration in a professional development context, and to attempt to form new insights into the role that the Bridge21 model plays in assisting teachers construct the expertise they need to teach computing.

4.2 Research Questions

As discussed in detail in chapter 2, teachers need assistance in developing both content and pedagogic knowledge in order to teach computing in schools. The Bridge21 CS CPD programme is designed to support teachers seeking to upskill themselves in this area and help teachers develop methods to design lessons for teaching new NCCA computing curricula. The adaptation of the Bridge21 model as a CS CPD method creates a unique opportunity to understand what impact a collaborative approach to professional learning plays in preparing teachers to teach computing. Two research questions and five sub-questions explore teacher perceptions and experiences of collaborative learning and are listed below. These questions are repeated in Table 1, p. 5.

The first research question explores:

> Q1: What are teachers' perceptions of the Bridge21 model as a method of CPD?

Three further sub-questions examine:

> Q1.1 what are teachers' reactions to the CPD workshop content?
>
> Q1.2 what content knowledge did teachers learn?
>
> Q1.3 what strategies did teachers' intend using for teaching computing?

A second research question then investigates:

> Q2: What are teachers' experiences of using the Bridge21 model to teach computing?

A further sub-question considers:

> Q2.1 what elements of the Bridge21 model did teachers identify as most relevant for teaching computing in practice?

The following methodological approach was designed to answer these research questions.

4.3 Methodological Approach

The methodological approach follows Yin's (2003) case study protocol (section 4.4) which describes the theories and processes used to collect data to address the research questions. Also adapted for this research are Yin's (2003) principles of data collection (section 4.5). The rational for adapting these protocols and principles is that they each provide the capacity to structure a methodological discussion, which is obtained from a single case study. This methodological approach also takes account of an interpretivist approach to case study methods (Walsham, 1993) and includes the use of a mixed method design to guide data collection and the linking of data sets (Creswell, 2005).

4.3.1 Interpretivist Case Study Research

One of the problems in conducting case study research is difficulty in organising the order or sequence in which phenomena occur in research contexts. Yin (2003) offers a solution to this problem by suggesting that researchers use a case study protocol and data collection principles to organise case study phenomena. Yin's (2003) case study protocol provides a procedure for conducting and organising case study research, offering a "*major way of increasing the reliability of case study research and is intended to guide the investigator in carrying out the data collection from a single case study*" (p. 67).

There are a number of limitations to using a case study methodology from an interpretivist perspective, which views phenomena as contextual, and concepts as subjective and interlinked. Walsham (1993) cautions that "*from an interpretive position, the validity of an extrapolation from an individual case or cases depends not on the representativeness of such cases in a statistical sense, but on the plausibility and cogency of the logical reasoning used in describing the results from the cases, and in drawing conclusions from them"* (p. 15). A case study can help to organise concepts into a structure, but there are still inherent problems in 'abstracting' meaning from collected data. Thus, clarification of the role of interpretivist theory in conducting case study research is required before completing Yin's (2003) protocols. For the interpretivist researcher, a case study is a framework through which to explore 'how' and 'why' phenomena occurred but also 'what' impact, the phenomena had on theorisation.

The researcher aligned with the interpretivist tradition, given the belief that our perceptions of reality are in part socially constructed, which means that research methods, including case study research, generate approximations of phenomena. What this means is that there are "*no correct or incorrect theories*" (Walsham, 1993, p. 6), but that there are many different ways of exploring phenomena in the world. The implications of aligning with this philosophical view is significant in that there "*is no objective reality which can be discovered by researchers and replicated by others, in contrast to the assumptions of positivist science*" (Walsham, 1993, p. 5).

To add some level of objectivity into the interpretivist process, Walsham (1993) further argues for using structures such as protocols and principles to organise case study research, clarifying the steps taken to conduct research and develop theoretical findings. Indeed, Walsham continues that the protocols are particularly important in case study research given the outcome of generating a written account which enables *"broader* *judgements to be made*" (1993, p. 6). The following section provides an overview of research designs used within the interpretivist domain and considers the use of a case study approach, against these other designs.

4.3.2 Justification for the Research Design

The research design provides a road map, setting out the different ways to collect qualitative and quantitative data (Yin, 2003). Research designs provide guidelines for choosing how to collect, analyse, and report on research findings (Creswell, 2005). Within the interpretivist paradigm, there are a number of different research designs that the researcher can choose to structure their research (Walsham, 1995a). Common designs used within the interpretivist domain include grounded theory, ethnography, action research, phenomenology, and case studies (Table 21).

Table 21 Interpretivist Research Designs adapted from Creswell (2005)

Design	Rationale
Grounded Theory	Developing theory driven from data analysis (De Villiers, 2005).
Ethnographic	Describing and interpreting social phenomena (Myers, 1999).
Action Research	Iterative development of theory to change practice (Baskerville, 1999).
Phenomenology	Exploration of the 'lived experience' (Mingers, 1992).
Case Study	Exploring theory within a particular context (Walsham, 1995b).

Grounded theory seeks to build new theoretical claims through the analysis of data (De Villiers, 2005), while ethnographic research includes descriptions and interpretations of observed phenomena (Myers, 1999). Action research provides an iterative design, with results shaping the design of further layers of research and reflection (Baskerville, 1999). In contrast to this, phenomenology, seeks to reach a deeper understanding of the 'lived' experience (Mingers, 1992), while case studies provide frameworks to bound phenomena within a context (Walsham, 1995b).

Compared to other designs, case studies enable the researcher to explore theory as it applies to a particular context. A case study design enables the researcher to link the research questions to a context, case and units of analysis (Yin, 2003). Furthermore, this mapping protocol enables the researcher to explore the impact of theory on a particular group, or individual (Wells, Hirshberg, Lipton, & Oakes, 2002). A limitation with case study research is that too much data can be collected, and that researchers are presented with the challenge of organising their data and the results into a structure which corresponds with the research questions (Eisenhardt, 1989). The researcher chose to implement a case study design because it provided a framework enabling the researcher to link the research questions with the research context, the case and units of data analysis, and focus on the use of the Bridge21 model as it was applied in a CPD and school context.

4.3.3 Single Case and Embedded Case Study Designs

There are two types of case study designs that are commonly used for exploring social phenomena (Yin, 2003). The first is a 'holistic' or single-case study design. A single-case design places emphasis on exploring a unique or critical instance of phenomenon, which may also be representative, revelatory, or longitudinal (Yin, 2003). However a limitation with the single-case designs is the lack of comparison data which can be used to support theoretical abstraction and generalisations (Yin, 2003). Yin advises that single-case designs are suited to longitudinal work, in so far as they provide a frame in which to examine change in phenomena over time. However, this can also mean that the scope of the research can change thus impacting upon the administration of research methods. Therefore, a process is required to ensure that single-case design research complies with its original aims and objectives. Yin (2003) acknowledges this limitation and argues for using a protocol as a way to address subjectivity , with protocols recommended to increase the 'reliability' of a case.

The second type of case study design is an 'embedded' design, which explores more than one 'unit of analysis.' In this instance, the case may explore a particular context, but the sub-units provide a mechanism to drill down into the context and explore underlying theories (Yin, 2003). A limitation with embedded designs is the difficulty in ensuring that sub-units remain conceptually connected to the case and the wider context as more granular analysis develops. An embedded design focuses on social phenomena which occurs within one context (Yin, 2003). An embedded design also facilitates conceptual linkage between quantitative and qualitative data sets, with the units of data analysis providing the capacity to explore specific phenomena related to the case study.

The researcher choose to implement an embedded design to explore the use of collaborative learning theory as it is used in the general context of teaching and learning computing content, and as it was applied within a case setting. The case was defined as the Bridge21 CS CPD programme, and units of data analysis examined teacher reactions, learning, behaviours and results in terms of CPD impact. The selection of an embedded design created the opportunity for this research to be compared with other Bridge21 case studies and evaluations which used a comparable level structure in their analysis similar to that of

Kirkpatrick (1994) and Guskey (2000).The following section sets out the protocols which were followed to implement the embedded case study design.

4.4 Applying Yin's (2003) Case Study Protocol

Yin (2003) states that an case study protocol should contain five sections:

(1) A brief description of the case study project.
(2) A description of general field procedures including an overview of the processes used to gain access to the research context and permissions to conduct research.
(3) The case study questions and an overview of the framework used to gather data to address the questions
(4) A description of the protocols followed to establish validly and report on the results.
(5) A report on the outcome of completed pilot case study research.

The remainder of this section describes each stage of Yin's protocol, which have been adapted for this research.

4.4.1 Section (1) – Description of the Case Study Project

This section covers background information about the project, including the rationale for conducting the study, and clarification of the research context, case and each unit of analysis as they relate to collaborative learning theory. Also explored is the researchers' rationale for opting to complete this research and to conduct research in the Bridge21 research lab. Furthermore, the rationale for conducting this research is discussed in light of concurrent research within the Bridge21 research lab which is exploring the application of the Bridge21 model as an approach to enhance subject teaching across post-primary education and related professional development programmes.

The genesis of this research came from the TCD requirement to explore the long term impact of the Bridge21 model on teacher learning, including on the teachers capacity to lead 'transformation change within the classroom' (CRITE, 2013). The computing modules would follow the Bridge21 methodology, with workshops delivered in the Bridge21 research lab. Each one-day computing workshop would provide an average of 8 hours of CPD over the course of a single day.

This research was commissioned by the Centre for Research into I.T. in Education (CRITE), in collaboration with the School of Computer Science and Statistics and the School of Education, in Trinity College Dublin. The commission involved the design of a framework to explore the impact that the Bridge21 model had on professional learning, and focus on collaborative methods in teaching and learning computing. Data collection spanned five academic years involving *N* = 1,215 CPD teachers and *N* = 385 in-service teachers (Table 22).

Table 22 CPD and School Samples

CS CPD Sample and Time Scale	Year 1	Year 2	Year 3	Year 4	Year 5	Total
	2013/2014	2014/2015	2015/2016	2016/2017	2017/2018	
CPD Sample	211	212	223	321	248	1,215
Pre-workshop Questionnaire Responses	-	-	-	212	81	293
Post-Workshop Questionnaire Responses	211	176	157	192	83	819
Pre and Post Questionnaire Responses Combined Sub Total	211	176	157	404	164	1,112
Field Notes (Pages)	63	29	17	-	-	109
School Sample	73	97	63	95	57	385
Post CPD Questionnaire Responses	10	16	8	15	15	64

The researcher was involved in the delivery of *N* = 72 one day computing workshops during this period, in the role of mentor and researcher, accruing approximately 600 hours of interaction time with teachers in CPD workshops. Teachers were invited to complete two questionnaires per workshop, so responses are not unique and include multiple responses from the same participants.

A total of *N* = 293 teachers completed the pre-workshop questionnaire, with low numbers reported due to phasing this instrument into the workshop data collection model at the start of the 2016/2017 academic year. In comparison, post-workshop questionnaire completions total *N* = 819, with teacher responses from each year of the CPD included in the post-workshop data set. The CPD sample also included field notes data gathered by the researcher during the first three years of CPD workshop analysis. The researcher generated *N* = 109 pages of full text notes, which added further context to the questionnaire data sets. The response rate for pre-workshop questionnaires (*N* = 293) was 24%, while the response rate for post-workshop questionnaires (*N* = 819) was 67%.

Finally, the teaching computing in school data set is smaller than the CPD workshop data set, given that teachers had left the CPD, and returned to the workplace when the questionnaire was administered. While the response rate is only 17%, the sample includes responses from teachers attending workshops at different times over all five years, capturing range and depth of teaching experience.

Having described the CPD and school samples, the following sections clarify the links between theory, context, case, and units of data analysis addressed through the embedded design.

(1) Theory

The *theory* under investigation is peer collaboration in a 21st Century context, expressed by Vygotsky (1978) as ZPD theory, and applied in CPD and school contexts through Bridge21 CPD.

(2) Context

The *context* covers the introduction of computing into schools in Ireland and is defined as the professional challenges facing teachers embarking on the process of preparing to teach computing, including programming, with teachers facing difficulties with learning the relevant content knowledge as well as the appropriate methods for teaching computing.

(3) Case

The *case* is described as a cohort of professional teachers attending the Bridge21 CS CPD programme, and each of its constituent computing workshop modules, covering computational thinking, scratch programming, text programming, and hardware configuration modules. This case study focuses on data collection, which occurred within computing workshops and schools, with evaluation forms used to explore teacher perceptions and experiences of Bridge21 CPD workshops.

(4) Units

Each of Kirkpatrick's (1994) evaluation levels were adapted as *units of data analysis*; with Level 1 – reactions, adapted as teacher reactions to the computing workshops; Level 2 – adapted as teacher perception of their learning; Level 3 – adapted as teachers intention to use elements of the computing workshop content and methods in their teaching; Level 4 – was adapted as results, in terms of elements of the workshop, and examples that teachers used for teaching computing.

Level 5 is not addressed, given that the researcher did not have direct access to schools. To address this limitation, the researcher has included examples from teachers reporting on their experiences of using the Bridge21 model and impact on learning and student engagement.

4.4.2 Section (2) – Field Work Procedures

There are a number of steps to follow to ensure that data collected from case study research follows ethical procedures and guidelines that are required to safeguard participants. The research took place within a University context, thus the researcher completed research documentation in line with TCD ethical procedures. This involved the design of a formal ethics application, the submission of Garda/Police vetting and the design of research instrumentation that would be administered to teacher participants of the CPD programme.

(1) Ethics / Garda Vetting and Data Protection

All relevant ethical approval was granted by the School of Computer Science and Statistics research ethics board prior to commencing data collection in January 2014. Samples of the documentation that were included in this application and were used with participants are provided in Appendix 9.1 and 9.2. All printed research materials were stored in a locked filing cabinet, which also included paper-based copies of electronic data. This repository contained data by module and date, to facilitate data retrieval from an archive spanning five consecutive years. Electronic data was stored in a password-protected database on the researcher's university computer in a locked office.

(2) Teacher Samples

The researcher had no control over who attended workshops or who would complete research questionnaires. Teachers opted to attend computing workshops of their own accord, thus workshop and school samples are defined as 'self-selecting' or 'opportunist' (Miles & Huberman, 1994). All research participants in CPD and school contexts were adults, over the age of 18 years.

(3) Procedure for Gathering Consent

The researcher designed a data collection procedure to run in tandem with each computing workshop. At the start of each workshop, the researcher presented a slide, which explored the aims and objectives of the research process. During this short 5-minute presentation the researcher invited teachers to ask questions about the research and used this time to report back tentative research results to teachers, as a way to ‘member check results’ and explore areas for further analysis. The researcher used this opportunity to distribute ethics and consent forms to each teacher. The researcher operated and ‘opt in and opt out’ ethics procedure. Teachers, who wanted to opt into the research process and consent to have their individual data, aggregated into evaluation metrics, signed consent forms. Unsigned forms indicated that teachers had opted out of the research process. Thus, data without a corresponding ethics consent form were not processed. The analysis of ethics forms, confirms an 83% opt-in rate, with the remaining 17% opting-out.

(4) Anonymity

All participant data was anonymised, with a coding system developed for reporting. Each participant was assigned either a CSCPD or a TCIN code. The CSCPD code refers to Computer Science Continuing Professional Development, and the TCIN code refers to Teaching Computing in Schools. Participants were allocated either code to denote the context in which the data was collected. A numeric value was then added to each code to distinguish between participant responses and contexts.

(5) Pre-workshop Questionnaire

Following the research presentation and collection of consent forms, teachers were invited to complete a short ‘pre-questionnaire’, exploring their preconceptions of the computing workshop model learning outcomes (Appendix 9.7). Forms were customised to reflect the content of the one-day computing workshop that teachers attended. This form took 5 minutes to complete, and provided an indicator of teachers’ perceptions of their ability to complete the tasks and engage with the content of that workshop. Form completion was optional.

(6) Researcher / Mentoring Role

The researcher played the role of ‘mentor’ in the CPD, which was a supporting role to the lead facilitator. This role involved liaising with teams during computing activities, assisting teachers with problem solving tasks, and using techniques such as ‘Socratic’ questioning to encourage teams, and individuals within teams, to work through problems before giving an explicit answer. The mentoring role also involved assisting teams with trouble-shooting technical problems as well as helping teams work though design issues emerging from project work and computer programming.

(7) Post-workshop Questionnaire

As the teachers had already completed ethics documentation, they were not required to complete further consent forms at the end of the one-day workshop when invited to complete the post- questionnaire at the end of the one-day workshop. This questionnaire followed the same format as the pre-workshop questionnaire (Appendix 9.8), with the initial section covering workshop learning outcome variables and a subsequent second section inviting teachers to 'react' to further variables exploring the workshop content as well as the teaching methods and teacher intentions to use the CPD content and methods in a school context (Appendix 9.9). Form completion was optional.

(8) Researcher Reflection

At the end of each computing workshop the researcher drafted field notes and asked permission from facilitators to take pictures of artefacts left behind from that days' workshop. This data source proved useful as an 'aid memoire' when reflecting upon workshops (Argyris & Schön, 1974). In addition, the researcher used a desk in the Centre for Research into I.T. in Education as a space to write up field notes and compile reports. This writing space proved invaluable as a place to write up field notes after a day's teaching or as a place to share research ideas with other researchers working on different aspects of the Bridge21 programme.

(9) Field Notes

The inclusion of field notes captured the researchers' observations as well as comments provided during focus groups. Appendix 9.10 provides the template used to structure field notes, and examples of the data formats and the files, collected, analysed, and then described in the researchers' field notes.

(10) Post CPD Questionnaire

Workshop ethics forms covered permission for the researcher to contact teachers who had attended computing workshops, with a link to a post CPD questionnaire that explored their experiences of teaching computing in schools (Appendix 9.12). The post CPD sample contained teachers who had completed at least one computing workshop. This sample is classified as 'self-selecting' or 'opportunist' (Miles & Huberman, 1994) with teachers receiving an email inviting them to complete the questionnaire at their convenience. On receipt of a response in the questionnaire database, the researcher acknowledged the response then sent a follow up summary of the workshop research results inviting feedback. This member checking procedure enabled teachers to comment on results and acted as a member checking procedure. Questionnaire completion was optional.

4.4.3 Section (3) – Case Study Questions

This phase of the protocol seeks to establish clarity between the research questions and methods used to address the research questions. The rationale for using a 'table shell' (Miles & Huberman, 1994) is to map links between the research questions, units of data analysis and the instruments that are used to obtain each data set. On Yin's (2003) recommendation, the researcher adapted Miles and Huberman (1994) table shell concept to map connections between the research questions, the units of data analysis, the research contexts, and each of the research instruments, as well as the method type used in field work (Table 23).

Table 23 Table Shell Linking Research Questions to Data Sets (Miles and Huberman, 1994)

Research Questions and Sub Questions	Unit of Data Analysis	Instrument and Method Type
CPD Context		
Q1: What are teachers' perceptions of the Bridge21 model as a method of CPD?		
Q1.1 What are teachers' reactions to the CPD workshop content?	Reactions	Post-Workshop Questionnaire, Mixed Methods, 1 scale, 1 question, field notes adding context.
Q1.2 What content knowledge did teachers' learn?	Learning	Pre and Post-Workshop Questionnaire, Mixed Methods, 1 scale, 1 question, field notes adding context.
Q1.3 What strategies did teachers' intend using for teaching computing?	Behavioural Intention	Post-Workshop Questionnaire, Mixed Methods, 1 scale, 1 question.
Field Notes	Context	Researcher reflections, literature analysis and artefact descriptions, with field notes adding context.
School Context		
Q2: What are teachers' experiences of using the Bridge21 model to teach computing		
Q2.1 What elements of the Bridge21 model did teachers' identify as most relevant for teaching computing in practice?	Results	Post CPD Questionnaire, Mixed Methods, 4 scales, 1 question.

Yin (2003) advises that table shells "*force you to identify exactly what data are being sought*" (p. 75), thus helping the researcher to identify appropriate fields of literature to explore and to design appropriate research instruments that will enable them to find the data required to address the research questions. Table shells therefore are perceived to provide a linkage between theory, and the methods that are used to explore the practical application of the theory in social contexts.

What follows is a description of the literature informing the design of research instruments mapped to a unit of data analysis that has been adapted for use in this research.

(1) Teacher Perceptions Scales and Questions (Appendix 9.4, 9.5 and 9.6)

This section describes instruments that were used to explore teachers' perceptions of CPD by way of examining their *reactions* to the content, their *perceptions* of their learning, and *behavioural intentions* to use the workshop content in teaching.

The researcher followed Oppenheim's (2000)and Vagias's (2006) commentary on the use of Likert scales used in questionnaire design. The researcher opted to use a 7 point Likert scale across all reaction, learning and intention CPD variables (Dawes, 2008). The 7 point graduated Likert scale was arranged 1 = Strongly agree to 7 = Strongly disagree. A seven point scale was selected to increase choice, with a view to clustering responses around either a very strong or a very weak response, thus giving a clearer indication of a positive or a negative response (Symonds, 1924). This Likert scale arrangement was used across all perception scales administered in Bridge21 CPD workshops.

Demographic makers and three scales were developed to explore the following constructs[3]:

a) Demographics explore teacher age profiles attending the CPD workshops, with further analysis exploring the number of workshop modules provided during the research period. The following sections outline the scales and questions that were used in a CPD context.
b) Teachers' *reactions* to the CPD. This mixed methods instrument was adapted from an existing Kirkpatrick instrument (Kristiansen, 2007) and informed by the results of pilot data. Irrelevant questions were identified through piloting and were removed from the questionnaire. Appendix 9.4 provides the variable table for this scale.
c) Teachers' *content knowledge*. A learning instrument to gauge the level of content knowledge of the participants was developed from the module learning outcomes specified in the Post Graduate Certificate Handbook (TCD, 2017), with variations for

[3] *For drafts of the CPD instruments see: Fisher, Byrne, & Tangney (2015a); Fisher et al., (2016).*

each module area. The researcher adapted the learning outcome text for each computing workshop module into new variables. Open questions were adapted from Bridge21 reflection template and was included in order to explore what teachers might do differently having attended the CPD. Appendix 9.6 provides the variable table used for the pre and the post scale.

d) Teachers' *intentions*. The intention scale and question were adapted from an existing Kirkpatrick (2007) instrument, and explores the extent to which participants intend to apply what they have learned in a training setting in the context of their work. This instrument explored the extent to which the CPD met with participants 'expectations', and their intentions to use the content of the workshops. One open question was adapted from Kirkpatrick (2007). Appendix 9.5 provides the variable table for this scale.

e) Field notes explored teacher engagement with the CPD content and methods. Templates were adapted from Emerson (1995), with Appendix 9.10 providing examples.

(2) Teacher Experience Scales and Questions (Appendix 9.11 and Appendix 9.12)

This section describes the research instruments that explore teachers' experiences of implementing the CPD. The instruments are made up of scales and questions that explore teachers' use of the Bridge21 model, as well exploring the impact of the CPD on the practice of teaching computing.

Again, the researcher followed Oppenheim's (2000) commentary on the use of Likert scales to guide the design of scales used in workshop questionnaires. In this instance, the researcher opted to use a 5-point Likert scale across behaviour and result scales. A five-point scale was selected to map in with other Bridge21 instrumentation, ensuring consistency across instrument sets exploring teaching computing in schools. This questionnaire was developed in collaboration with Dr. Katriona O'Sullivan[4] from Trinity Access 21, who oversaw the development and design of the teaching computing in schools questionnaire. Earlier iterations of the teaching computing in schools questionnaire were piloted with teachers attending CPD workshops to test questions, provide feedback and refine constructs in consultation with teachers. The final version of the teaching computing in schools questionnaire consisted of the following scales and questions.

[4] *For proof of concept see Fisher, O'Sullivan, Tangney, and Byrne (2017).*

a) *Demographics and previous experience*. Following an introduction covering ethics this instrument was tailored to probe teachers' experience of teaching computing in schools. Demographics captured primary and secondary profiles and the number of workshops attended by participants. A frequency of teaching computing scale explored when during school, computing was taught in the curricula and was adapted from Israel et al (2015). Appendix 9.11.5 provides the variable scale exploring teaching computing in the curriculum.

b) *Bridge21 elements*.

    i. The first Bridge21 scale was adapted from the descriptions of each element of the Bridge21 model provided in (Lawlor et al., 2018; Lawlor, Conneely, & Tangney, 2013; Lawlor et al., 2016). These publications provided an operational description of the Bridge21 model that could be converted into variables. Appendix 9.11.1 provides the variable table for this scale.

    ii. The second Bridge21 scale explored barriers to the use of the Bridge21 in a school context. This scale was adapted from a pre-existing Kirkpatrick (2007) instrument to explore perceived barriers to implementing training programmes in work contexts. Appendix 9.11.2 provides the variable table for this scale.

    iii. A third scale was adapted from Sentance and Csizmadia (2017a) to explore other methods used in teaching computing which was added as a comparison to Bridge21. Appendix 9.11.3 provides the variable table for this scale.

    iv. A fourth scale investigated the elements of the Bridge21 model with which teachers perceived they required further support. This scale was adapted from earlier Bridge21 CPD research (Conneely, Girvan, & Tangney, 2012). These variables explored teacher requests for further CPD in using elements of the Bridge21 model. Appendix 9.11.4 provides the variable table for this scale.

    v. A further question adapted from an existing Kirkpatrick instrument (Kristiansen, 2007) to explore suggestions to further enhance the CPD programme, with teachers also reporting in their experience of teaching students, reflecting on the use of the Bridge21 model as a method for teaching CS, and impact on student engagement. Appendix 9.11.6 describes the questions used in this analysis.

4.4.4 Section (4) – Guidelines for Case Study Report

A guideline for reporting provides a structure that enables the researcher to communicate the results of case study research in a logical sequence. There are different ways that researchers can present their results: using a timeline; clustering results depending on the type of instrumentation that is used to gather the data; discussing results that relate to particular themes; or organising the presentation of results as they relate to the research questions (Miles & Huberman, 1994). Yin (2003) suggests defining reporting parameters in the case study protocol to clarify the structure that will be used to present the results, and to highlight areas where data is missing, or where further research is required to address a particular gap. For case study research, Yin (2003) recommends using a 'linear-analytical' reporting style, which involves explaining the steps followed to collect data, as well as reporting on limitations with the process. The researcher used a linear-analytical reporting style proposed by Yin (2003) which sets out the steps taken to implement the research design as well as the choices that are made to design the research instruments that are used to collect data.

While a linear-analytical reporting style follows a sequence recommended for empirical research the researcher used a realist approach (rather than impressionist or confessional representations) to oversee the inclusion and subsequent representation of participant voices within the linear-analytical process (Van Maanen, 1988). A realist style attempts to include verbatim excerpts of reconstructed voices within texts as a way to give 'voice' to participants, which reflect their lived experiences of phenomena. Reports that use a realist approach are designed to give 'presence' to participants and their self-reported experiences. A limitation with realist reconstructions is that participant quotes can be used out of context to justify a particular point of view or theoretical claim (Miles & Huberman, 1994). Thus, it is the researchers' responsibility to render realist accounts with care through the provision of sufficient holistic descriptions to describe the context under investigation and the methodology used to select and represent these 'voices' in an appropriate context. The researcher used pattern coding (LeCompte & Schensul, 1999) as a way to cluster together participant voices to create a context to tell the story of their experiences.

4.4.5 Section (5) – Pilot Case Study

I Initial versions of the *Workshop* and *Teaching Computing in Schools* questionnaires were piloted, and analysed before administering for formal data collection beginning in January 2014 (Table 24, p. 103).

Table 24 Piloting CPD and School Questionnaires					
CS CPD Sample and Time Scale	Year 1	Year 2	Year 3	Year 4	Responses
	2013/2014	2014/2015	2015/2016	2016/2017	
CPD Questionnaire	23				23
School Questionnaire	14	5	21	34	74

A first draft of the *workshop* questionnaire was piloted with teachers attending one of three introductory computing workshops. A sample of *N* = 23 teachers opted to complete the pilot workshop questionnaires, from a self-selecting sample of *N* = 41 teachers attending workshops in the Bridge21 laboratory in the last quarter of 2013. Piloting led to the development of variable tables (Appendix 9.4, 9.5, 9.6). Pilot questionnaire results were omitted from formal data analysis.

A first pilot draft of the teaching computing in schools questionnaire was circulated with a self-selecting sample of *N* = 14 teachers attending introductory computing workshops in Year 1 (2013/2014) to explore Bridge21 variables. Analysis of these variables fed into the development of a second version of the questionnaire which was completed by self-selecting sample of *N* = 5 former Year 2 CPD participants (2014/2015) exploring the addition of programming constructs. Further analysis led to the development of a third version of the questionnaire completed by another self-selecting sample of *N* = 21 former Year 3 CPD participants (2015/2016) exploring the addition of learning outcomes and barrier constructs. A fourth iteration included teaching methods and CPD constructs, and was piloted with a further self-selecting sample of *N* = 34 former Year 4 CPD participants (2016/2017). Thus the final, or fifth version, included tested scales and questions, with the variables tested by *N* = 74 participants (Appendix 9.12. provides a copy of the final version).

All piloting samples were 'opportunistic' (Miles & Huberman, 1994) and consisted of teachers who were asked on a collegial basis to complete questionnaires during workshops. The results were used to refine questions used in the final version the teaching computing in schools questionnaire which was issued to a retrospective sample of *N* = 385 CPD teachers starting from October 2017.

4.5 Applying Yin's (2003) Data Collection Principles

The previous section described the instruments that would be used to explore teachers' perceptions and experiences of the Bridge21 CS CPD programme and the literature that underpinned their development. What has not been discussed are the implications of choosing to use scales and questions that collect quantitative and qualitative data. The following section applies Yin's (2003) principles of data collection to explore the implications of designing research methods which require teachers to answer questions using text and numbers.

4.5.1 Principle (1) – Multiple Sources of Evidence

The first principle of data collection, according to Yin (2003) involves setting out the rational for the type of methods that are used to collect case study data. Yin further suggests that the collection and reporting of a single source of evidence ill advised in case study research). The use of a variety of data sources provides the researcher with an opportunity for triangulation through "*the development of converging lines of inquiry*" (Yin, 2003, p. 98). A mixed methods approach to data collection is described as a process, which involves the capture of numeric and text data. The perceived benefit of collecting two different types of data at the same time is the capacity to provide clarity, where one source may lack 'accuracy' (Creswell, 1998, p.193). However, it is important to understand that the order in which mixed methods data sources are collected has an impact; the results "*do not lie innocently in the world; rather, they are themselves constituted by an interpretive act*" (Fish, 1980, p. 13). For example, Miles and Huberman (1994) suggest that if we lead with quantitative methods, followed by qualitative methods, this process is perceived as giving depth. However, if we lead with qualitative methods, followed by quantitative methods, this is perceived to give precision. It is important to acknowledge the limitations in selecting one method over another. Owing to the size of this study, participants were provided with the quantitative data first, followed by qualitative responses, which were, designed to give depth and granularity to their feedback.

Triangulation is a common method of ascertaining validity in mixed methods research: "*many experts indicate that triangulation characteristically depends on the convergence of data gathered by different methods*" (Ely, Anzul, Friedman, Garner, & McCormack-Steinmetz, 1991, p. 97). The process of triangulation helps to tease out "*inconsistencies and contradictions [that] may help us to refine and revise our framework and findings*" (Ely et al., 1991, p. 98). Triangulation also allows data to be merged at specific time points in the research process, prompting the researcher to explore connections between qualitative and quantitative evidence in order to reveal new connections between the results. Miles and Huberman (1994) further advise that the order in which link we together data sets, can influence or impact the type of results that we produce.

4.5.1.1 Implementing a Mixed Methods Strategy

The researcher chose to implement a mixed methods strategy, with quantitative results informing the design of questions to give deeper insight and meaning to statistical results gathered in CPD and school contexts. The administration of mixed methods (also called an explanatory design) gathers data in two phases: using quantitative instrumentation to gather data and build a more general 'picture of the problem statement' (Creswell, 2005) followed by qualitative methods to add further definition and context that is used to explain phenomena (Ivankova et al., 2006). This two-layered approach to data collection enables the researcher to 'triangulate' the results as they emerge from fieldwork, with the researcher assessing the impact of the statistics on the outcome of qualitative results. This process involves the researcher then recalibrating statistical analysis and devising further questions to reach a more explicit understanding of phenomena. However, while an explanatory design provided the capacity to administer quantitative questions to identify areas for deeper analysis the researcher was cognisant of the limitation that "*as researchers, are part of the world that we are researching, and we cannot be completely objective about that, hence other people's perspectives are equally valid as our own*" (Cohen, Manion, & Morrison, 2000, p. 106).

4.5.1.2 Theories for Linking Data Sets

There is some debate as to the best way to link quantitative and qualitative data sets, particularly in relation to the order in which each data set is processed. Miles and Huberman (1994) propose four designs which researchers can use to link data sets. The first design involves conducting field work which includes the continuous and integrated collection of quantitative and qualitative data sets over time to address the "*the case at hand*" (p. 41). The second design follows a multi-wave approach, where there are gaps between the combined collection of quantitative and qualitative data sets. The third design follows a linear path, with qualitative data sets gathered first to shed light on new or emerging phenomena, which is then followed by more focused quantitative analysis, which is shaped by the qualitative results, with a further wave of qualitative data gathering conducted to add further context to the results. The fourth design reverses the process followed in the third design, with quantitative results gathered first, followed by qualitative analysis to add context to quantitative results, with further quantitative research seeking to test phenomena.

(1) 'Multi-Wave' Based Method

Each of the designs proposed by Miles and Huberman (1994) place different emphasis on the theories, models and processes that are used to generate research results, with are covered in the previous sections. The researcher followed the second the 'wave' design proposed by

Miles and Huberman (1994) given that the researcher had intermittent access to the research site, but the opportunity to collect multiple instances of the same data at specific time periods over time.

(2) Teacher Interview Data

Walsham (1993) reports that it is not always possible to obtain all the data that we seek during the research process. One way to address this limitation is to "*involve a wide variety of stakeholders groups"* in the research process and to use a research design which explores the *"concerns, issues and values of stakeholders*" (Walsham, 1993, p. 179). A limitation with the design used in this research is a lack of teacher interviews to explore teacher experiences of using the Bridge21 model in a school context. While the researcher talked to teachers in an informal capacity during each of the one-day computing workshops in the research sample (*N* = 1,215), formal interviews were difficult for the researcher to organise due to time restrictions in the researchers' data collection schedule. The researcher acknowledges that the omission of teacher interview data limits theorising on teacher use of the Bridge21 model for teaching computing in schools. To address this limitation, the researcher used her position as a mentor in the CPD workshops to talk to teachers about their experiences of implementing the Bridge21 model and used these responses to add context to the teaching computing in schools data set. Finally, Walsham (1993) reminds us that it is important to document the research journey and to acknowledge limitations, including unintended consequences, as well as gaps and problems encountered during the research process. To conclude, Walsham (1993) advises that it is research is complex and it is important to learn from each research experience and encounter, and that "*everybody is a learner during the evaluation process*" (p. 185).

4.5.2 Principle (2) – Create a Case Study Database

The second principle of data collection, according to Yin (2003) involves the creation of databases to track the collection of data so that external researchers can "*inspect the raw data that led to the case study's conclusions*" (p. 101). Yin (2003) further advises that databases should be organised to collect diverse types of data, with some used to form the case study report, and others to assist the researcher make sense of their data. The following discussion provides summaries describing the set up and administration of four databases used by the researcher to organise workshop, school, and research data sets. All databases were stored on an encrypted, password-protected university owned computer that was stored in a locked research lab, within Trinity College Dublin.

The first database (Figure 4–1, p. 107) was a quantitative database, using SPSS to organise and store raw numeric data obtained though the administration of hard copy forms issued during pilot and formal data collection phases. This database facilitated the manual input of numeric data from workshop and computing in schools data sets, enabling the researcher to explore trends though the manual process of data entry. This process was followed, and these databases maintained until June 2018, when both were decommissioned, and online data collection commenced, removing the requirement for manual maintenance. The raw data was exported into statistical processing software (SPSS) to facilitate the analysis of results. See Appendix 9.14 for examples of numeric data tables generated through using SPSS statistical processing software.

Figure 4-1 Quantitative Research Database IBM SPSS

B	C	D	E	F	G	H	I	J	K	L
Year 1	Year 2	Workshop [	Workshop (	Workshop Name	The overall	The topics	The pace o	The level o	The facilitat	The facilitat
4	2017/2018	11/11/2017	TA21-MOD-3	Scratch 1 - Animation	2	3	3	1	2	1
4	2017/2018	11/11/2017	TA21-MOD-3	Scratch 1 - Animation	2	2	2	2	2	2
4	2017/2018	11/11/2017	TA21-MOD-3	Scratch 1 - Animation	2	2	2	2	2	2
4	2017/2018	11/11/2017	TA21-MOD-3	Scratch 1 - Animation	1	1	2	1	1	1
4	2017/2018	11/11/2017	TA21-MOD-3	Scratch 1 - Animation	1	2	2	2	2	1
4	2017/2018	11/11/2017	TA21-MOD-3	Scratch 1 - Animation	4	4	4	4	2	3
4	2017/2018	11/11/2017	TA21-MOD-3	Scratch 1 - Animation	1	1	1	2	1	1
4	2017/2018	11/11/2017	TA21-MOD-3	Scratch 1 - Animation	1	1	1	1	1	1
4	2017/2018	11/11/2017	TA21-MOD-3	Scratch 1 - Animation	1	1	1	1	1	1
4	2017/2018	11/11/2017	TA21-MOD-3	Scratch 1 - Animation	2	2	3	1	2	1
4	2017/2018	11/11/2017	TA21-MOD-2	Computational Thinkin	3	3	2	2	2	2
4	2017/2018	11/11/2017	TA21-MOD-2	Computational Thinkin	2	2	2	1	1	1
4	2017/2018	11/11/2017	TA21-MOD-2	Computational Thinkin	3	3	2	2	3	2
4	2017/2018	11/11/2017	TA21-MOD-2	Computational Thinkin	1	1	1	1	1	1
4	2017/2018	11/11/2017	TA21-MOD-2	Computational Thinkin	1	1	1	1	1	1
4	2017/2018	11/11/2017	TA21-MOD-2	Computational Thinkin	2	2	2	1	1	1
4	2017/2018	11/11/2017	TA21-MOD-2	Computational Thinkin	2	3	3	1	2	1
4	2017/2018	11/11/2017	TA21-MOD-2	Computational Thinkin	2	2	5	1	1	1

The second database (Figure 4–2), was a qualitative data base, using NVIVO to store raw text data obtained though the administration of hard copy workshop questionnaires issued during pilot and formal data collection phases. The raw data was exported to text processing software (NVIVO) to facilitate the manual coding process. See Appendix 9.13 for examples of coded text.

Figure 4-2 Qualitative Research Database QSR International NVIVO

A	B	C	D	E	F	I	J	K
ID	Year_1	Year_2	Date	Name	Course Code	Instrument	Question Text	Participant Response
5655	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T2)	Please list three skills you have learned	media conversion for use in scratch usin
5656	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T2)	Please list three skills you have learned	The importance of backing up your work
5657	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T2)	Please list three additional skills which	More time using scratch 2.0
5658	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T2)	Please list three additional skills which	Familiarisation myself with all the comm
5659	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T1)	Please list three new skills you would lil	Game design
5660	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T1)	Please list three new skills you would lil	To incorporate score increments
5661	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T1)	Please list three new skills you would lil	To understand variables and operators
5662	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T1)	Please list three existing skills you wou	Use of design
5663	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T1)	Please list three existing skills you wou	relevance to teaching
5664	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Reaction Form	What key learning did you take away frc	overall use of commands and variables
5665	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T2)	Please list three skills you have learned	use of variables
5666	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T2)	Please list three skills you have learned	understanding of the programme (scratcl
5667	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T2)	Please list three skills you have learned	different commands
5668	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T2)	Please list three additional skills which	events and concurrency
5669	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T2)	Please list three additional skills which	additional coding
5670	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Learning Objective Form (T2)	Please list three additional skills which	other commands in scratch
5671	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Reaction Form	More time	Individual use of scratch
5672	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Reaction Form	Less time	Ice breakers
5673	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Reaction Form	Having completed the Bridge21 computi	More group work in class, encourage pu
5674	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Reaction Form	Finally, how might this Bridge21 comput	Plan more activities and facilitate pupil ic
5675	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Reaction Form	More time	programming
5676	3	2016/2017	3/12/2016	Scratch 2 - Game Design	TA21-MOD-4	Reaction Form	Having completed the Bridge21 computi	teaching of other subjects with scratch a

The third database (Figure 4–3, p. 108) was a qualitative administration database using Microsoft Excel, organised to track ethics form completions during pilot and formal data collection phases. Each participant was given a unique code to denote the context in which the data was collected and a unique number, with CSCPD for data captured in a workshop, or TCIN for data captured through the teaching computing in schools questionnaire. The capture of these details facilitated the linking of a completed ethics form to a participant response. This database enabled the researcher to track ethics completions over time, with the researcher able to respond to opt in/opt out ethics requests, as well as to identify and remove data from databases upon request.

Figure 4-3 Ethics Database – Microsoft Excel

ID	Year_	Year_2	Date	Workshop	Course Code	Repea	Surname	Non Cert	Y0_Ethics	Y
1	0	2013/2014	16/11/2013	Scratch 1 - Animation	TA21-MOD-3	1	CSCPD_1	Non Cert	Ethics Received	-
2	0	2013/2014	16/11/2013	Scratch 1 - Animation	TA21-MOD-3	1	CSCPD_2	Non Cert	Ethics Received	-
3	0	2013/2014	16/11/2013	Scratch 1 - Animation	TA21-MOD-3	1	CSCPD_3	Non Cert	Ethics Received	-
4	0	2013/2014	16/11/2013	Scratch 1 - Animation	TA21-MOD-3	1	CSCPD_4	Non Cert	Ethics Received	-
5	0	2013/2014	16/11/2013	Scratch 1 - Animation	TA21-MOD-3	1	CSCPD_5	Non Cert	Ethics Received	-
6	0	2013/2014	16/11/2013	Scratch 1 - Animation	TA21-MOD-3	1	CSCPD_6	Non Cert	Ethics Received	-
7	0	2013/2014	16/11/2013	Scratch 1 - Animation	TA21-MOD-3	1	CSCPD_7	Non Cert	Ethics Received	-
8	0	2013/2014	16/11/2013	Scratch 1 - Animation	TA21-MOD-3	1	CSCPD_8	Non Cert	Ethics Received	-
9	0	2013/2014	16/11/2013	Scratch 1 - Animation	TA21-MOD-3	1	CSCPD_9	Non Cert	Ethics Received	-
10	0	2013/2014	16/11/2013	Scratch 1 - Animation	TA21-MOD-3	1	CSCPD_10	Non Cert	Ethics Received	-
11	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	1	CSCPD_11	Non Cert	Ethics Received	-
12	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	2	CSCPD_12	Non Cert	Ethics On File	-
13	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	1	CSCPD_13	Non Cert	Ethics Received	-
14	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	2	CSCPD_14	Non Cert	Ethics On File	-
15	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	2	CSCPD_15	Non Cert	Ethics On File	-
16	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	1	CSCPD_16	Non Cert	Ethics Received	-
17	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	1	CSCPD_17	Non Cert	No Ethics	-
18	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	1	CSCPD_18	Non Cert	Ethics Received	-
19	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	1	CSCPD_19	Non Cert	Ethics Received	-
20	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	1	CSCPD_20	Non Cert	Ethics Received	-
21	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	1	CSCPD_21	Non Cert	Ethics Received	-
22	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	1	CSCPD_22	Non Cert	Ethics Received	-
23	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	1	CSCPD_23	Non Cert	Ethics Received	-
24	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	2	CSCPD_24	Non Cert	Ethics On File	-
25	0	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	2	CSCPD_25	Non Cert	Ethics On File	-

Finally, the fourth database contained bibliographic data (including case study reports and relevant academic citations) and this data was stored in academic referencing software Endnote.

4.5.3 Principle (3) – Maintain a Chain of Evidence

The third principle of data collection, according to Yin (2003), involves providing a road map which enables those unfamiliar with the research to follow the processes used to collect case study evidence relating to the problem statement, research questions, the case study results and conclusions. Yin describes a chain of evidence as enabling an external observer, or a reader, to "*trace the steps in either direction (from conclusions back to initial research questions or from questions to conclusions)*" (2003, p. 105). This process provides construct validity for the case study, thereby increasing the quality of the research (Yin, 2003). In order to achieve this goal, the researcher included examples of quantitative and qualitative evidence in the results, linking data from the research database with the discussion of findings. Secondly, the research evidence is linked with the context in which it was collected, with the prefix 'CSCPD' used to identify a workshop context, and 'TCIN' to denote a computing in schools context. Thirdly, the researcher adapted the Kirkpatrick (1994) framework as units of analysis which enabled the researcher to link the research context to the case, and the case to units of data analysis, in order to provide a "*clear cross-referencing to methodological procedures and to the resulting evidence"* (Yin, 2003, p. 105).

4.6 Applying Yin's (2003) Case Study Validity Framework

In general, validity refers to the soundness of the research. Whatever framework is chosen to validate research, it is important to consider validity as a process that requires the researcher to consider their own role in influencing the research (Cohen et al., 2000), how data sources merge and relate to one another (Ely et al., 1991), and who the audiences of our texts will be (Kent, 1996). The concept of validity is a one that is shared by the interpretivist and positivist communities (Trauth & Jessup, 2000). Thus in applying criteria to establish validity "*it is understood that the argument will reflect our particular theoretical, social, political, and personal interests and purposes*" (Lynch, 1996, p. 54). Methodological accounts should therefore include a description of the steps involved in validating the results, as well as the methods used to judge the validity of claims.

Cohen et al. (2000) argues that "*the researcher will need to locate (his) her discussions of validity within the research paradigm that is being used*" (p. 106) and as such, the choice of validity framework is guided by the researcher's interpretivist philosophical tradition and associated methodology. However, there have been claims that interpretivist research in general, and case study research in particular, fails to provide sufficient clarity to substantiate research claims and support generalisation, and that claims about the nature of reality can never be absolute (Denzin, 1997; Wells et al., 2002). Having set out the methodological argument for using an interpretive, case study design, the researcher opted for a validity framework compatible with interpretivist research, and chose Yin's (2003) criteria (Table 25, p. 110). The following discussion presents the validly framework implemented by the researcher to ensure that the results are sound and reliable.

4.6.1 Test (1) - Construct Validity

Yin (2003) defines construct validity as a process used to establish the "*correct operational measures for concepts being studied*" (Yin, 2003, p. 34). The researcher started the validation process by drafting research questions prior to data collection, which were reviewed by key informants. This process set the parameters for the evaluation, and set the trajectory of data collection according to the protocols established by the Kirkpatrick (1994) framework. In order to maintain a chain of evidence the researcher used the results gathered from each phase of analysis as primers for exploring new leads, or used to customise research instruments in order to increase the scope and depth of questions asked. A variety of instruments and research methods were used to collect quantitative and qualitative data, providing "*multiple sources of evidence*" (Yin, 2003, p. 36).

Table 25 Validity Framework for Case Study adapted from Yin (2003)		
Test	Construct	Rationale
Test (1)	Construct Validity	• Use multiple sources of evidence • Establish a chain of evidence • Have key informants review draft reports
Test (2)	Internal Validity	• Use pattern matching • Address rival explanations • Use logical models
Test (3)	External Validity	• Refer to theory • Use replication logic
Test (4)	Reliability	• Develop a case study protocol • Develop a case study database

4.6.2 Test (2) - Internal Validity

Internal validity refers to the level of confidence we can place in the cause-and-effect relationships identified through the research. One of the problems experienced when conducting case study research though an interpretivist lens, is the problem of subjective influence (Denzin, 1997). In order to address this, logical models are used in interpretivist research to manage theoretical uncertainty as such models provided a structure within which to examine new responses as well as explore dissenting responses within one data set (see chapter 5 which describes the analysis process.)

When presenting the research findings, researchers should consider "*how can we decide whether something is true on the basis whether we have decided it is true*" (Hammersley & Atkinson, 1995, p. 95). Indeed, there are, as Hammersley and Atkinson (1995) recommend, an implicit need to specify and qualify what criteria, and they recommend that "*any criteria should be heuristic, relying on tacit and always questionable assumptions in their application, these applications therefore being subject to potential debate*" (p. 61).

A key component of establishing validity is the need to consider 'rival explanations.' This does not mean that judgements of validity cannot be made on the results of interpretive research. Rather, "*the testing of the value of these insights to others can be carried out by exposing the approach through verbal and written discourse to enable broader judgements to be made*" (Walsham, 1993, p. 6). In subscribing to an interpretive view, the researcher had to make choices which influence the lens taken explore the research results. This, "*only one of the many possible valid accounts of the phenomena studied*" (Hammersley, 2001, p. 54) are presented in this thesis. Following Hammersley and Atkinson's suggestion to make explicit the measure taken to validate qualitative and quantitative results, what follows is a description of these measures.

(1) Qualitative Measures - Pattern Coding

The researcher opted for a pattern coding framework for this study. A pattern coding process provided the researcher with the flexibility to link theory to open coding, and open coding to the use of deductive methods to reduce the overall data into sub-themes and themes (Fereday & Muir-Cochrane, 2006). A pattern process also provided a loose framework catering for constant comparison and pattern building as the heuristic to develop themes from the data (Spradley, 1979). The rational underpinning the selection of this approach is discussed in chapter 5, Section 5.5.

(2) Quantitative Measures 1 – Descriptive Statistics

Descriptive frameworks set out to build a picture of phenomena, bringing together variables related to age, profession, depth of experience and expertise to add context (Oppenheim, 2000). Moreover, descriptive frameworks explore frequencies, helping to pin point surges in responses towards a particular question or capture the socio-demographics or profiling information helping to set a base line to guide further exploration and analysis (Rose & Sullivan, 1996). This study utilised a descriptive framework across all but one quantitative data set, to provide contextual or profiling information in addition to information designed for trending over time (Rose & Sullivan, 1996). The rational underpinning the selection of this approach is discussed in chapter 5, Section 5.4.

(3) Quantitative Measures 2 - Bivariate Statistics

Bivariate analysis frameworks seek to establish relationships between 'dependent' and 'independent variables' (Oppenheim, 2000). Bivariate statistics are useful for exploring small samples, where comparisons between are required. For example Bivariate statistics are used to explore statistical significance between two samples where participants complete the same Likert scale questions, prior to and after completing a learning experience. Explained another way, Bivariate analysis uses comparative algorithms which explore differences or similarities between two 'variables' or particular items (Rose & Sullivan, 1996).The researcher used bivariate analysis to explore teacher perceptions of their learning prior to and after CPD workshops (see chapter 5, and Section 5.4).

4.6.3 Test (3) - External Validity

Establishing external validity requires preparing the data for generalising and *"establishing the domain to which a study's findings can be generalised"* (Yin, 2003, p. 34). While abstracting concepts and generalising themes from coding is perceived as offering partial insights into social phenomena (Ryan & Bernard, 2000), interpretivist researchers are somewhat reluctant to make substantive claims that are perceived as contributing to or supporting broad generalisations (Klein & Myers, 1999). However, the interpretivist also needs to make sense of their data and communicate findings in a form relevant and accepted by wider domains.

The researcher developed two algorithms to oversee the processing of qualitative and quantitative data sets (chapter 5). These algorithms were derived from theory and designed to support data analysis and sense making, using techniques drawn from interpretivist literature, generate results giving a very limited but somewhat generalisable view of the research findings.

4.6.4 Test (4) - Reliability

Reliability is defined as providing the reader with an audit trail of the processes followed to conduct case study research, thus demonstrating that the procedures, if repeated, would be likely to produce the same results (Yin, 2003). In order to comply with this step, Yin (2003) suggests following a case study protocol creating a case study database. The researcher used Yin's (2003) case study protocol template to describe the design of methods used to complete the evaluation work. Section 4.5 describes the databases that were used to keep track of the data over five years of research.

4.6.5 Summary

This section set out the validity framework adapted by the researcher to respond to claims that interpretivist research in general, and case study research in particular, fails to provide sufficient clarity to substantiate research claims and support generalisations. Furthermore, this section described validity tests used for validating case study research with the treatments describing how each test was addressed in the context of this research. The researcher acknowledges that there are multiple frameworks used to validate research results and specialised frameworks designed for the interpretivist, and further frameworks for case work (see Lincoln and Guba (1985) who explore validity across positivist, post-positivist, constructivist and interpretivist domains). The researcher applied Yin's (2003) validity tests in acknowledgement that it provided a checklist of common tests accepted as validity markers in case study research. The researcher also used Yin's (2003) validity framework as it provides markers compatible with the case study and data collection protocols, and provides a road map through the methodological design.

4.7 Limitations

The researcher acknowledges that there are limitations in applying an interpretivist case study approach. Firstly, the methodology designed for this research could be viewed as being based on "*subjective, value laden interpretations*" (Lynch, 1996, p. 54). Furthermore, some may call into question the lack of 'rigor' used to validate research results when using an interpretivist approach (Lincoln & Guba, 1985). In defence, the researcher agrees with Lofland and Lofland that "*naturalistic investigation, with its preferences for direct apprehension of the social world, has somewhat fewer problems with validity than do research traditions which rely on indirect perception*" (Lofland & Lofland, 1984, p. 50). What this means in the context of this research is that the researcher has implementing this particular methodological approach to provide one of many interpretations. This research is founded in the belief that that there are no absolute truths, but rather truth claims are based on a common understanding of what is real and meaningful.

It is also important to acknowledge there are limitations with the protocols selected for use in this research. Yin's (2003) protocols provide a structure to guide the researcher through the process of describing the methods used to implement their methodology. However Yin's (2003) protocols provides limited opportunity for the researcher to integrate validity metrics compatible with the interpretivist paradigm, and the researchers interpretivist ontological beliefs (i.e., Cohen et al., 2000, pp. 105-106). One example of this is the treatment of 'reliability,' which and Cohen et al. (2000) argues should be extended to include a reflexive account of the research provided by the researcher, as well as well as a more detailed description of the research context, case and units.

While there is no single, correct approach, the use of published protocols provides an opportunity for others to assess the appropriateness of methods, theories and tests used to verify research outcomes. In this work, the researcher adapted Yin's (2003) protocols to add clarity to a somewhat complex process of the methodological design. These protocols were utilised in an attempt to ensure that to the researcher's own representation of phenomena were as accurate as possible (Lofland & Lofland, 1984, p. 50), thus enabling the researcher "*to feel confident about the research findings and to convince others of their accuracy*" (Fetterman, 1989, p. 20).

Finally, validity is not achieved through simply describing the methods used in data collection, rather "*research takes place in, and is addressed to, a community*" (Lincoln, 2002, p. 334). Community consensus is achieved when individuals or groups with no knowledge or experience of the research can empathise with the researchers retelling of the methodological process and make informed decisions on the outcomes. Thus, the researcher has only succeeded in validating their methodological approach when "*the ultimate test is whether the research tools, and the results obtained, are accepted by other scholars as having validity*" (G. Marshall, 1998, p. 687).

5 Analytical Framework

The methodological approach set out the parameters for conducting research and provided the theory for the design of research methods used to collect data to address the research questions. This chapter provides the analytical frameworks, which the researcher designed and implemented to guide data analysis of qualitative and quantitative data. Table 26 provides a model representing the steps required to analyse data from mixed data sets from an interpretivist perspective. The model has been developed though reading literature which criticises interpretive researchers for failing to document the process used to develop meaning from their data (Klein & Myers, 1999; Walsham, 1993). The following data analysis model designed by the researcher, aims to provide clarity, describing the theories, strategies, and models and that are used in for interpretive research.

Table 26 Data Analysis Model

Section	Content	Strategy adopted for this research
5.1	Theories for Analysing Data	Evaluation Strategy - used verifying theory as it applies to a particular context (Stake, 1983a).
5.2	Analysing Mixed Methods Data Sets	Explanatory Mixed Method Design - understand the 'causal nature' of phenomena (DeMaris & Selman, 2013).
5.3	Using Logical Models to Guide Analysis	Hypothetic-Deductive model - used for generating then deducing concepts (LeCompte & Schensul, 1999).
5.4	Statistics and Analysis of Quantitative Data	Descriptive Framework - used for calculating frequencies per variable (LeCompte & Schensul, 1999).
5.5	The Role of Coding in Qualitative Data Analysis	Pattern Coding - used for checking existing values against new values (Strauss & Corbin, 1998).

5.1 Theories for Analysing Data

The problem with data analysis is that there is no one agreed procedure to produce a standard set of results (Stufflebeam & Webster, 1983). Rather, researchers use lots of different processes to generate research findings which are then used to develop research claims drawn from their data (Caracelli & Greene, 1993). The decisions supporting the use of analytical processes are also in part influenced by the researcher's philosophical beliefs (O'Donoghue, 2007), insofar that domain based rules govern what claims can be made and what procedures are used to transform the data into research evidence (Burch & Heinrich, 2016). The challenge for the interpretivist researcher rests with exploring what role theory plays in scaffolding the data analysis processes so that claims are reconstructed in a way sympathetic to the meaning making process (Armstrong, Davis, & Paulson, 2011). Moreover, interpretive researchers face additional complexity insofar that the processes used to produce research claims are perceived as 'intersubjective' (Prus, 1996) and are therefore perceived as difficult to pin down, isolate or combine into a form that produces definitive evidence. Prus (1996) describes intersubjectivity as the process by which we come to understand phenomena and learn to draw meaning from phenomena, which we experience in social contexts.

The interpretive researcher embarking on the path of data analysis is faced with a number of challenges, each seeking to reconstruct participant voices and their written responses in a plausible framework (Denzin, 1997). The first challenge concerns deciding which analytical strategy to use to process the data and what role theory plays in the analytical process (Fetterman, 2001). The second challenge concerns deciding the order followed to combine qualitative and quantitative data sets, (Burch & Heinrich, 2016). The third challenge concerns the dilemma of selecting a logical model appropriate to assign meaning to the data or to derive meaning from the data (McLaughlin & Jordan, 2015). The fourth challenge concerns selecting a coding framework compatible with the researcher's philosophical beliefs and compliant with the researcher's treatment of intersubjectivity (Prus, 1996).

5.1.1 Choosing an Analytical Strategy

The process of collecting, analysing and making sense of the data is complex, prompting researchers to develop or adopt strategies to scaffold the data analysis process (Creswell & Clark, 2011). Furthermore, selecting which analytical strategy to use to scaffold analysis is problematic, as each strategy is governed by a series of rules which influence the way research results are developed (Lincoln & Guba, 1985). Moreover, care must be taken when setting out to apply a particular strategy to the data as different strategies can lead the researcher towards developing theory, testing out theoretical propositions or evaluating theory as applied to a particular context. The challenge for the interpretivist researcher rests with choosing a strategy which adheres to the conventions of a particular paradigm (Weiss, 1997).

(1) Theory Based Strategies

There is much contention as to what role theory plays in the process of data analysis (Astbury & Leeuw, 2010; O'Donoghue, 2007; P. J. Rogers & Weiss, 2007; Stame, 2004). Furthermore, strategies which gather data in incremental amounts over time are neatly positioned to help the researcher develop patterns which are then used to build concepts from their data (Johnson & Christensen, 2014). Mixing together and combining data sets produces a strong evidence base designed to provide new theoretical insights (Glaser & Strauss, 2012) while at the same time steering the researcher on a course towards generating theory (A. Clarke & Dawson, 1999). Weiss (2000) suggests that 'theory based' strategies provide frameworks useful for developing emergent ideas into emergent findings. Weiss continues that theory based strategies contribute to knowledge insofar that "*even relatively small increments of knowledge about how and why programs work or fail to work cannot help but improve*" (2000, p. 44) and help to unveil deeper meanings.

(2) Theory Testing Strategies

Theory testing strategies use data collection methods administered at repeated and regular intervals to capture the moment of observable change (Cronbach, 1983b). Testing models play a pivotal role in helping the researcher explore the predicators of change useful for measuring the depth or the reach of particular interventions used within social programmes (Ajzen, 1991). Testing strategies are designed to explore theoretical propositions using models to seek out plausible or causal connections to uncover the reasons why a particular change in behaviour may have occurred (Trafimow, 2003). Choosing a testing strategy over a theory building strategy enables the researcher to mine the data for evidence of change (Guyatt, Walter, & Norman, 1987) rather than building an evidence base for change. Testing strategies also enable the researcher to produce trending information from large and complex data sets where large samples and increased responses are needed to support claims intent on reporting on observable shifts in social behaviour (Greene, 2012). Indeed, Cronbach et al. (1963) argues that theory testing provides a base for generalising.

(3) Evaluation Strategies

While theory building strategies seek to build an evidence base from the ground up and theory testing strategies seek to analyse the data to support generalizable research claims, evaluation strategies provide a framework geared to explore theory as it is applied in a context (W. V. Harris, 2000). Evaluation strategies recognise subjectivity in the interpretive process, seeking to understand the semantics of phenomena as reported by participants involved in a process (Ricoeur, 1974); at the same time, it uses measures to map potential results to models to help structure and explain the results in a form suitable for generalising. Evaluation strategies also seek to link formal data analysis procedures with theory to develop a connected view of theory as it is applied in a *particular* context (Stake, 2000). Evaluation strategies can also combine interpretation, comparison, and testing procedures to substantiate research claims.

Though criticised as a weak strategy for attempting to appease multiple philosophical positions (Lincoln & Guba, 1985), evaluation strategies provide a working hypothesis to explore theory applied in context.

For these reasons, the researcher opted for an evaluation strategy to conduct the process of data analysis. An evaluation strategy enabled the researcher to bring together data from multiple sources in acknowledgement that studies of social phenomena are complex and therefore require a multifaceted approach to the meaning making process (Stake, 2000). Moreover, an evaluation strategy enabled the research to work with theory over the duration of the analysis where it was used to: help identify concepts which might yield new or innovative theoretical insights (Eisenhardt, 1989), explore trends and patterns for testing out theoretical propositions (Yin, 2003); and evaluate the use of a particular theoretical model as it was applied in practice (Stoecker, 1991). Opting to follow an evaluation strategy provided the researcher with the opportunity to follow emergent leads and themes through the data while at the same time using targeted collection methods to delve deeper into the data to look for possible explanations.

5.2 Analysing Mixed Methods Data Sets

The previous section explored the importance of choosing an analytical strategy to collect and link data sets. This section explores two problems, with the first concerned with decoding the term data (Lave, 1988) and the second concerned processing the data in a form compatible with building, testing or evaluating theory (Rogoff, 2003). Different data types can be processed as a separate entity, in a consecutive sequence or concurrently with other sources (Giddings & Grant, 2006). Furthermore each combination creates the potential to influence theorising (Cohen et al., 2000).

5.2.1 Triangulated Designs

Triangulated mixed methods designs complement exploratory and explanatory mixed methods designs, as they attempt to simultaneously collect data over time (Creswell, 2005). The underlying rationale supporting triangulated designs is that each data is ‘neutral’: text and numbers both play a role in helping the researcher construct an understanding of research phenomena (Fetterman, 2001). Rather, viewing qualitative and quantitative data equally as one, polarized, source of ‘data’ (Maxwell, Bashook, & Sandlow, 1986) enables the researcher to analyse data within a framework released from paradigmatic restrictions (Lave, 1988). Moreover, triangulated designs help the researcher corroborate or strengthen the underlying rationale used to produce research results (Wolcott, 1994). Triangulated designs enable the researcher to work more closely with the data by providing the means to build a substantive base to address a specific problem (Wolcott, 1994).

5.2.2 Exploratory Designs

The exploratory mixed method design concentrates on using methods to develop an initial understanding of the research context (Stebbins, 2001) then using methods to seek out and ‘explain relationships found in the data’ (Creswell, 2005, p. 516). An exploratory design also enables the researcher to start organising emergent concepts into frameworks (Stebbins, 2001) and is useful for helping the researcher understand ‘what is going on’ (Wolcott, 1994). From an interpretivist perspective, exploratory designs are useful for helping the researcher start conversations with small groups in order to test out ideas and gather responses which can be worked into developing conceptual models (Wholey, 2015). Moreover, exploratory designs help refine the research focus (Guba & Lincoln, 1981)- using measures to set in motion the exploration of trends, sequences and patterns followed by measures which help delve deeper into theory building (Weiss, 2000).

5.2.3 Explanatory Designs

Explanatory mixed methods designs gather data in two phases; using instrumentation to gather data to build a more general ‘picture of the problem statement’ (Creswell, 2005) followed by methods to add further definition and context to explain phenomena (Ivankova et al., 2006). Explanatory designs set out to understand the ‘causal nature’ of phenomena (DeMaris & Selman, 2013) to construct an evidence base to support theoretical generalising (Cronbach et al., 1963). From an interpretivist perspective, explanatory designs enable the researcher to use methods to scope then deepen focus. This design enables the researcher to draw boundaries, cordoning off the topics of interest which can then be addressed with targeted analysis to reveal its causal nature (Scriven, 1974). Designed to complement theory testing strategies (Guba & Lincoln, 1981) explanatory designs enable the researcher to mine the data, delving deeper to uncover the cause of change (Ivankova et al., 2006).

An explanatory mixed methods design enabled the researcher to treat the data as ‘neutral’ in an attempt to help the researcher identify trends and patterns within the data (Harris, 2000). An explanatory design also helped the researcher to start building theoretical models and schemes at the start of the data collection process, providing empirical data as a base for scoping out leads to be further explored through qualitative research (Fetterman, 2001). Furthermore, an explanatory design helped provide a base for rigour (Guba & Lincoln, 1981) while also creating a mechanism to make connections between qualitative and quantitative data (Jick, 1979). Combining an evaluation analytical strategy (see section 5.1.1, item 3) with an explanatory mixed methods design provided the researcher with the flexibility to respond to phenomena as it emerged while at the same time use techniques to verify themes or concepts emerging from the data (Michalski & Cousins, 2000).

5.3 Using Logical Models to Guide Analysis

Having explored the implications of selecting a strategy and design compatible with mixed methods data collection, this section explores the role that logical models play in the interpretive process (Schwandt, 2000). The interpretivist process can be described in two parts: first as a process which encourages the researcher to keep in close contact with the data in order to understand the tacit nature of 'structures, words and events' (Ricoeur, 1974), and second as a technique concerned with using models as heuristics (Ball, 1993) designed to help unlock the politics of meaning (Geertz, 1973). Logical models play a key role in bringing the interpretive processes and related techniques into alignment. Furthermore, logical models provide structures useful for helping researchers make sense of their data. Logical models also provide structures useful for organising data into categories, patterns or trends and play a vital role in helping the researchers link concepts together (LeCompte & Schensul, 2013). There are three common models used for developing conceptual links (Table 27).

Table 27 Logical Models for Developing Concepts (LeCompte & Schensul, 2013)

Model	Rationale
Inductive Model	Used for generating concepts (Lofland, 1971).
Deductive Model	Used for deducing concepts (Miles & Huberman, 1994).
Hypothetico-Deductive Model	Used for generating then deducing concepts (LeCompte & Schensul, 1999)

5.3.1 Inductive Models

This strategy uses constant comparison as the analytical mechanism used to find links between phenomena (Lofland, 1971). Inductive logical models provide structures useful for helping the researcher link the raw data to theory (McLaughlin & Jordan, 2015). Inductive models seek to derive meaning from the data, and are useful for helping the researcher arrange, sort and order concepts into structures to support theory building (LeCompte & Schensul, 1999). LeCompte and Schensul (1999) suggest sorting the data into 'units', linking the units to 'categories' and categories to concepts connected to theory. Inductive models encourage working backwards and forwards through the data, to draw out patterns to uncover embedded ,meanings (Stebbins, 2001).

5.3.2 Deductive Models

Deductive models set the researcher on a path of theory testing in so far that the process involves 'selecting, focusing, simplifying, abstracting and transforming the data' into a form which supports sense making (Miles & Huberman, 1994). Performed over the duration of fieldwork, deductive models provide a way of analysing the data using techniques which consolidate thinking. Deductive models encourage the researcher to sub subdivide the data into meaningful clusters, read through the data, develop categories and then apply a sorting mechanism designed to merge the resulting concepts into themes (LeCompte & Schensul, 1999). In contrast to inductive models, deductive models start with a particular taxonomy in mind which is applied to the data with the purpose slotting the data into categories (Bailey, 1994). The process of deduction could be described as classification process as it involves sorting through the data in a systematic way to reduce complexity, identify differences, set out a list of dimensions which can help the researcher understand links or relationships between phenomena (Bailey, 1994, pp. 13-14).

5.3.3 Hypothetico-Deductive Models

While each model offers different ways of interacting with the data, hypothetico-deductive models play a duel role in helping the researcher generate concepts and then reduce concepts into themes using theory as a focusing mechanism (Timmermans & Tavory, 2012). Popper (1972) is accredited with developing the hypothetico-deductive method as an approach for gathering evidence of the occurrence of phenomena linked to research questions or an overarching hypothesis. Hempel (1983) further describes the hypothetico-deductive method as a philosophical approach, which is used for 'appraising theory' which is linked to a particular context. Hypothetico -deductive models start with a theory and use inductive and deductive reasoning to develop concepts which link back to theory (Baggini & Fosl, 2003). Hypothetico-deductive models also provide the conceptual flexibility to work back and forward through the data on the premise that certain concepts reveal themselves more readily while others require more in-depth investigation (Eco, 1984). Hypothetico-deductive models provide the flexibility to revisit, question, reconstruct, deconstruct and reassemble ideas in a form designed to abstract a more meaningful understanding of concepts (Denzin, 1989). Furthermore, hypothetico-deductive models act as scalable frameworks helping the researcher evaluate the phenomena, thus care must be taken not to use models to reduce the data to disconnected variables removed from their context (Blumer, 1969).

Having explored the role different logical models play in scaffolding the interpretive process, the researcher opted for a hypothetico-deductive logical model as the framework used to organise the data into a from suitable for abstracting meaning and supporting sense making. The researcher followed a hypothetico-deductive approach to analyse the data in clusters, with clusters mapped to units of data analysis, to focus theorising within a context. Hypothetico-deductive models start from a position of theory in the form of a working hypothesis or problem statement (Lincoln & Guba, 1985). Starting the analytical process with 'theory in mind' helped the researcher to explore the data on two levels (Trafimow, 2003) – first at an abstract level by sweeping across the data to look for commonalities, abnormalities and deviations between concepts – and second at a more concrete level involving concept merging, comparison and reorganising with the purpose of linking concepts back to theory. Moreover, using inductive and deductive reasoning together helped the researcher develop a broad based view of the data while at the same time spot gaps or respond to incongruities within the data (Timmermans & Tavory, 2012). Finally, while logical models help explore anomalies; the process of coding the data brings more focused precision to meaning making.

5.4 Statistics and Analysis of Quantitative Data

There are three numeric frameworks used to add depth and context to quantitative data sets. The first of these is the multivariate analytical framework.

5.4.1 Multivariate Analysis

Multivariate analysis helps the researcher combine the data into new formations. Multivariate frameworks are concerned with revealing the 'causal nature and effect' of phenomena (Oppenheim, 2000), where multiple variables are combined to develop explanations. Again machine based statistical procedures can help the researcher combine or segment data sets to explore particular trends and help the researcher to develop an explanatory picture of phenomena before or after a particular event or research intervention (Oppenheim, 2000). Multivariate analysis is designed to analyse multiple concepts at the same time, helping to reveal gaps, inconsistences or new explanations from unexpected data combinations (Rose & Sullivan, 1996). Multivariate techniques are also positioned to locate particular differences or similarities between phenomena by producing a set of results to help explain the properties of one particular variable or set of variables against the properties of others (LeCompte & Schensul, 1999).

5.4.2 Bivariate Analysis

Bivariate analysis helps the researcher explore connections between data, and uses frameworks seek to establish relationships between 'dependent' and 'independent variables' (Oppenheim, 2000). Machine supported analytical processes such as 'cross-tabulations' help the researcher break down the data using different measurements of association to explore connections or links between two variables (Oppenheim, 2000). Bivariate analysis uses comparative algorithms, which explore differences or similarities between two variables or particular items. Used as a focusing tool, bivariate analysis helps to explore or regress particular concepts (Rose & Sullivan, 1996).providing a way to open up numbers into a form which helps explore relationships between data (LeCompte & Schensul, 1999). Bivariate analysis is used in interpretive research to help researchers explore perceived differences between sample groups, providing indicators of the statistical difference between two sets of compared numbers. Indeed, LeCompte and Schensul (1999) suggest using bivariate statistical analysis as a methodology to explore 'how variables relate to one another' (p. 155). Bivariate analysis can play an important role in helping researchers "*determine whether these relationships truly exist or are simply random or chance occurrences*" (LeCompte & Schensul, 1999, p. 156). However, LeCompte and Schensul (1999) advise using bivariate statistics in context, with other methods including descriptive analysis helping to bound, and focus exploration.

5.4.3 Descriptive Analysis

The third and final framework examined in this section is the descriptive framework. Descriptive frameworks set out to gather metrics which provide counts or summative measures useful for developing trends or building profiles (Oppenheim, 2000). Descriptive frameworks also set out to build a picture of phenomena, bringing together variables related to age, profession, depth of experience and expertise – in essence fact finding or profiling data designed to add context (Oppenheim, 2000). Moreover, descriptive frameworks explore frequencies, helping to pin point surges in responses towards a particular question or capture the socio-demographics or profiling information helping to set a base line and structure for quantitative data sets retrieved through field work (Rose & Sullivan, 1996). Furthermore, descriptive frameworks help establish the 'mean, median and mode' for responses and enable the researcher to track averages related to a particular cohort, grouping or over a more longitudinal, distributed range (LeCompte & Schensul, 1999).

Having explored three frameworks for processing numbers, this study primarily utilised a descriptive framework, seeking to provide contextual information (Rose & Sullivan, 1996) in combination with a bivariate framework which explored links between phenomena in the learning data set. These analytical methods were used in contrast to a multivariate approach which seeks compare multiple phenomena to drill down into the cause and effect of phenomena, to produce longitudinal indications or trends (Rose & Sullivan, 1996). Descriptive and bivariate analysis was applied without compromising the researcher's philosophical belief systems, with the results perceived as indicators rather than absolutes (Caracelli & Greene, 1993). Moreover, these numerical frameworks played a key role in meaning making, especially when used with qualitative data. Though sometimes viewed as 'simple statistics' (Malec, 2018) descriptive frameworks provide category information that can be linked back to a particular cohort or group, providing information important for highlighting knowledge gaps or exploring particular preferences (Ajzen, 1991). Bivariate statistics also played an important role in helping the researcher "*examine relationships between pairs of variables. Bivariate statistics are used to find, describe, and test the significance of the association between two variables*" (LeCompte & Schensul, 1999, p. 152).

5.5 The Role of Coding in Qualitative Data Analysis

Having explored the role logical models play in shaping interpretation, the following section explores the links between logical models and the coding process. Coding is a process that involves assigning meaning to text or numbers (Miles & Huberman, 1994). Coding frameworks are abstract structures designed to assist the researcher in developing meaning from unprocessed data (Creswell, 1998). There are similar frameworks used for processing numeric data, where scales are used to support theorising (Oppenheim, 2000). Likewise, processing numerical data also requires making decisions as to what scales are used to make sense of the data as each system can influence the development of research claims (DeMaris & Selman, 2013). A limitation with coding text is that 'weak' or poorly constructed codes can fail to convey meaning, making concept mapping and further sense making difficult (Eco, 1984). The following explores coding frameworks used to derive meaning from data.

5.5.1 Open Coding

Coding frameworks provide guidelines to help the researcher construct codes and then assign codes to words, sentences or large segments of text (Strauss & Corbin, 1998). In the case of text, the researcher assigns word codes or alpha numeric codes to the raw data in an attempt to extract out or release meaning from participant responses in an organised and structured way (LeCompte & Schensul, 1999). The process of coding helps the researcher reorganise data sets in order to make conceptual connections between disparate pieces of data (Ryan & Bernard, 2000). The act of reorganising the data is a form of abstraction used to label and then reassemble phenomenon into categories, themes or domains which the researcher uses to explore commonalities or differences within the data (Strauss & Corbin, 1998). This type of coding process, also called 'open coding' helps the researcher develop categories which can then be used to map the categories to other domains, frameworks or models (Fereday & Muir-

Cochrane, 2006). Open coding is particularly effective in helping to classify each item of data, following a line-by-line sorting process, in preparation for using comparison to identify patterns or links between phenomena.

5.5.2 Axial Coding

Axial coding is a more structured process which seeks to catalogue the data and then cluster related themes around a centre point or axis (Strauss & Corbin, 1998). Similarly to open coding, the actual linking process 'takes place not descriptively but rather at a conceptual level' (Strauss & Corbin, 1998, p. 125). Axial coding seeks to slice through the data rather than split the data in an attempt to establish the underlying cause of phenomena – causality and casual explanation form the basis of the axial coding approach (Scriven, 1974). The process of axial coding moves between the process of logical induction and reductionist deduction (Strauss & Corbin, 1998); merging the meaning of one concept with the meaning of another which are further compared then refined into more defined concepts. A limitation with axial coding is that it involves a continuous cycle of checking and refining concepts. This has the effect of expanding theory, making the process of reaching an end point problematic (Lincoln & Guba, 1985).

5.5.3 Pattern Coding

Pattern coding frameworks help the researcher map emergent codes to a theoretical scheme or framework (Strauss & Corbin, 1998). The pattern coding process uses open coding to identify concepts and emergent themes, and then it draws from axil coding in an attempt to link emergent concepts to a central point or axis. Pattern coding frameworks as described by Strauss and Corbin (1998) map the outcome of the coding process to 'a list of existing categories' either developed by the researcher or present in other analytical frameworks. Limitations with this approach relate to the problem of ensuring that the concepts found through coding and analysis fit with the categories borrowed from other domains. Mapping concepts to other frameworks may highlight inconsistences or may not be compatible with the data emerging from coding (Miles & Huberman, 1994).

The researcher opted for a pattern coding framework for this study. A pattern coding process provided the researcher with the flexibility to link theory to open coding, and open coding to the use of deductive methods to reduce the overall data into sub-themes and themes (Fereday & Muir-Cochrane, 2006). A pattern process also provided a loose framework catering for constant comparison and pattern building as the heuristic to develop themes from the data (Spradley, 1979). Also, the pattern process provided the researcher with the scope to work codes from pre-existing frameworks into the coding process, providing focus and direction during coding (Stake, 1983b).

A pattern coding framework offered the benefit of providing flexibility to accommodate divergent paths through the data. However, working from a less defined structure may cause initial problems related to working out how to link concepts emerging from the data back to the overarching framework. Accordingly, the researcher organised the coding processes into distinct phases which could be linked back to the overarching theoretical framework. Examples of this process can be seen in Appendix 9.13, with coding organised into phases and organised under questions which could be linked back to the theoretical framework developed for this research (see Section 5.3). The process of generating word codes from reading text is also called 'in-vivo' coding, and is used to assign meaning and context to phenomena reported by participants (Bryant, 2017). In-vivo coding or word coding was performed on qualitative data with the purpose of identifying themes, which could be abstracted into broader themes (Hammersley & Atkinson, 2006, p. 195). The process of word coding consisted of reading and re-reading qualitative responses and constructing word codes from reading the text. Word codes were constructed by the researcher and assigned to segments of text, then arranged or clustered into themes (Hammersley & Atkinson, 2006).

5.5.4 Limitations

Coding is a form of analysis which helps to identify patterns or themes within large data sets (Miles & Huberman, 1994). Codes are useful tags or labels, which give data meaning and significance. Codes can be words or phrases: but 'words tend to be fatter than numbers' (Miles & Huberman, 1994, p. 56). However a limitation with coding is that the researcher needs "*to decide on what codes are relevant to the emergent themes of the work*" (Hammersley & Atkinson, 2006, p. 199), and there is more than one way to create themes (Guba & Lincoln, 2000). Also, themes are difficult to categorize because they are "*abstract (and often fuzzy) constructs that investigators identify before, during, and after data collection*" (Ryan & Bernard, 2000, p. 275). Creating themes creates "*a potentially convenient way of organizing a great deal of cultural information into a relatively coherent ordering of a few categorie*s" (Hammersley & Atkinson, 2006, p. 224). However, over time, as more data became available, sub-themes were refined, evaluated, and merged into themes. The researcher acknowledges these limitations and chose pattern coding as it enables the researcher to work through text responses multiple times, facilitating recoding, and the creation of new categories. Another way to ensure that codes are developed to address research questions is to use a taxonomy to structure coding. The following section explores how taxonomies can be used to focus coding.

5.6 Mapping Each of Kirkpatrick Levels to the Data Analysis Model

Taxonomies are structures which help researcher design codes compatible with a particular theory so that the researcher can link emerging results back to a theoretical framework including research questions (Bailey, 1994). LeCompte and Schensul (1999) suggest using taxonomies at the start of the coding process to develop codes, which can be mapped back to a particular theory or questions (pp. 82-83). A limitation with using a taxonomic approach is that the researcher is 'locked into a theoretical structure' at the start of the coding process (Bailey, 1994). Furthermore, starting the coding process informed by the structure of a taxonomy provides limited opportunity to follow divergent paths through the data (Lincoln & Guba, 1985). A taxonomy can help to maintain focus during coding (Alkin, 2012) as LeCompte and Schensul's (1999) process illustrates in Table 28.

Table 28 Using a Taxonomy to Guide Coding (LeCompte & Schensul, 1999)

Step	Rational	Adapted
(1)	Create or Adapt Taxonomy.	Kirkpatrick levels adapted as coding categories.
(2)	Identify Patterns across the Data.	Categories mapped to individual questions.
(3)	Establish Themes in the Data.	Individual question content analysed over time.
(4)	Connect Themes back to Taxonomy.	Results presented under categories in levels.

5.6.1 Step (1) - Create or Adapt a Taxonomy

LeCompte and Schensul (1999) suggest starting the process by identifying core categories to be addressed through data analysis. Having defined the research questions prior to the selection of the evaluation framework, the researcher adapted each Kirkpatrick (1994) level into a unit of data analysis with research methods designed to collect data corresponding to each unit. The mapping covers workshop and school contexts, and before and after impacts, with Vygotsky and the Bridge21 model used as the overarching theories involved in shaping and influencing the analysis of the data.

5.6.2 Step (2) - Identify Patterns across the Data

LeCompte and Schensul (1999) then recommend using pattern coding to identity concepts within qualitative data sets. Pattern coding (also known as theming) involves reading and re-reading texts, using comparative techniques to identify similarities and differences between constructs and then assigning constructs codes to denote the difference. The researcher used pattern coding procedures in the following way. The researcher first sorted the data per level (category). Then the researcher split out all numeric data from text data. Descriptive descriptions and frequency calculations were generated for statistical data. The researcher then proceeded to assign word codes to text records and code for similar or contrasting views, as more data was added to the data set over time. Final analysis ceased in June 2018, on receipt of last responses from teachers teaching in schools.

5.6.3 Step (3) - Establish Themes in the Data

Pattern coding initiated a process, which enabled the researcher to spot trends and themes emerging in the data. However, further refinement of codes was required as the size of data sets and the number of participants attending workshops increased. LeCompte and Schensul (1999) encourage extending the analysis to look for contrasting or unexpected results not accounted for in the taxonomy. The researcher adapted the coding of text data to include iterative coding cycles (sweeping once though the data to code for emergent phenomena and then sweeping two further times though the data to cross check and merge similar codes and filter for contrasting codes in the data). This process enabled the researcher to reduce large amounts of data to a sub set of themes used to structure the discussion of results. Numeric data was reviewed by the researcher at the point of collection and reviewed during data entry where anomalies where reported in field notes.

5.6.4 Step (4) - Linking Themes back to the Taxonomy

The final part of the analytical process required mapping the findings (in the form of themes and percentages) back to the overarching theoretical framework or taxonomy. This final phase required the researcher to ensure that findings developed from coding were compiled into tables or graphs linked back to a category which in turn mapped onto a unit in the Kirkpatrick (1994) framework. LeCompte and Schensul (1999) advise that this final coding stage should describe theoretical anomalies such as possible theoretical contributions, instances where theoretical constructs have been tested and offer contrasting results or where theory is evaluated and offers unique insight concerning changes in practice. Using a taxonomy proved useful as a way to structure coding and the analysis of the results and provided the researcher with a way to map an existing evaluation framework onto a complex, codding process. LeCompte and Schensul’s taxonomy helped the researcher focus the research to look for constructs and provided sufficient scope to revise and include new themes into the analysis in response to new data.

5.6.5 Limitations

Finally, while a taxonomy provides a structured approach to the analysis of constructs, it limits the possibility to look for new leads or explore fresh evidence not related to the framework. Furthermore, taxonomies help to 'bound' or focus the research, but they also close off access to phenomena not compatible with the framework or can cause mapping inconsistencies where no clear mapping between constructs emerges. Engineering is therefore required to work out how to include additional concepts into frameworks and manage coding in light of new theories not included at the start of the process. Finally, social research, and the coding and analysis of qualitative data obtained from social settings is perceived as difficult (Ely et al., 1991; Lofland & Lofland, 1984; Wolcott, 1994). The mapping provided in this sub-section provides one of many possible combinations used to address the research questions, thus the results presented in chapter 7 present one of many interpretations rendered from the adaptation of Kirkpatrick (1994).

5.7 Algorithms used to Process Quantitative and Qualitative Data Sets

Deciding which algorithms to choose to process text and numeric data is problematic (LeCompte & Schensul, 1999). As previously discussed, there are a number of different coding frameworks that are used to process text and numeric data, with each using techniques that place different emphasis on the results that are generated. Having opted to follow to use a hypothetico-deductive model to shape coding and analysis, the researcher used pattern coding procedures designed to remain open to concepts emerging from the data, but also coded the data with the aim of mapping themes back to an overarching taxonomy (LeCompte & Schensul, 1999). Opting to code within a structure prompted the researcher to develop algorithms to produce results that could be mapped back to a central taxonomy and then the research questions using the Kirkpatrick (1994) framework.

5.7.1 Algorithms Developed for this Research

Having linked the coding process to a taxonomy, the researcher then developed algorithms to process numeric data sets and a separate algorithm to process text data sets (Table 29, p. 130). The researcher opted to use the descriptive statistics function in a statistical software package (SPSS) as the core statistical analytical function to process numeric data. Descriptive statistics provided the researcher with the ability to calculate court, sum, mean, median frequencies.

Table 29 Algorithms for Generating Statistical Results			
Questionnaire	Calculation	Impact	Effect
Mixed Methods Questionnaire 1 – CPD Context			
Level 1 Reaction Likert Scale Post-Test CPD Responses.	Sum, Average of Means, Percentage, Cronbach (1951) Alpha.	Cumulative, over time, giving an overall mean score value and year on year analysis.	Descriptive Statistics, including Frequencies to provide indicators.
Level 2 – Learning Likert Scale Pre and Post-Test CPD Responses	Sum, Average of Means, Percentage, Cronbach (1951) Alpha.	Cumulative, over time, giving an overall mean score value and year on year analysis.	Descriptive Statistics, including Frequencies to provide indicators. Excludes Digital Media responses.
	Wilcoxon (1945) Signed Rank Test	Test 1 - Pair-matched pre-test median and post-test median.	Statistical significant difference between pair matched pre and post per workshop module and first pre and last post scores across workshop modules.
		Test 2 – pair matched first pre and last post median.	
Level 3 Behaviour Likert Scale Post-Test CPD Responses	Sum, Average of Means, Percentage, Cronbach (1951) Alpha.	Cumulative, over time, giving an overall mean score value and year on year analysis.	Descriptive Statistics, including Frequencies to provide indicators.
Mixed Methods Questionnaire 2 – Schools Context			
Level 4 – Results Likert Scale Post-Test CPD Responses	Sum, Average of Means, Percentage, Cronbach (1951) Alpha.	Cumulative, over time, giving an overall mean score value.	Descriptive Statistics, including Frequencies to provide indicators.

The researcher did not use cross comparative statistics given the philosophical dilemma of perceiving numeric data as having equal standing to that of qualitative data (Lave, 1988). Given the researcher's interpretivist position, descriptive statistics was used in the first instance to produce numeric calculations, providing sum, mean, percentages and frequencies (Goetz & LeCompte, 1981). The decision to use descriptive statistics as indicators is based on the rational that text or numbers provide partial insights into the participants perceptions of phenomena (LeCompte & Goetz, 1982).

The researcher then used bivariate statistics to explore the statistical difference between two variables. As stated above, CPD participants were invited to complete a pre-test Likert scale exploring learning outcome variables prior to the start of each computing workshops. CPD participants were then asked to complete a corresponding post-test at the end of each computing workshop, covering the same Likert scale questions. Bivariate statistics were used on a sub set of learning outcome results where the researcher obtained matched pre and post results. LeCompte and Schensul (1999) argue that while descriptive statistics play an role in providing context and scope, the next step in data analysis "*should be to examine the relationship between pairs of variables*" (p. 156). The researcher followed LeCompte and Schensul (1999) recommendation, building in bivariate statistics into the logical model in the form of a Wilcoxon (1945) Signed Rank Test as a way to show statistical difference between learning outcome variables.

5.7.1.1 Wilcoxon (1945) Signed Rank Test

The researcher used a Wilcoxon (1945) Signed Rank Test, on a sub-set of responses, which compares two medians to determine whether they are statically different enough to show meaningful difference between two populations. The Wilcoxon test is the equivalent to a *t*-test, but is used where there is subjectivity in the data collection process. The learning outcome data set was (a) not of 'interval strength' nor (b) evenly distributed, thus deemed 'subjective' and not suitable for a *t*-test. The researcher used a Wilcoxon Signed Rank Test to explore the statistical difference between pre and post learning outcomes. A Wilcoxon test calculates statistical significance to determine the difference in median values. Medians are perceived as a more accurate 'indicator of central tendency' (Leys, Ley, Klein, Bernard, & Licata, 2013).

The Wilcoxon (1945) Signed Rank Test, is used to "*allocate a sign to each observation, according to whether it lies above or below some hypothesized value*" (Whitley & Ball, 2002, p. 511). There are reservations about using the Wilcoxon test in that it is reported as 'extremely simple to perform,' and lacks the capacity to perform the necessary statistical analysis to support more complex 'hypothesis testing' (Whitley & Ball, 2002). In defence, Whitley and Ball (2002) argue that Wilcoxon is "*useful for dealing with unexpected, outlying observations that might be problematic with a parametric approach*"(p. 513), and that it offers the capacity to examine the significance of the increase or decrease in results between two variables, in relation to each other.

Wilcoxon has been used in behavioural science to determine the change between pre and post results in a medical training (de Lima, Laranja, Bromberg, Roesler, & Schröder, 2005), and has been used in nursing to assess the impact of change on policy ward management (Thoroddsen & Ehnfors, 2007). Within education, the Wilcoxon test has been used to explore the relationship between pre and post-test scores within sample, which were collected at different points over time (Drennan & Hyde, 2008), with Boyas, Bryan, and Lee (2012) using Wilcoxon as an indicator to obtain a better representation of actual 'gains' in performance variables. Furthermore, Bhanji, Gottesman, de Grave, Steinert, and Winer (2012) used Wilcoxon in addition to descriptive statistics, to explore the extent to which the change in median reflected a statistical significant increase or a decrease in scores. Of particular relevance to this research is the rational provided by Bhanji et al. (2012) who built Wilcoxon into the assessment of pre and workshop post learning outcome variables, as a part of their implementation of the Kirkpatrick model, to give validity to the reporting of self-assessed responses.

5.7.1.2 Pattern Coding Process

The researcher designed a similar algorithm to process textual data. Section 5.6 covers the pattern coding process that was used to generate themes, which were then mapped to units of data analysis, which corresponded to each level in the Kirkpatrick model. However, a summary of the same process is now described. Having sorted text data into categories the researcher then set out to code text data mapped to a category. The data was processed in the order in which it was collected. The processing of the data per category served the following purpose. First, coding within a defined or bounded category enabled the researcher to read through the answer to the same questions multiple times in order to identify patterns within the data. On reading and re-reading responses multiple times, the researcher then proceeded to code the data at two levels, first using coding to tag emerging phenomena, and second to merge similar codes together. The researcher followed this process over the duration of the research and during periods of analysis, which occurred during the input of new data into the text database, and again when analysing text data to produce papers. This process was followed for all text, from workshop and school questionnaires, and for text contained in field notes.

5.7.2 Data Set Description and Constraints

To recap, data sets were processed in the order in which they were collected. Teachers were invited to complete a pre-test questionnaire on arrival to each workshop, and were invited to complete a corresponding post-test questionnaire at the end of each computing workshops. A further sub-sample of CPD teachers completed a post CPD questionnaire, on return to schools. The post CPD questionnaire was issued at the end of the research process, capturing feedback from teachers who had attended at least one workshop, within 6 months up to 5 years since departing the CPD.

The CPD data set is larger than the teaching computing in schools data set. The researcher attended the delivery of most of the computing workshop modules, over the duration of the research and administrated the questionnaire to teachers attending the CPD. See Appendix 9.3 for a list of workshops included in the data set. The teaching computing in schools data set is somewhat limited in that only a sub-set of teachers agreed to participate in follow up research, with the researcher issuing the teaching computing in schools questionnaire to teachers via electronic mail, rather than face to face in a workshop.

The overall data was then split into categories and processed per data type. Text data was analysed using pattern coding procedures (see section 5.5) to develop themes while statistical data sets were processed to produce percentage responses and statistical significance markers (see section 5.4). Both algorithms (for processing text and numbers) provided the researcher with the means to make sense of the data using techniques compatible with an interpretivist approach. While there are more complex algorithms for interrogating the data (e.g., Creswell & Clark, 2011; Miles & Huberman, 1994), the algorithms described in this sub-section helped the researcher make sense of the data using techniques for exploring patterns and following trends across and within large and mixed data sets. Both algorithms enabled the researcher to work with the data in a form designed to preserve the original structures, extending quotes to themes and aggregating numbers to percentages, using numbers as indicators.

5.8 Summary

Finally, Interpretivists look for patterned regularities – the occurrence of more than one instance of the same social phenomena (Wolcott, 1994). The process of reading and re-reading the data is perceived to bring clarity. Interpretivists use coding and recoding "*to gain a* *new perspective on our material and to focus further data collection, and may lead us in unforeseen directions*" (Charmaz, 2000, p. 515). Some interpretivists argue meaning making involves "*breaking up textual data into themes and categories; a process which requires the reorganisation of data into discreet chunks or segments and identifying them in accordance with a coding system*" (Hammersley & Atkinson, 2006, p. 193). While others, such as Hammersley and Atkinson (2006) argue that deconstructing texts is central to the process of analysis (p. 195). However there are interpretivists who argue "*no matter how the researcher actually does inductive coding, by the time he or she has identified themes and refined them to the point of texts, a lot of interpretive analysis has already been done*" (Guba & Lincoln, 2000, p. 781). Whichever process is used, interpretivists agree that "*there can never be a final, accurate representation of what was meant or said - only different textual representations of experiences*" (Denzin, 1997, p. 5). To conclude, this chapter set out the rational for an analytical frameworks used to guide data analysis, with the aim of adding 'objectivity' to what can be described as a 'subjective' process (Prus, 1996).

6 Findings

The analytical framework provided an overview of the process that was used to generate results from the data. This chapter covers the findings in detail. All findings are organised according to the Kirkpatrick (1994) model which was adapted as units of analysis. The first part of this chapter explores the CPD research findings, covering demographics, teacher perceptions of the CPD, reactions to the content, learning by the participants and teacher intention to use methods learned in the CPD, and examples teachers plan to use to increase student engagement. The CPD data set was generated from teachers attending Bridge21 CPD workshops between January 2014 and June 2018. During this period, a self-selecting sample of *N* = 1,215 teachers attended *N* = 72 Bridge21 CPD workshops and were invited to complete pre and post-workshop questionnaires.

The second part of this chapter investigates the participants' subsequent teaching of computing in schools, covering demographics, with further sections exploring what elements of the Bridge21 CPD teachers used to teach computing, barriers to teaching computing, other methods used for teaching CS as well suggestions for enhancing the Bridge21 CPD programme. Also included are examples covering use of the Bridge21 model for increasing student engagement. The school data set was generated from a self-selecting sub-sample of *N* =385 CPD participants who were invited to complete a post CPD questionnaire after they had at least one year experience of teaching computing in schools. This sample is smaller than that for the CPD, given that a smaller sub-set of teachers consented to be included in follow up researcher. Of the *N* = 385 teachers sent the follow-up questionnaire, *N* = 64 responded, thus the school findings are defined as 'representative'.

Both the CPD and the teaching computing in schools statistical results were produced using frequency, descriptive and significance tests. Cronbach's (1951) Alpha reliability coefficient values were calculated for Likert scaled items to assess the internal consistency of the scale variables. Wilcoxon's (1945) Signed Rank Test was used to compare the statistical difference between pre and post-workshop CPD learning outcome variables. LeCompte and Schensul's (1999) logical framework was used to guide pattern coding and theming which was used in the analysis of all qualitative data.

6.1 Units of Data Analysis

This first section revisits the rational for using Kirkpatrick's (1994) as units of data analysis to structure the discussion of the research findings. The Kirkpatrick framework provides the capacity to explore participant perceptions of CPD programmes and their experiences of using the CPD content in their place of work. A further benefit of using the Kirkpatrick framework is the ability to adapt each 'level' into a unit of data analysis. The researcher considered developing units of analysis through the process of coding from the data (Glaser & Strauss, 2012). However, a limitation with developing units through direct coding is that the process may generate 'vague' categories making it difficult to organise the research findings into units for discussion (Goetz & LeCompte, 1981). An alternative approach is proposed by Lofland (1971), who created units of analysis from theory, as a 'device' to structure analysis. Having considered the emergent approach proposed by Glaser and Strauss (2012), and the prescriptive approach proposed by Lofland (1971), the researcher chose to adapt Kirkpatrick's four levels as units of data analysis inspired by Lofland's used of predefined units.

6.1.1 Scope

Before continuing with the analysis, there are limitations with adapting Kirkpatrick (1994) as units of data analysis, including a lack of capacity to explore phenomena out with the model (Bates, 2004), and the difficulty in implementing all four levels (Reio et al., 2017). Adapting each of Kirkpatrick's levels into separate units of data analysis means that the data exploring the student experience is omitted form the analysis. In defence, the researcher adapted Kirkpatrick, as Kirkpatrick provides a starting point to focus on the teacher experience, and a starting point for exploring the impact of the Bridge21 CPD workshops on teacher preparation for teaching computing. Furthermore, the researcher did not have a professional background in teaching, was based in the CPD context, and did not have front line access to schools and their students. To address this limitation the researcher reanalysed the CPD and teacher data sets to include examples describing teacher planned and actual use of the CPD content and methods in a computing classroom context exploring the CPD impact on student engagement. These examples provide evidence of teachers planned and implemented use of the Bridge21 model as a method for enhancing student engagement in a computing context. Moreover, given that computing is a new subject in the Irish curriculum, it is too early and beyond the scope of this thesis to make substantive statements about student learning.

6.1.2 Road Map

In accordance with Kirkpatrick (1994), the remainder of the sections covered in this chapter follow each of Kirkpatrick's four levels in sequence. The first section explores teacher reactions to the CPD workshops covering demographics, teacher perceptions of the workshop content and satisfaction with the CPD. Section two investigates teacher perceptions of their learning, with further statistical analysis examining the change between pre and post-workshop results. Analysis of teacher learning outcomes is followed by research, which explores strategies used for teaching computing, and teachers' intentions to use the Bridge21 model, and examples teachers plan on using in a classroom context. Having investigated teachers' intentions, the fourth section explores the results in terms of teachers' experience of teaching CS in schools covering demographics, use of the Bridge21 model, barriers to implementation, other methods use for teaching CS, further CPD for using Bridge21 in a computing context, with qualitative analysis provide examples used by teachers. Limitations with implementing the Bridge21 model are then explored with this chapter concluding with a summary.

6.2 Teachers' reactions to the CPD workshop content

The previous section provided the rational for adapting the Kirkpatrick (1994) framework as units of data analysis to structure discussion of the findings. This section explores the research findings which are mapped to the first level in Kirkpatrick's model and examines teacher reactions to the CPD. The reactions data set was generated from the administration of a post CPD workshop mixed methods questionnaire (Appendix 9.9) with field notes providing context to the results. The questionnaire content was adapted from an existing instrument (Kristiansen, 2007) and was administered at the end of each CPD workshop to explore teacher reactions to the workshop. The data was collected to answer research the question: Q1.1 what are teachers' reactions to the CPD workshop content?

6.2.1 Data Set Limitations

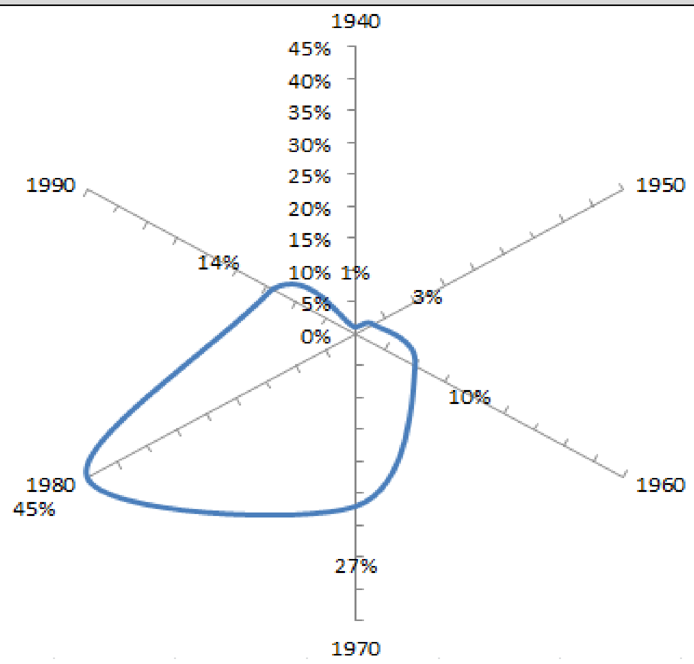
Before exploring the reaction research findings the limitations of the data set need to be discussed. The reactions data set contains responses from teachers who had completed the mandatory module TA21-Mod-1: Digital Media Literacy and 21st Century Learning. This module introduced teachers to digital technologies as well as the Bridge21 model, which means that teachers had some (limited) prior exposure to the Bridge21 model and its use as a method of teaching prior to engaging with the CS CPD. Having covered data set limitations, the next section explores teacher CPD demographics.

6.2.1.1 Demographics

This section explores teacher CPD demographics, covering the age profiles, gender representation, and workshop frequency. Descriptive statistics were used to analyse numeric data sets.

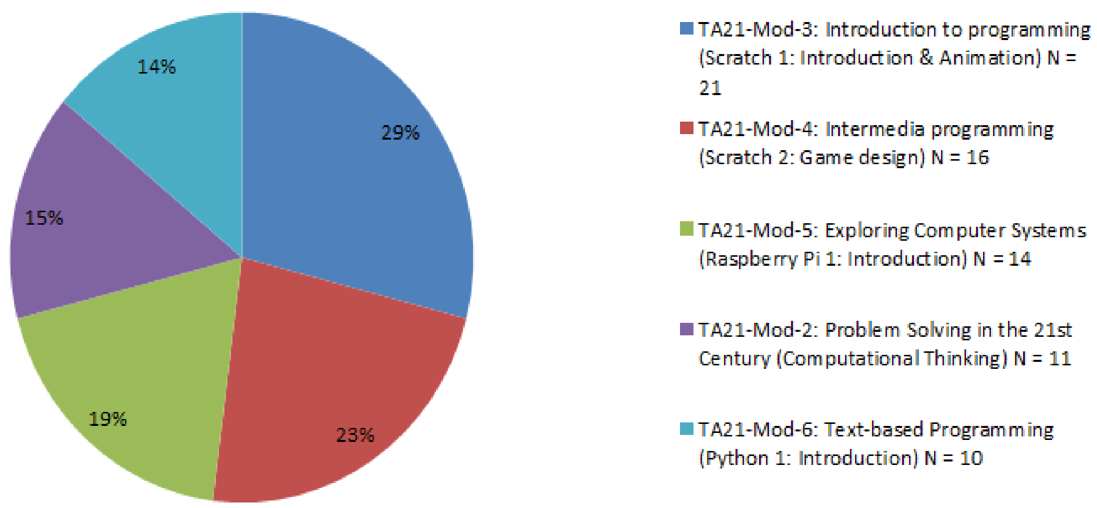
Teacher age profile analysis explored the age profile of CPD teachers (Figure 6-1, p. 138). The age profile data set contained *N* = 443 individual dates of birth which were obtained from the post-workshop questionnaires completed by teachers (Appendix 9.9). Individual dates of birth were anonymised for reporting, with individual dates of birth grouped into ten-year bands. Analysis shows that teachers in their 30's (*N* = 198) were the largest cohort of teachers represented in the CPD sample, with 45% of teachers reporting that they were in this age range. Further analysis shows that teachers in their 40's (*N* = 121) were the second largest cohort attending the CPD workshops, with 27% of teachers reporting dates of birth in this age range. The third largest cohort, were teachers in their 20's (*N* = 61), with this CPD cohort representing 14%. Analysis of the remaining 14% confirmed that teachers aged 50 and over, were represented in the CPD sample.

Figure 6-1 Teacher CPD Demographics Age Profile



CPD workshop frequency analysis (Figure 6–2, p. 139) explored the number of times each workshop was delivered during the five-year research period. Analysis of *N* = 72 computing workshops type (Appendix 9.3) confirm that 29% of workshops covered Scratch 1 – Animation content (*N* = 21), with 23% covering Scratch 2 - Game Design content (*N* = 16). Furthermore, 19% covered the Raspberry Pi (*N* = 14), with a further 15% covering Problem Solving/Computational Thinking content (*N* = 11). The remaining 14% of workshops covered Python (*N* = 10).

Figure 6-2 CPD Workshop Module Frequency Analysis



6.2.1.2 Summary

Demographic analysis reveals that teachers in their 30's were the largest cohort represented in the CPD workshop sample, with 45% of teachers represented in that age group (Figure 6–1, p. 138). The most frequently delivered workshop (Figure 6–2) was Scratch 1 – Animation (29%), followed by Scratch 2 -Game Design (23%). These demographics show diversity in age and CPD coverage.

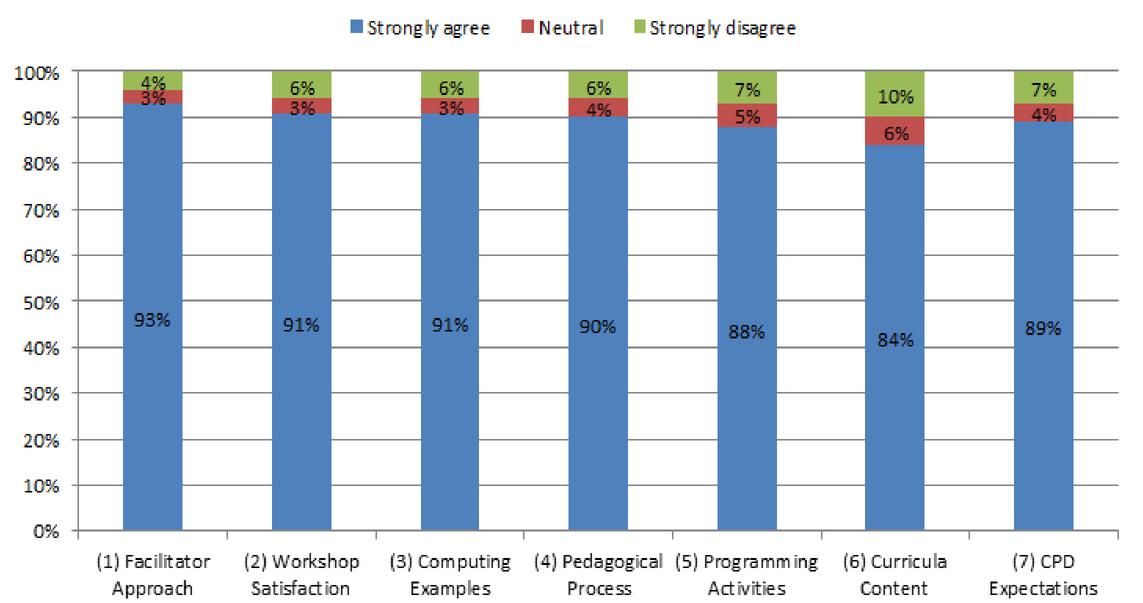
6.2.2 Quantitative Analysis – All Years

Analysis of the post-CPD workshop reaction variables (Table 30, p. 140), which are arranged 1 = Strongly agree to 7 = Strongly disagree, confirm that teachers overall reactions to the CPD workshops as positive (m = 1.71). Teachers agreed that they were very satisfied with the curricula content (m = 2.19) and with the pedagogical process used to teach the content (m = 1.94). Teachers reported that they were very satisfied with the computing examples that were used in the workshops (m = 1.77), and agreed that they had also enjoyed the programming activities (m = 2.00). Furthermore, teachers strongly agreed that they enjoyed learning computing through a facilitator approach to learning (m = 1.45), with teachers also agreeing that the CPD met their professional development expectations (m = 1.89). A Cronbach's alpha score of = .961 indicates a strong level of internal consistency for this scale. Each of these variables is explained in turn, starting with (1) the facilitator approach.

Table 30 Reaction Post-Workshop Variables All Years							
Cronbach's alpha = .961 indicating a strong degree of reliability or consistency with this scale							
7 Point Likert Scale [1 = Strongly agree; 7 = Strongly disagree]							
Measure of Agreement							
Variables	(1) Facilitator Approach	(2) Workshop Satisfaction	(3) Computing Examples	(4) Pedagogical Process	(5) Programming Activities	(6) Curricula Content	(7) CPD Expectations
Responses	817	813	809	817	815	813	428
Mean	1.45	1.71	1.77	1.94	2.00	2.19	1.89

The following diagram (Figure 6-3) provides a visual representation of the level of agreement for linked to each mean value. The Likert scales supporting the following analysis are reported in Appendix 9.14.1. Scales are arranged 1 = Strongly agree to 7 = Strongly disagree, with the results of supporting the coding and analysis of qualitative results reported in Appendix 9.13.2.

Figure 6-3 Reaction Variable Analysis – Summary - All Years



(1) Variable - Facilitator Approach

These post-workshop results confirm that 93% of teachers strongly agreed that a facilitator approach to the CPD was appropriate; with 3% of teachers providing a neutral view and 4% of teachers disagreeing that facilitation was an appropriate teaching method. Further analysis of field notes reports that a facilitator led CPD gave groups autonomy: "*I liked the freedom to do* *what you like*" (CSCPD_1196). The use of facilitation in a computing CPD context also encouraged teams to decide for themselves what content they wanted to explore and learn: "*I liked that we worked as a team and that we had the freedom to choose our own topics and do our own research. I felt good about what I was learning and I learned some skills that I can contribute"* (CSCPD_1115).

(2) Variable - Workshop Satisfaction

These post-workshop results also confirm that that nearly all teachers agreed (91%) that the workshops were worth attending and that teachers were very satisfied with the workshop experience. Only 3% of teachers neither agreed nor disagreed with the statement, with 6% indicating that they were not satisfied with a workshop format of CPD. Further qualitative analysis of field notes reports that the workshops provided positive learning experiences: *"I felt it was a positive experience and benefitted from the process. Team working was strong and I learned from others"* (CSCPD_825). The workshops were also perceived as a space where teachers were free to share knowledge and expertise: "*I formed new friendships and support mechanisms. I felt challenged at times and did not process info correctly. Overall enjoyed the workshop and how it created a safe learning environment*" (CSCPD_380).

(3) Variable - Computing Examples

Furthermore, the post-workshop results confirm that 91% of teachers agreed that the examples were satisfactory. Only 3 % of teachers neither agreed nor disagreed that materials were satisfactory, with remaining 6% of teachers reporting that they were not satisfied with the examples. Further qualitative analysis of field notes reports that examples helped teachers make connections between concepts: "*I achieved an understanding of how hardware and software comes together"* (CSCPD_987). Examples were also designed to have a practical impact "*I learned good examples of using Bridge21 Philosophy, good insights into programming scratch to python, and interfacing with electronics*" (CSCPD_1033).

(4) Variable - Pedagogical Process

Also, the post-workshop results confirm that 90% of these teachers agreed that the pedagogical process was clearly communicated and was to their satisfaction. Only 4% of teachers provided a neutral view, with 6% somewhat dissatisfied with the communication of the process. Further qualitative analysis of field notes reports that the process helped teachers think about computing: "*we were set a problem, the blocks available, and how they came together, we made an animation, and the whole process went well*" (CSCPD_1195). Teachers also planned to use the same pedagogical process in the context of their teaching: "*it will* *encourage me to work more towards structured problem solving and work with students to develop processes they can apply to everyday problems"* (CSCPD_2017).

(5) Variable - Programming Activities

These post-workshop results show that 88% of teachers were satisfied with the activities, with 5% neutral and 7% reporting that the programming activities were difficult. Further qualitative analysis of field notes reports that the programming activities were designed to give teachers a practical understanding of computing and programming: "*there was more to computing than I thought. That the language used can be simplified to activities students could do."* (CSCPD_2030). Teamwork also played a critical role in helping individuals' complete complex programming activities: *"group work is essential to keep yourself motivated when the programs are too complex for the individual"* (CSCPD_1041).

(6) Variable - Curricula Content

These post-workshop results further confirm that teachers strongly agreed that the level of difficulty of the curricula content was appropriate for them (84%), with 6% of teachers neural in their view, and the remaining 10% somewhat dissatisfied with the curricula. Further qualitative analysis of field notes reports that the curriculum used tasks, which encouraged problem solving: "*I could observe the effect of various minor changes in code on the outcome and managed to complete most of the tasks"* (CSCPD_620). The curriculum was also comparable with methods already used by teachers: "*I already use a similar approach in my classroom (student-created content, group work) but would like to use this model. I'd like to share this model, these concepts and the technology used with my colleagues*" (CSCPD_260).

(7) Variable - CPD Expectations

Finally, these post-workshop results, confirm that teachers strongly agreed that the CPD met with their expectations (89%), with 4% of teachers neural in their view, and 7% dissatisfied. Further qualitative analysis of field notes reports that the workshops provided teachers with content they could apply in their teaching: "*I had the opportunity to think more about my topic so I can delve deeper in to it in the future. Colleagues achieved a sense of camaraderie and confidence*" (CSCPD_ 258). The CPD also provided a structure, which teachers could relate to students and the classroom: "*I got really good to get resources that can be immediately put to good use in the classroom. Better way of giving CPD because it can be difficult to apply theory given at CPD into practical resources*" (CSCPD_372).

6.2.2.1 Summary

Teachers enjoyed learning computing through a facilitator led approach to CPD (*m* = 1.45), with 93% of teachers agreeing that a facilitator lead approach to teaching gave them the time to direct their learning and explore concepts of interest with their teams. Teachers also responded positively to the examples that were used in the workshops (*m* = 1.77), with teachers strongly agreeing that they were helpful in understanding computing content (91%).

Teachers agreed that the pedagogical process was satisfactory (*m* = 1.94), with 90% of teachers agreeing that the process was clearly communicated and met to their satisfaction. The depth of programming activities covered in the workshops were also perceived as satisfactory (*m* = 2.00), with 88% of teachers agreeing that the programming activities were well organised. Teachers also reported that they were very satisfied with the curricula content (*m* = 2.19), with 84% of teachers satisfied with the level of difficulty. Finally, the teachers agreed that the workshops met with their CPD expectations (*m* = 1.89), with 89% of teachers satisfied with the CPD workshops, and 91% of teachers satisfied with the workshop format (*m* = 1.7.1).

Finally, qualitative analysis provides examples, which show that teachers enjoyed the workshop examples, and reported that a facilitator led CPD experience gave them the freedom to choose topics to research and explore, supported by their peers. Teachers also reported that the computing workshops provide a safe environment to share, discuss, and explore methods and content used for teaching computing lessons. Teachers provided examples which demonstrated that the pedagogical process provides a structure to introduce problem solving into computing lessons, with teachers giving further examples which demonstrate how computing activities can be adapted to give students a contextual view of computing, with the workshops providing resources for teachers.

6.2.3 Quantitative Analysis – Year on Year

The previous section provided an aggregated view of the reaction findings. Further analysis exploring the same data set, year on year; provides the capacity to explore change over time. The following section explores the reaction findings, organised by year starting with Year 1. Table 31, p. 144 provides a summary of mean values reported year on year, which corresponds with the following analysis. Likert scale tables supporting the year on year analysis are listed Appendix 9.14.2.

Table 31 Reaction Variables – Mean – Year on Year							
7 Point Likert Scale [1 = Strongly agree; 7 = Strongly disagree]							
Variables	(1) Facilitator Approach	(2) Workshop Satisfaction	(3) Computing Examples	(4) Pedagogical Process	(5) Programming Activities	(6) Curricula Content	(7) CPD Expectations
Year 1 - 2013/2014							
Responses	211	211	207	209	209	209	-
Mean	1.45	1.49	1.55	1.9	2.06	2.19	-
Year 2 – 2014/2015							
Responses	176	172	174	176	176	176	-
Mean	1.58	1.72	1.85	1.84	1.86	2.12	-
Year 3 – 2015/2016							
Responses	156	157	155	157	156	155	154
Mean	2.04	2.04	2.07	2.25	2.35	2.45	2.12
Year 4 – 2016/2017							
Responses	192	190	190	192	191	190	191
Mean	1.48	1.7	1.74	1.95	1.92	2.16	1.76
Year 5 – 2017-2018							
Responses	83	83	83	83	83	83	83
Mean	1.35	1.60	1.65	1.7	1.7	1.93	1.76

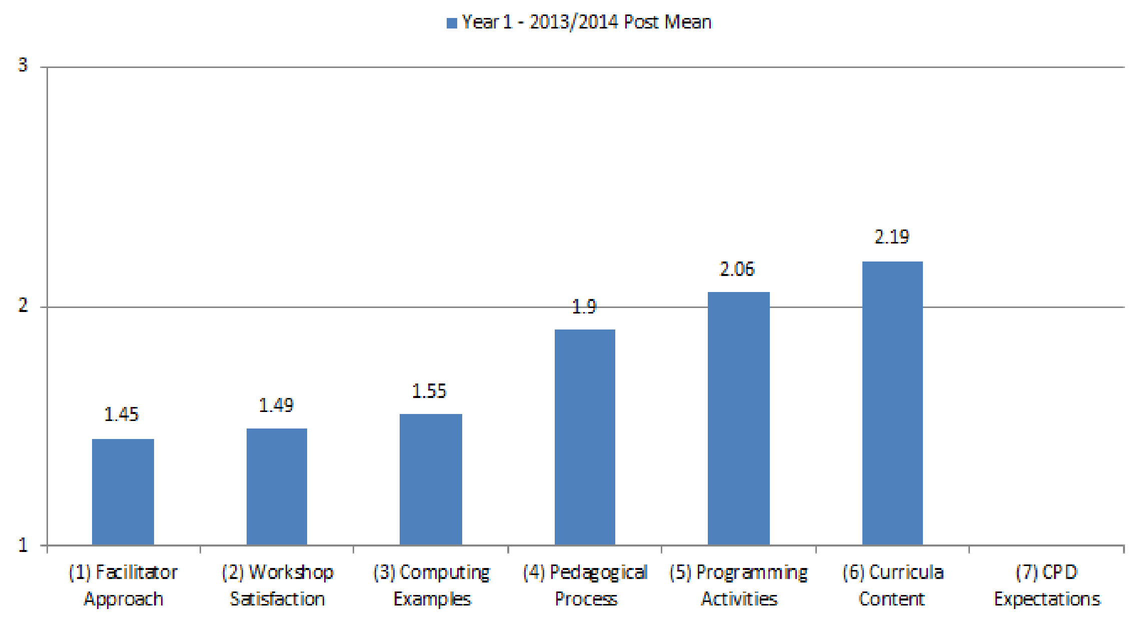
(1) Year 1 – 2013/2014

The year on year data set includes responses from teachers were attending the first year of the CPD programme, with workshops not yet linked to the one year, part time Post Graduate Certificate in 21st Century Teaching and Learning programme. Teachers attending the computing workshops registered for interest and were encouraged to adapt the CPD content to their teaching needs. This data set contained post-workshop responses, with the CPD expectation variable not yet reported.

Quantitative analysis of the first year results (Figure 6-4) found that teacher reactions to the workshops were very positive and that 96% of teachers agreed that facilitation was an appropriate CPD approach (*m* = 1.45), with a further 96% agreeing that the workshops were worth attending (*m* = 1.49). In addition, 97% of teachers agreed that the examples were helpful (*m* = 1.55) and 91% of teachers agreed that the overall process was clearly communicated (*m* =1.9). Finally, 87% of teachers agreed that the programming activities were well organised and easy to understand (*m* = 2.06), with a further 83% of teachers agreeing that the level of difficulty of the content was appropriate (*m* = 2.19).

Figure 6-4 Reaction Post-Workshop Variables Year 1

7 Point Likert Scale Results Clustered 1 = Strongly agree; 2 = Agree; 3 = Somewhat agree



(2) Year 2 – 2014/2015

The year two data set includes responses from the first cohort of teachers attending the one-year part time Post Graduate Certificate in 21st Century Teaching and Learning programme. Other factors impacting upon the data set include the publication of the NCCA short course in Coding (2014a) and the Digital Media Literacy short course (2014b) occurring in the same year. Computing workshops were scheduled for one full day during term time, with shorter assignment sessions scheduled the following week, covering project planning and implementations. Full day workshops were run on Saturdays, with shorter assignment support sessions run on the following Friday evening.

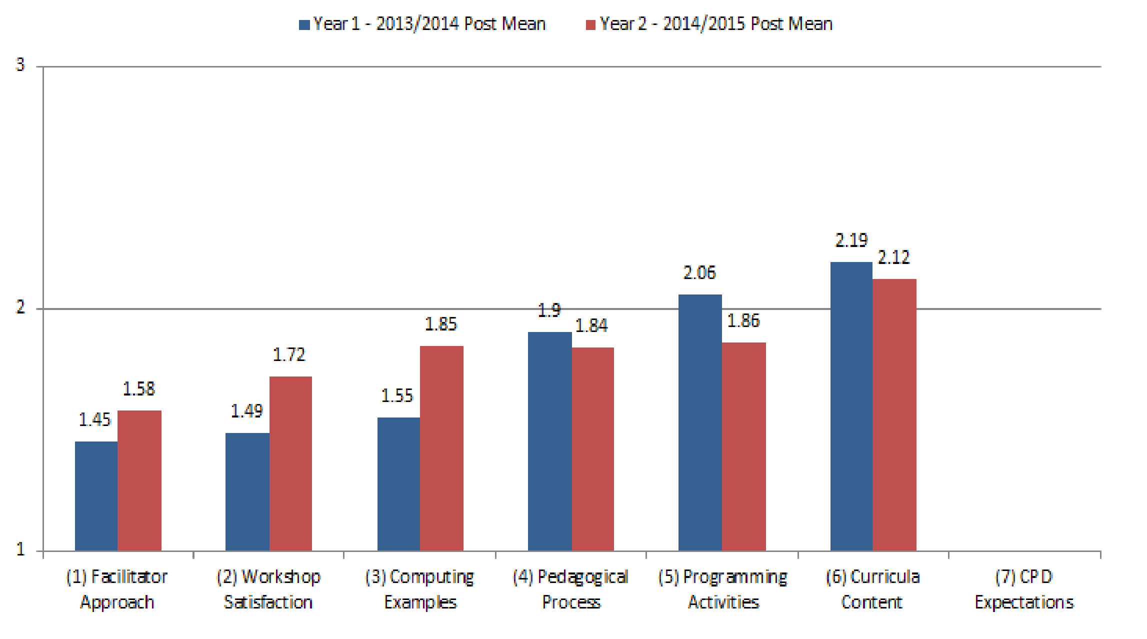
Quantitative analysis of the second year results (Figure 6–5, p. 146) found that 91% of teachers agreed that facilitation was an appropriate CPD method (*m* = 1.58), with 90%

agreeing that that the workshops were worth attending (*m* = 1.72). Also 88% of teachers agreed that the examples were helpful (*m* = 1.85) and 90% of teachers agreed that the pedagogical process was clear (*m* = 1.84). Finally, 90% of teachers agreed that the programming activities were well organised (*m* = 1.86), with 86% of teachers confirming that that the level of difficulty of the content was appropriate (*m* = 2.12).

Comparison of means between year one and year two show that teacher reactions to the workshops remain positive, with the following variations. The year two results show more positive reactions to the programming activities (-0.2), the pedagogical process (-0.06), and the curricula content (-0.07) compared to year one. While and increase in mean for the variables facilitator approach (+0.13), computing examples (+ 0.03) and CPD satisfaction (+ 0.23) suggests the need for more supports for learning computing examples and more directed guidance with activities.

Figure 6-5 Reaction Post-Workshop Variables Year 1 / Year 2

7 Point Likert Scale Results Clustered 1 = Strongly agree; 2 = Agree; 3 = Somewhat agree



(3) Year 3 – 2015/2016

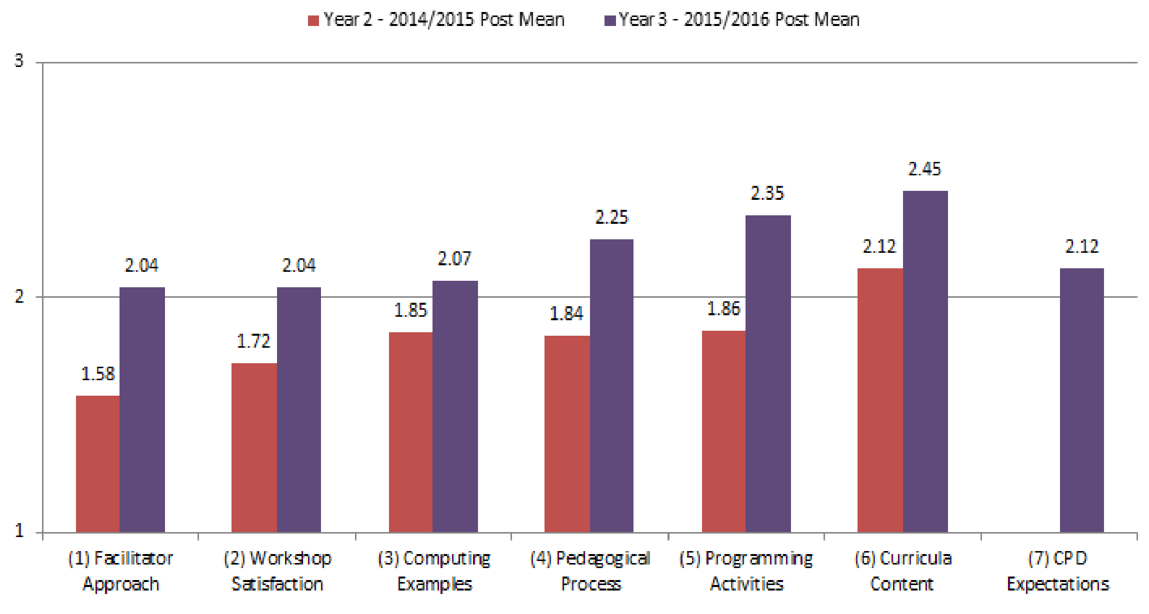
The year three data set includes responses from the second cohort of teachers attending the one-year part time Post Graduate Certificate in 21st Century Teaching and Learning programme. Workshops were again scheduled on Saturdays, with assignment sessions the following Friday.

Quantitative analysis of the third year results (Figure 6–6, p. 147) confirm that 84% of teachers were satisfied with facilitation as a CPD approach (*m* = 2.04) and that 82% of teachers agreed that the workshops were worth attending (*m* = 2.04). Also 83% of teachers agreed that the examples were helpful (*m* = 2.07), with 84% of teachers agreeing that workshops met their CPD expectations (*m* = 2.12). Furthermore, 78% of teachers agreed that the level of difficulty of the content was appropriate (*m*= 2.45), with four fifths (80%) agreeing that the programming activities were easy to understand (*m*= 2.35). Finally, 83% of teachers agreed that the pedagogical process was clearly communicated (*m*= 2.25).

Comparison of means between year two and year three results capture a change in satisfaction across reaction variables compared to the previous year. The largest increase in mean was reported for the facilitator approach (+0.46), programming activities (+0.49) and the pedagogical process (+0.41). Further increases were also reported for teacher satisfaction with the workshops (+0.32), the computing examples (+0.22), and the curricula content (+0.33), with analysis reporting for the first time on variable that the workshops met teachers expectations (*m* = 2.12).

Figure 6-6 Reaction Post-Workshop Variables Year 2 / Year 3

7 Point Likert Scale Results Clustered 1 = Strongly agree; 2 = Agree; 3 = Somewhat agree



(4) Year 4 – 2016/2017

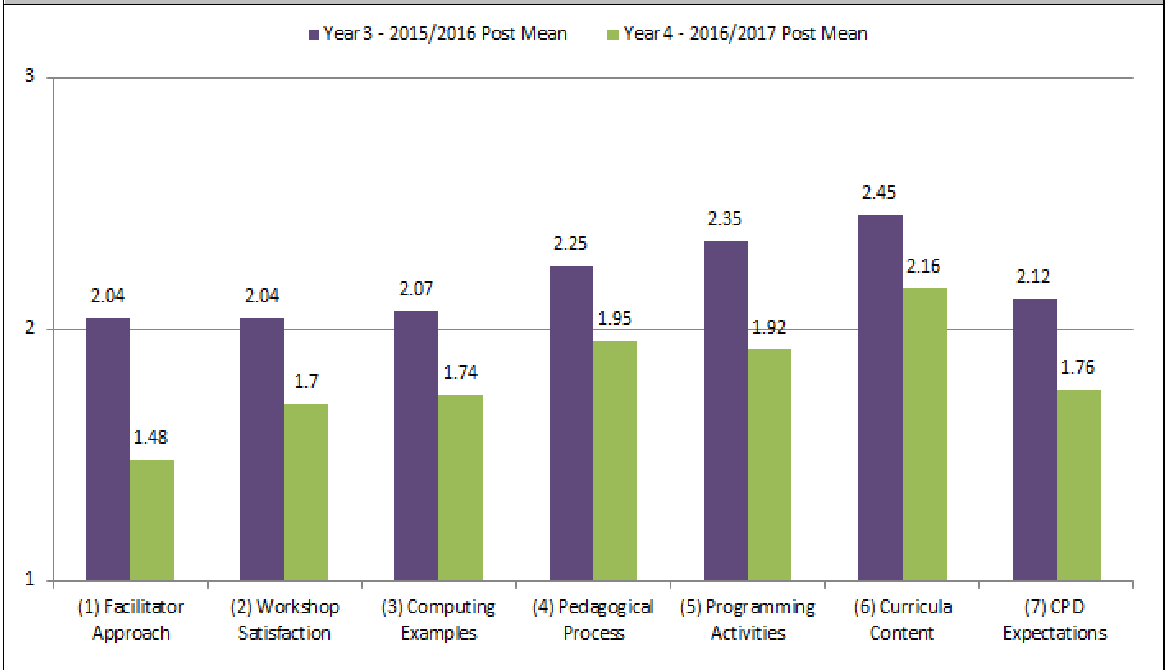
The year four data set includes responses from the third cohort of teachers attending the one-year part time Post Graduate Certificate in 21st Century Teaching and Learning programme. Field notes record adjustments to the CPD structure including providing assignment support sessions after the one-day workshops and the addition of new content into the CPD covering worked examples.

Quantitative analysis of the fourth year results (Figure 6–7) confirm that 93% of teachers agreed that the use of facilitation as an appropriate method of CPD (*m* = 1.48), with 91% agreeing that that the CPD workshop were worth attending (*m*= 1.7). Moreover, a further 91% of teachers agreed that the computing examples were helpful (*m* = 1.74), with 90% agreeing that the pedagogical process was clearly communicated (*m* = 1.95). Also, 93% of teachers agreed that the programming activities were well organised (*m* = 1.92), with 86% of teachers reporting that the difficulty of the content was appropriate (*m* = 2.16). Finally, 91% of teachers agreed that the CPD met their expectations (*m* =1.76).

Comparison of means between year three and year four results report an increase in teacher reactions to the workshops across all variables. The year four results confirm a drop in mean compared with the previous year which indicates a positive reaction to a facilitator approach (-0.56), programming activities (-0.43), and CPD expectations (-0.36) compared to year three. Teacher reactions also remain positive in terms of teacher satisfaction with the workshops (-0.34), the computing examples (-0.33), the pedagogical process (-0.3), and the curricula content (-0.29), which confirm teacher satisfaction across all reaction variables compared to the previous year.

Figure 6-7 Reaction Post-Workshop Variables Year 3 / Year 4

7 Point Likert Scale Results Clustered 1 = Strongly agree; 2 = Agree; 3 = Somewhat agree



(5) Year 5 – 2017/2018

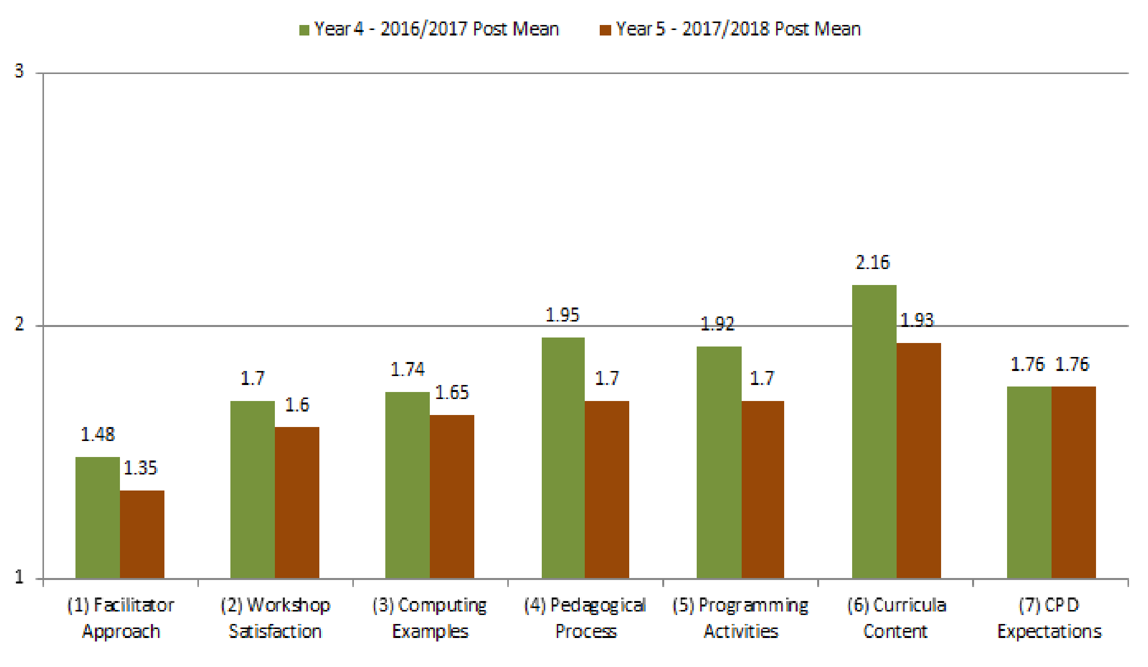
The year 5 data set includes responses from the fourth cohort of teachers attending the one-year part time Post Graduate Certificate in 21st Century Teaching and Learning programme.

The publication of the Draft Curriculum for Leaving Certificate Computer Science (NCCA, 2017) and the draft curricula in Coding for primary schools (NCCA, 2018a) provided new options for teaching CS.

The quantitative analysis of year five results (Figure 6–8) confirm that teacher reactions as very positive. A total of 99% of teachers agreed that facilitation was an appropriate CPD approach (*m* = 1.35), with 97% agreeing that the computing workshops were worth attending (*m* = 1.6). In addition, 97% of teachers agreed that the computing examples / case studies were helpful (*m* = 1.65); and 98% of teachers agreed that the overall process was clearly communicated (*m* = 1.7). Nearly all teachers (99%) strongly agreed that the programming activities were well organised and easy to understand (*m* = 1.7), with a further 91% of teachers agreeing that the level of difficulty of the content was appropriate for them (*m* = 1.93). Finally, 93% of teachers strongly agreed (*m* = 1.76) that the CPD met their expectations.

Figure 6-8 Reaction Post-Workshop Variables Year 4 / Year 5

7 Point Likert Scale Results Clustered 1 = Strongly agree; 2 = Agree; 3 = Somewhat agree



Comparing means between years four and five show a further increase in teacher reactions to the workshops across all variables. The year five results show a stronger positive reaction to a facilitator approach (-0.13) and programming activities (-0.22), with CPD expectation satisfaction levels remaining the same compared to year four. While teacher reactions to the workshops (-0.1), the computing examples (-0.09), the pedagogical process (-

0.25), and the curricula content (-0.23), confirm, continuing teacher satisfaction across all reaction variables compared to the previous year.

6.2.3.1 Summary

In summary, the year on year quantitative evidence reports that teacher reactions to the workshops remained positive across time with 96% of teachers in year 1 agreeing that the workshops were worth attending (*m* = 1.49), compared to 97% in year 5 (*m* = 1.6). Similar patterns are reported for a facilitator approach to CPD, with 96% of teachers agreed that facilitation was an appropriate CPD approach (*m* = 1.45) in year 1, increasing to 99% agreement in year 5 (*m* = 1.35). Teachers also responded positively to the examples that were used in the workshops, with 97% of teachers in year 1 agreeing that the computing examples were helpful (*m* = 1.55), compared to 97% of teachers agreeing that the examples they were helpful in understanding computing content (*m* = 1.65).

Positive reactions are confirmed for the remaining variables. Teachers agreed that the pedagogical process was satisfactory (*m* = 1.9), with 91% of teachers agreeing that the process was clearly communicated and met to their satisfaction in year 1 compared to 98% agreement in year 5 (*m* = 1. 7). The depth of programming activities covered in the workshops were also perceived as satisfactory (*m* = 2.06), with 87% of teachers agreeing that the programming activities were well organised in year one compared with 99% agreement of teachers in year 5 *(m* = 1.7*)*. Teachers also reported that they were very satisfied with the curricula content (*m* = 2.19), with 83% of teachers satisfied with the level of difficulty in year 1, with year 5 results reporting similar findings, with 91% of teachers agreeing that the level of difficulty was appropriate (*m* = 1.93). Finally comparing CPD expectation variables between years 3 and years 5 reveal that in 84% of teachers agreed that the workshops met with their CPD expectations in year 3 (*m* = 2.12), compared with 93% of teachers in year 5 who agreed that the CPD meet their expectations (*m* = 1.76). These results show strong positive reactions to the Bridge21 CS CPD workshop model, with sustained teacher satisfaction.

6.2.4 Qualitative Analysis

The previous sections explored the quantitative reaction data. This section provides the qualitative analysis, which adds context to the quantitative results. The qualitative responses were obtained from a self-selecting sample of *N* = 723 teachers who responded to the question: *What key learning did you take away from today's Bridge21 CPD workshop?* This question was adapted from an existing questionnaire (Kristiansen, 2007) and two themes emerged from coding the responses (Appendix 9.13.1). The first theme (1) voiced teacher reactions to a collaborative approach to learning; and the second theme (2) indicated teacher reactions toward a facilitator-mentoring approach to teaching.

(1) Theme 1 - Teacher Reactions to Collaborative Learning

Teachers enjoyed the experience of collaborative learning (Figure 6–9). Teachers reported that they were satisfied with the experience of sharing ideas and exploring concepts with peers. This is typified in the following examples, where one teacher reported that *"working in groups to solve problems is better and less stressful than trying to figure it out yourself"* (CSCPD_715). A further teacher also reported that they enjoyed the experience of "*working collaboratively, trying to figure it out"* (CSCPD_714). Collaborative working was viewed as "*fun and instructive*" (CSCPD_713), with one teacher reflecting that "*thinking is hard work - groups are good for this*" (CSCPD_630). Learning in a team also meant that teachers could learn computing skills from peers, with one teacher speaking positively about "*taking advantage of the range of skills in the group*" (CSCPD_546).

Figure 6-9 A Team working on a Bridge21 Computing Activity (Fisher et al., 2016)

Furthermore, a view was expressed that "*the group is stronger than the individual*" (CSCPD_588), with a one teacher highlighting the "*importance of learning from other members of your team*" (CSCPD_319) and another reporting on the importance of "*learning from listening to other team members"* (CSCPD_361). A further aspect of learning in a team was the opportunity to "*network with others*" (CSCPD_408). Teachers were more open to exploring new ideas with peers, with one teacher reflecting that they enjoyed "*teamwork - all can learn from each other*" (CSCPD_542). For example one teacher particularly liked the opportunity to observe projects that other teams had worked on, adding that "*it was interesting to see other groups' projects"* (CSCPD_18). In a further example, one teacher reported that they enjoyed observing the work of their peers stating that is "*it's the little things that can make a big difference"* (CSCPD_345). Teachers liked working in a team to learn computing and of "*taking advantage of the range of skills in the group*" (CSCPD_546). Teachers also reacted positively to the autonomy of sharing ideas, with teachers sharing expertise in "*how to code a game (and) how to finish a project*" (CSCPD_206).

(2) Theme 2 - Teacher Reactions to Facilitation

Teachers responded positively to a facilitator approach to CPD. Teachers reported that they were surprised at the impact that a facilitated approach to teaching computing had on their learning. In the following example, one teacher observed that facilitation is a complex teaching methodology and that it involves "*stepping away from learners, not to do it for people"* (CSCPD_531). One teacher provided an example which describes how they observed that facilitation is a practical approach to teaching that involves "*getting students to think about the skill they are using"* (CSCPD_171). A further teacher acknowledged that they needed to make more time in their own lessons to "*give space to students to think"* (CSCPD_172). Another teacher reflected that the workshop experience had helped them to develop the confidence to use facilitation and had learned that "*it's ok to give something a go even if you are not experienced"* (CSCPD_573). The workshop process had, for another teacher, been a 'transformational' process. This particular teacher reflected that they had learned to see their students "*in a different light and that it's not just about me imparting info"* (CSCPD_513). These findings provide examples, which show that teachers enjoyed the experience of learning through facilitation, with teachers enjoying the freedom to ask questions, direct their learning and request help from facilitators when needed. Teachers planned to use a similar approach in their own teaching: "*I enjoyed the group work where we had to analyse the problems and discuss the factors that would be important to consider in solving the problems. I could see that this strategy would be very useful to use with students*" (CSCPD_287).

6.2.5 Discussion

The reaction findings were analysed to address research question Q1.1 what are teachers' reactions to the CPD workshop content. A key finding is the level of positive agreement across all reaction variables over time, which confirms that teachers enjoyed the CPD. A further finding is diversity in age, and workshop representation demonstrates the depth of professional experience and variety in computing workshop content, which was covered in the CPD sample over five years. These findings resonate with the work of Sergis et al. (2018) who suggest that collaboration plays an important role in the process of learning how to engage with computing concepts and practices. Teague and Roe (2008) further suggest that collaborative activities, including project-based learning experiences are essential in helping learners develop the confidence and practical expertise to write computer programmes. While Crook (2018) advises that CPD which encourages learning by doing, with peers, gives learners the opportunity to discuss and to critique what they have learned.

Major et al (2012) reminds us that computing is a complex subject and that introductory computing courses need to use a creative approach to teaching and learning to help learners understand content they may find difficult. The results confirm that teachers reacted positively to the curriculum that was followed, and that the pedagogical process provided teachers with the opportunity to work through computing problems, enabling teachers to gain practical expertise in computing and problem solving, supported by peers with different experiences and knowledge.

To conclude, the reaction findings explored in this section show that teachers responded to a collaborative, project-based and facilitator led approach to learning computing, with teams playing a vital role in helping teachers explore concepts and practice computing tasks supported by their colleagues. These results confirm that a collaborative, project-based and facilitator led, approach to learning computing gives teachers the confidence to take part in computing activities. These results also fall in line with supporters of a collaborative approach to teaching and learning CS (Ben-Ari, 2001; Ridgway & Passey, 1991; Sentance, 2018b), who agree that learning with peers plays an important role in helping teachers develop the content knowledge and the confidence to program.

6.3 Content knowledge developed by teachers

The previous section explored teacher reactions to the workshop content. This section explores the research findings which are mapped to the second Level in Kirkpatrick's (1994) model and examines teacher perceptions of their learning. The learning data set was generated from the administration of a pre and post-workshop mixed methods questionnaire (Appendix 9.7 and Appendix 9.8). The content of the questionnaire was based on the workshop learning outcomes (TCD, 2017). The data was collected from both questionnaires to answer research the question: Q1.2 What content knowledge did teachers learn?

6.3.1 Data Set Limitations

Before exploring the learning findings, the limitations of the data set are discussed. The researcher collected post-workshop scores from *N* = 50 workshops before introducing pre-tests. Analysis of the post-test results indicated that teachers were confident in their learning. However further analysis was needed to understand the impact of the workshops on teacher learning. Thus, pre-tests were piloted, and then phased into data collection to explore change over time. This means that there are more post-test responses in the data set, than there are in pre-test-workshop responses.

To correct this imbalance, the researcher conducted a phased approach to statistical analysis. The first phase used descriptive statistics to explore trends between pre and post-test learning outcome results. The averages of all pre-workshop responses were compared with the average of all post-workshop responses. Having established a difference between pre and post-test results, the researcher embarked on a second phase of statistical analysis. Phase two analysis explored the difference between pre and post-test results, year on year to explore the change in teacher learning outcomes over time; however further testing was required to explain the significance of the change between pre and post-test results.

Phase three involved using the Wilcoxon (1945) Signed Rank Test, in replacement of a *t*-test to explore the statistical significance of the change between pairs of pre and post-test results. The researcher conducted a fourth phase of statistical analysis, again using Wilcoxon (1945) Signed Rank Test to explore change between a first pre-test and last post-test result, thus capturing change over time. These last set of results confirm that that the CPD had a positive impact, that teachers experienced an increase in knowledge across each learning outcome, and that the results were statically significant ($p< .005$). Finally, pattern coding was used to analyse qualitative responses discussed in section 6.3.6. Each phase of the analysis is now discussed in detail.

6.3.2 Phase 1 - Quantitative Analysis – All Years

This analysis covers the sum of all pre and all post-workshop learning outcome responses (Table 32, p. 155 and Table 33, p. 156). Likert scales supporting this analysis are provided in Appendix 9.14.3.

Analysis of the pre-workshop quantitative learning variables shown in Table 32, p. 155 arranged 1 = Strongly agree; 7 = Strongly disagree, suggest that teachers were not initially confident in their ability to plan Bridge21 activities (*m* = 4.84), nor confident in their ability to teaching programming using Bridge21 (*m* = 5.17). In addition, teachers were not confident in their ability to program (*m* = 5.14) nor did teachers think that they understood computing concepts (*m* = 5.12). Teachers also reported that they were unsure of programming processes (*m* = 4.87) and that they lacked computing knowledge (*m* = 4.83). A Cronbach's alpha score of .956 indicates a strong level of internal consistency for the scale.

Table 32 Learning Outcome Pre-Workshop Variables adapted from TCD (2017)					
Sum of all Pre-Workshop Responses					
Cronbach's alpha =. 956 indicating a strong degree of reliability or consistency with this scale					
7 Point Likert Scale [1 = Strongly agree; 7 = Strongly disagree] Measure of Agreement					
Variables					
(1) Planning Bridge21 Activities	(2) Programming Ability	(3) Programming Concepts	(4) Programming Processes	(5) Content Knowledge	(6) Teaching Programming using Bridge21
Responses					
291	293	292	292	293	292
Mean					
4.84	5.14	5.12	4.87	4.83	5.17

Further analysis of the corresponding post-workshop quantitative learning variables shown in Table 33, p. 156 arranged 1 = Strongly agree to 7 = Strongly disagree, confirm that teachers reported that they were more confident in their ability to plan Bridge21 activities (*m* = 2.7); and more confident in their ability to program (*m* = 2.59). Teachers also reported that they were more confident in their ability to teach programming using the Bridge21 model (*m* = 2.79) and that they had a greater understanding of basic concepts (*m* =2.56). Furthermore, teachers reported that they were more sure of programming processes (*m* = 2.42) and gained in content knowledge for teaching computing (*m* = 2.8). A Cronbach's alpha score of .931 indicates a strong level of internal consistency for this scale. Each of these variables are explained in turn, starting with (1) Planning Bridge21 Activities, with extracts from field notes included to provide context to the discussion of the quantitative results.

The following diagrams (Figure 6-10 and Figure 6-11 on p. 157) provide a visual representation of the level of agreement linked to each reported mean value. The Likert scales supporting the following analysis are based on a measure of agreement and arranged 1 =

Strongly agree to 7 = Strongly disagree, with corresponding qualitative analysis results reported in Appendix 9.13.4.

Table 33 Learning Outcome Post-Workshop Variables adapted from TCD (2017)					
Sum of all Post-workshop Responses					
Cronbach's alpha =.931 indicating a strong degree of reliability or consistency with this scale					
7 Point Likert Scale [1 = Strongly agree; 7 = Strongly disagree] Measure of Agreement					
(1) Planning Bridge21 Activities	(2) Programming Ability	(3) Programming Concepts	(4) Programming Processes	(5) Content Knowledge	(6) Teaching Programming using Bridge21
Responses					
431	431	430	431	431	427
Means					
2.7	2.59	2.56	2.42	2.8	2.79

(1) Variable - Planning Bridge21 Computing Activities

These pre-workshop results (Figure 6-10, p. 157) show that 27% pf teachers agreed that they were confident in their ability to plan Bridge21 activities, with 15% neutral in their view and the remaining 58% stating that they were not prepared. In contrast (Figure 6-11, p. 157) the post results show that over three quarters of teachers (76%) agreed that they were confident in their ability to plan Bridge21 computing activities, reporting an increase in confidence to use the Bridge21 model, with 13% of teachers neutral in their view and the remaining 11% still stating they were not confident. Further analysis of field notes confirm that the Bridge21 processes supported the use of computing activities that teachers could be used to motivate students: *"it's important to build in a sense of achievement into the process"* (CSCPD_38). Bridge21 also provided a process of supporting team-based problem solving, that teachers could use to organise students into groups so that they could support each other in their learning: *"I'll use the Bridge21 model to help in my approach to problem solving in my present role. It's also given me new methods of working with teams and groups"* (CSCPD_84).

Figure 6-10 Pre Workshop Learning Outcome Variable Analysis – Summary

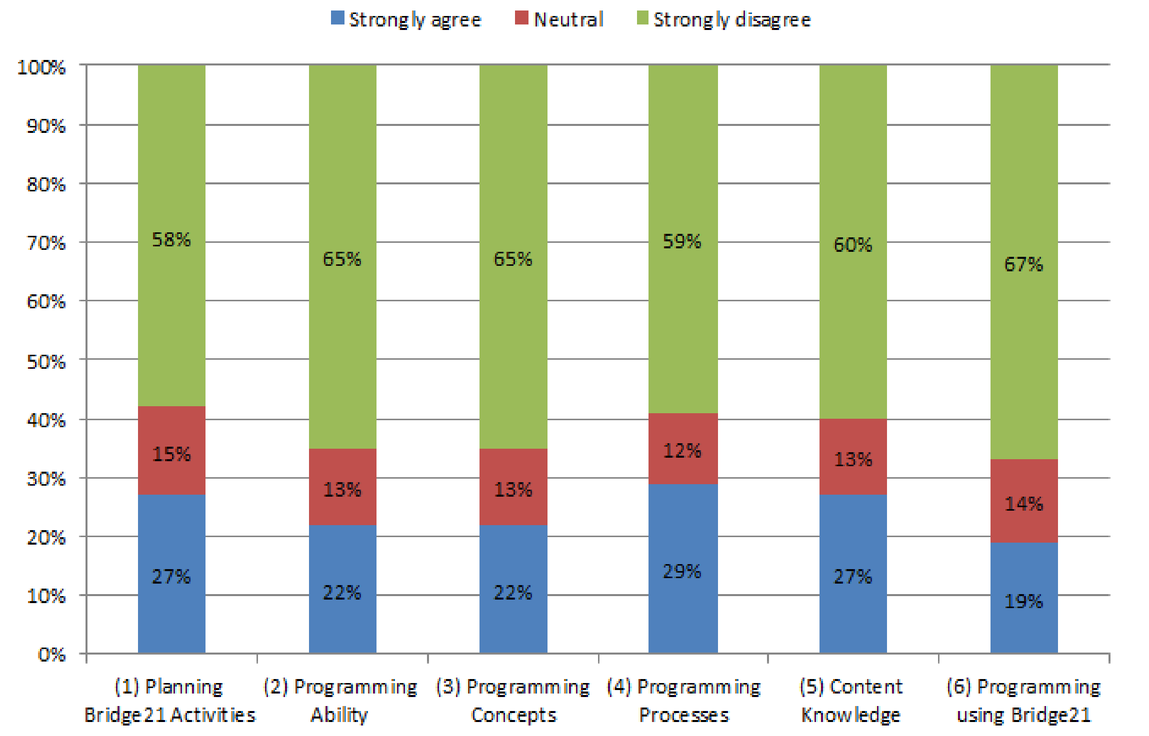
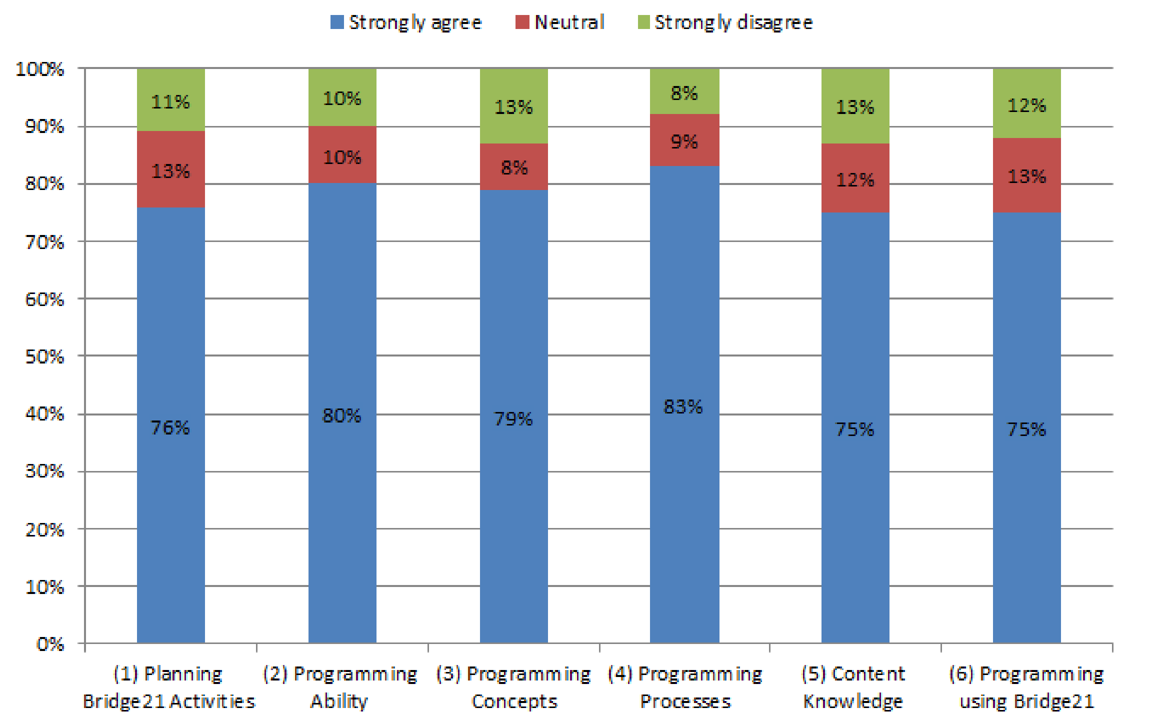


Figure 6-11 Post Workshop Learning Outcome Variable Analysis – Summary



(2) Variable - Planning Bridge21 Computing Activities

These pre-workshop results (Figure 6-10, p. 157) show that 27% pf teachers agreed that they were confident in their ability to plan Bridge21 activities, with 15% neutral in their view and the remaining 58% stating that they were not prepared. In contrast (Figure 6-11, p. 157) the post results show that over three quarters of teachers (76%) agreed that they were confident in their ability to plan Bridge21 computing activities, reporting an increase in confidence to use the Bridge21 model, with 13% of teachers neutral in their view and the remaining 11% still stating they were not confident. Further analysis of field notes confirm that the Bridge21 processes supported the use of computing activities that teachers could be used to motivate students: "*it's important to build in a sense of achievement into the process"* (CSCPD_38). Bridge21 also provided a process of supporting team-based problem solving, that teachers could use to organise students into groups so that they could support each other in their learning: "*I'll use the Bridge21 model to help in my approach to problem solving in my present role. It's also given me new methods of working with teams and groups"* (CSCPD_84).

(3) Variable - Programming Ability

These pre-workshop results (Figure 6-10, p. 157) show that 65% of teachers reported a lack of confidence in their ability to program, with 13% neutral in their option with the remaining 22% somewhat confident in their ability to program. The post-workshop results (Figure 6-11, p. 157) report a tenth of teachers (10%) registering a lack of confidence in their ability to program, with 10% neutral in their option, and the remaining four fifths of teachers 80% confident in their ability to program after the CPD. Further analysis of field notes confirm that the workshop experience using teamwork, supported confidence building: "*my computational thinking abilities have grown considerably. Revisiting Scratch and the Raspberry Pi have given me confidence and the introduction to Python has encouraged me to learn"* (CSCPD_232). Working with peers helped teachers reinforce concepts: "*I worked well with partner to complete tasks. Felt more confident and able to do tasks"* (CSCPD_152).Peer led learning, helped teachers build confidence in programming "*I will be able to give my opinion on python uses to colleagues. I will also investigate Python myself"* (CSCPD_151).

(4) Variable - Programming Concepts

These pre-workshop results (Figure 6-10, p. 157) show that 65% of teachers agreed that they were not confident in understanding basic concepts, with 13% neutral in their view, and 22% agreeing that they had a greater understanding of basic concepts. The post-workshop results (Figure 6-11, p. 157) indicate 79% of teachers agreed that they had a greater understanding of basic concepts, with 8% neutral in their view and 13% disagreeing that they achieved a greater understanding of concepts. Further analysis of field notes confirm that teachers perceived programming as difficult, giving the example that "*writing code needs high concentration levels*" (CSCPD_239), but peer collaboration helping to give meaning to concepts: "*I enjoyed the discussion about computing concepts such as data collection and abstraction and the introduction to the Python programming language through Scratch"* (CSCPD_252). Encouragement from peers motivated teachers to engage with complex concepts, with one teacher giving the example: "*I enjoyed getting programmes to work in Python and I enjoyed the challenge. My colleagues were eager and enthusiastic*" (CSCPD_259).

(5) Variable - Programming Processes

These pre-workshop results (Figure 6-10, p. 157) show that that 59% of teachers agreed that they were not confident in understanding basic programming processes such as concurrency or initialisation, with 12% neutral in their view, and 29% agreeing that they understood basic programming processes. The analysis of corresponding post-workshop variables (Figure 6-11, p. 157) indicate that 83% of teachers agreed that they had a greater understanding of programming processes, with 9% neutral in their view and 8% disagreeing that they achieved a greater understanding of basic programming processes. Further analysis of field notes confirm that teachers reported a better understanding of computing processes, with one teacher reporting that the CPD had helped them to understand "*how to initialize a sprite How to insert a loop how to insert a sound"* (CSCPD_2095), with a further teacher understanding "*concurrency and separating out different elements of coding*" (CSCPD_1855). Teachers also reported understanding "*operators and variables, and using the forever loop"* (CSCPD_2829), with further examples demonstrating use of "*variables "if" or "else*" (CSCPD_3027), and understanding "*the importance of reading code to identify errors and fix errors*" (CSCPD_2196).

(6) Variable - Content Knowledge

These pre-workshop results (Figure 6-10, p. 157) show that that 60% of teachers agreed that they were not confident in their ability to teach computing content, with 13% neutral in their view, and 27% agreeing that they were confident in their ability to teach computing content. In contrast, the analysis of corresponding post-workshop variables (Figure 6-11, p. 157) indicate that 75% of teachers agreed that they were confident in their ability to use their computing knowledge to teach computing content, with 12% neutral in their view, and 13% agreeing that they were not confident in their ability. Further analysis of field notes confirm that the workshops provided tasks which enabled teachers to make connections between computing concepts which they would use in teaching: "*I'm finally beginning to see the relationship between setting up a program, to automate a physical process (for example* *wiring)*" (CSCPD_340). Peer learning helped teachers develop ideas they could integrate into classroom activities: *"I enjoyed working in teams. I met like-minded people and I got lots of ideas from other teachers"* (CSCPD_215). A further teacher proposed a plan outlining the content they wanted to teach: "*I will integrate Scratch into my daily teaching through project work for my more visual learners. I will also share Scratch's cross-curricula usefulness with my colleagues*" (CSCPD_82).

(7) Variable - Teaching Programming using Bridge21

Finally, these pre-workshop results (Figure 6-10, p. 157) show that 67% of teachers agreed that they perceived themselves as not confident in their ability to use the Bridge21 model for teaching programming, with 19% of teachers agreeing that they were confident in using the Bridge21 model, with a further 14% of teachers neutral in their view. In contrast, the post-workshop results (Figure 6-11, p. 157) confirm that three quarters of teachers (75%) agreed that they were confident in their ability to teach programming using the Bridge21 model, with 13% neutral responses, and the remaining 12% disagreeing they were more confident in using the Bridge21 model. Further analysis of field notes show that teachers were motivated to use Bridge21 for introducing computing: "*I certainly liked the Bridge21 method of learning and would encourage that way with groups. I'll look at some of the software again and I can share my current / past experiences of teaching and using computers"* (CSCPD_264). Teachers also shared plans describing use of the Bridge21 model in a school context: "*I will be applying to teach an introduction to scratch computer programming in my 1st and 2nd year computer studies classes. I can share what I have learnt with other computer studies teachers"* (CSCPD_129). One teacher planned on using elements of the Bridge21 model "*I'm going to introduce aspects of Bridge21 model to programming class"* (CSCPD_267), while another teacher planned on using teamwork to give students more control: "*I will be aware of my teaching methods and delegating roles and not taking over if the students are doing a task.*" (CSCPD_283).

6.3.2.1 Summary

In summary, the descriptive statistics evidence reports that, collaborative, project-based, and facilitator led workshop experiences helped teachers quickly prepare for planning and implementing Bridge21 activities. Over half of teachers (58%) agreed that they were not confident in their ability to plan Bridge21 activities prior to the workshop (m = 4.84), in contrast to 76% of teachers reporting in the post-workshop results that they were more confident in planning activities using the Bridge21 model (m = 2.7). Furthermore, 65% of teachers reported that they were not confident in their ability to program (m = 5.14). In contrast to this, the post workshop results show that 80% of teachers agreed that they felt confident in their ability to program (*m* = 2.59), with teachers reporting that they were more confident programmers after the CPD. Also 83% of teachers agreed that they had a greater understanding of programming processes after attending the computing workshops (*m* = 2.42), compared to 59% of responses captured before (*m* = 4.87).

Before the workshops, 65% of teachers indicated that they did not have basic understanding of computer programming concepts (*m* = 5.12). After the workshop and in contrast to this, 79% of teachers agreed that they had developed a greater understanding of basic computer programming concepts (*m* = 2.56). Prior to completing the CPD, 67% of teachers reported that they did not feel confident in their ability to teach computer programming using the Bridge21 model (*m* = 5.17). After completing the workshops, 75% of teachers agreed that they were confident in their ability to teach computer programming using the Bridge21 model of 21st century learning (*m* = 2.79). Finally, 60% of teachers agreed that they were not confident in their computing knowledge (*m* = 4.83), compared to 75% of teachers agreeing they were more confident after the workshops (*m* = 2.8). Having used descriptive statistics to identity a change between pre and post-test responses, the next Section uses the Wilcoxon test to explore the significance of this change on a workshop basis.

6.3.3 Phase 2 – Quantitative Analysis – Year on Year

The previous section provided an aggregated view of the learning outcome findings. Further analysis exploring the same data set, year on year; provides the capacity to explore change over time. The following section explores the learning outcome findings, organised by year starting with Year 2. Table 34, p. 162 provides a summary of mean values reported year on year, used in the following analysis. Likert scales supporting this analysis are reported in Appendix 9.14.4.

(1) Year 2 Learning Outcome Variables – 2014/2015

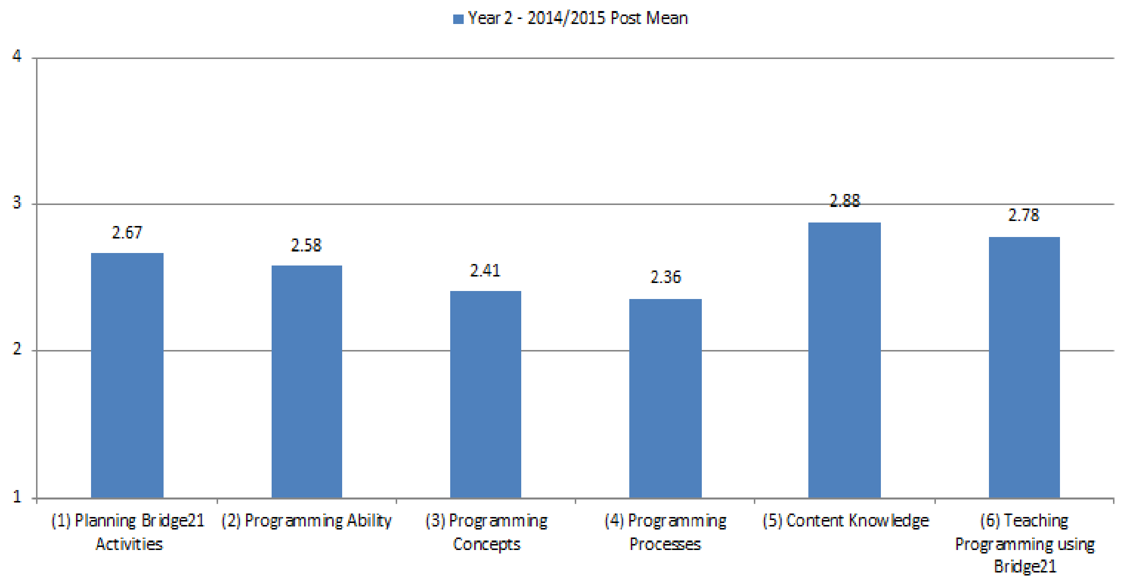
The learning outcome questionnaire was phased into the workshop data collection model in the second year of CPD data collection. The post-workshop learning outcome data was collected to correspond with the collection of the reaction data set at the end of workshops. The post-workshop learning outcome data set includes responses from the first cohort of teachers attending the one year, part time Post Graduate Certificate in 21st Century Teaching and Learning programme.

Table 34 Learning Outcome Variables – Mean – Year on Year						
7 Point Likert Scale [1 = Strongly agree; 7 = Strongly disagree] Measure of Agreement						
Variables						
	(1) Planning Bridge21 Activities	(2) Programming Ability	(3) Programming Concepts	(4) Programming Processes	(5) Content Knowledge	(6) Teaching Programming using Bridge21
Year 2 - 2014/2015 Pre and Post Workshop Responses and Means						
Pre	-	-	-	-	-	-
Mean	-	-	-	-	-	-
Post	81	81	81	81	81	78
Mean	2.67	2.58	2.41	2.36	2.88	2.78
Year 3 - 2015/2016 Pre and Post Workshop Responses and Means						
Pre	-	-	-	-	-	-
Mean	-	-	-	-	-	-
Post	102	102	102	102	102	101
Mean	2.88	2.68	2.67	2.51	2.93	2.83
Year 4 - 2016/2017 Pre and Post Workshop Responses and Means						
Pre	210	212	211	211	212	211
Mean	5.01	5.43	5.35	5.15	5.03	5.36
Post	171	171	170	171	171	171
Mean	2.8	2.75	2.74	2.58	2.85	2.85
Year 5 - 2017/2018 Pre and Post Workshop Responses and Means						
Pre	81	81	81	81	81	81
Mean	4.38	4.37	4.53	4.12	4.28	4.69
Post	77	77	77	77	77	77
Mean	2.30	2.14	2.16	2.01	2.45	2.58

Quantitative analysis (Figure 6-12) found that 80% of teachers agreed that they were confident in their ability to plan computing activities according to the Bridge21 model (*m* = 2.67), with 83% of teachers reporting that they were confident in their ability to program (*m* = 2.58). Moreover, 85% of teachers agreed that they had a greater understanding of basic programming concepts after attending the CPD (*m* = 2.41), with 84% of teachers agreeing that they had a greater understanding of basic programming processes (*m* = 2.36). Finally, 77% of teachers confirmed that they felt more confident in their ability to use their computing knowledge to teach CS (*m* = 2.88), with 77% of teachers confident in their ability to teach computing using the Bridge21 model (*m* = 2.78). This data set provided a baseline.

Figure 6-12 Learning Outcome Post-Workshop Variables Year 2

7 Point Likert Scale Results Clustered 1 = Strongly agree; 2 = Agree; 3 = Somewhat agree; 4 = Neither agree nor disagree



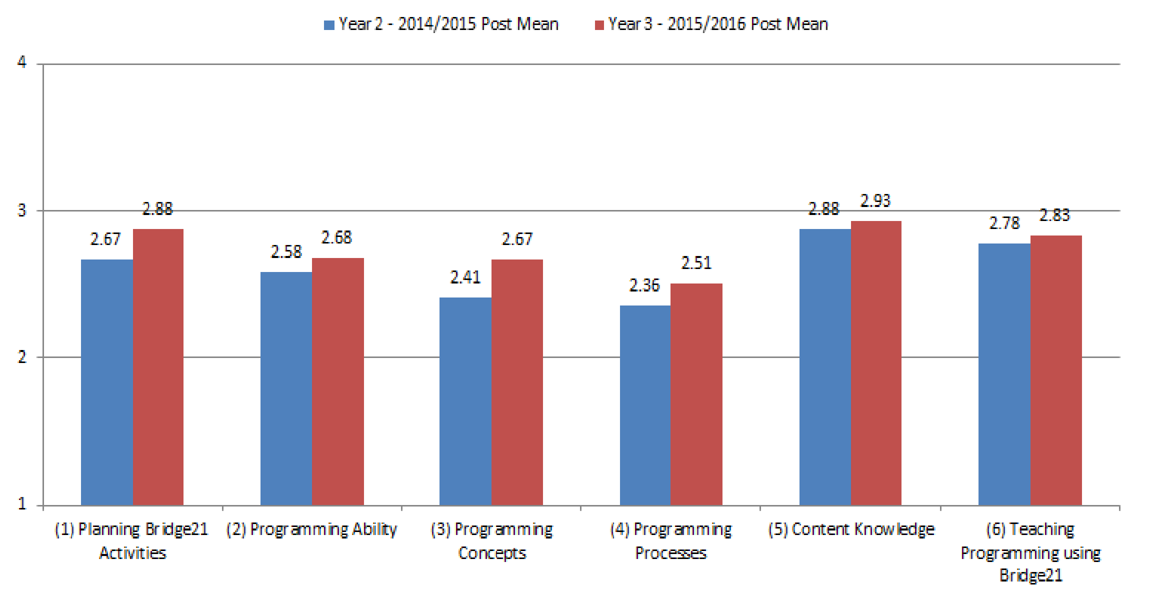
(2) Year 3 Learning Outcome Variables – 2015/2016

The post-workshop learning outcome questionnaire was administered at the end of the computing workshops for a second year providing a basis for comparison with the previous year. The following analysis compares two sets of post-test learning outcome data to explore teacher attitudes to their learning. This data set includes responses from the second cohort of teachers attending the one year, part time Post Graduate Certificate in 21st Century Teaching and Learning programme.

Quantitative analysis comparing year two and year three post-workshop learning outcome variables (Figure 6-13, p. 164) reports a slight increase in mean for programming concepts (+0.26), indicating that teachers were less confident compared to the previous year, with 75% of teachers agreeing that they understood basic concepts, with 8% neutral in their view, and the remaining 17% reporting disagreement in year 3. An increase in mean was also reported for planning Bridge21 activities (+0.21); with 68 % of teachers agreeing that they were confident in their ability to plan Bridge21 computing activities, with 16% neutral in their view, and a further 16% confident in their planning ability in relation to Bridge21. Teacher attitudes to the programming processes used in the workshops also reported a slight increase in mean (+0.15), with 79% of teachers reporting a greater understanding of basic programming processes, with 10% neutral in their view, and the remaining 11% reporting disagreement.

Figure 6-13 Learning Outcome Post-Workshop Variables Year 2 / Year 3

7 Point Likert Scale Results Clustered 1 = Strongly agree; 2 = Agree; 3 = Somewhat agree; 4 = Neither agree nor disagree



A slight increase in mean was reported for the content knowledge (+0.05), with 68% of teachers reporting that they were confident in their ability to teach computer programming content, with 17% neutral in their view, and a further 15% registering disagreement. Analysis of the teaching Bridge21 computing also recorded a slight increase in mean (+0.05), with 71% of teachers agreeing that they were confident in their ability to use the Bridge21 model in the CS classroom, with 12% neural in their view, with 17% in disagreement. Finally, analysis of the programming ability variable reported a slight increase in mean of (+0.1), with 78% of teachers confident in their ability to program, with 10% neural in their view, and 12% not confident. These results signalled a need for pre-tests to be added to the data set in order to explore the impact of the self-reported increases in means across the year three data set.

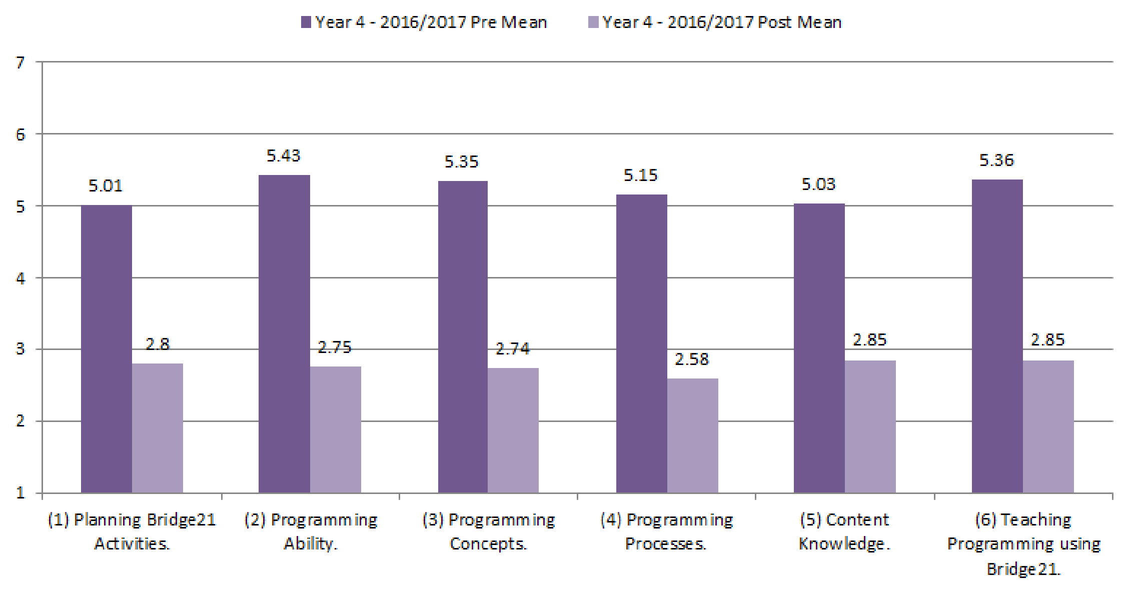
(3) Year 4 Learning Outcome Variables – 2016/2017

The inclusion of pre-learning outcome questions was phased into the workshop data collection model to complement the post-test results. These result provided insight into teacher perceptions of their learning before and after the workshops according to the workshop learning outcomes.

Quantitative analysis (Figure 6-14) comparing pre and post results in Year 4 reported the largest decrease in mean reported for programming ability (-2.68), with 75% of teachers agreeing that they were more confident after attending the workshops, with 16% neutral in their view, and 9% disagreement. The results also report a decrease in mean for understanding programming concepts (-2.61), with 75% of teachers agreeing that they had a greater understanding of basic concepts. A decrease in mean was also reported for programming processes (-2.57), with 80% of teachers reporting that they had a greater understanding of programming processes. Moreover, 70% of teachers also reported that they were more confident in their ability to teach programming using the Bridge21 model (-2. 51). Further mean decreases were reported for planning Bridge21 Activities (-2.21), with 72% teachers agreeing that they were confident in their ability to plan Bridge21 activities. Finally, 74% of teachers agreed that they had greater knowledge for teaching programming reflected in a decrease in mean of -2.18.

Figure 6-14 Learning Outcome Pre and Post-Workshop Variables Year 4

7 Point Likert Scale Results [1 = Strongly agree; 7 = Strongly disagree]



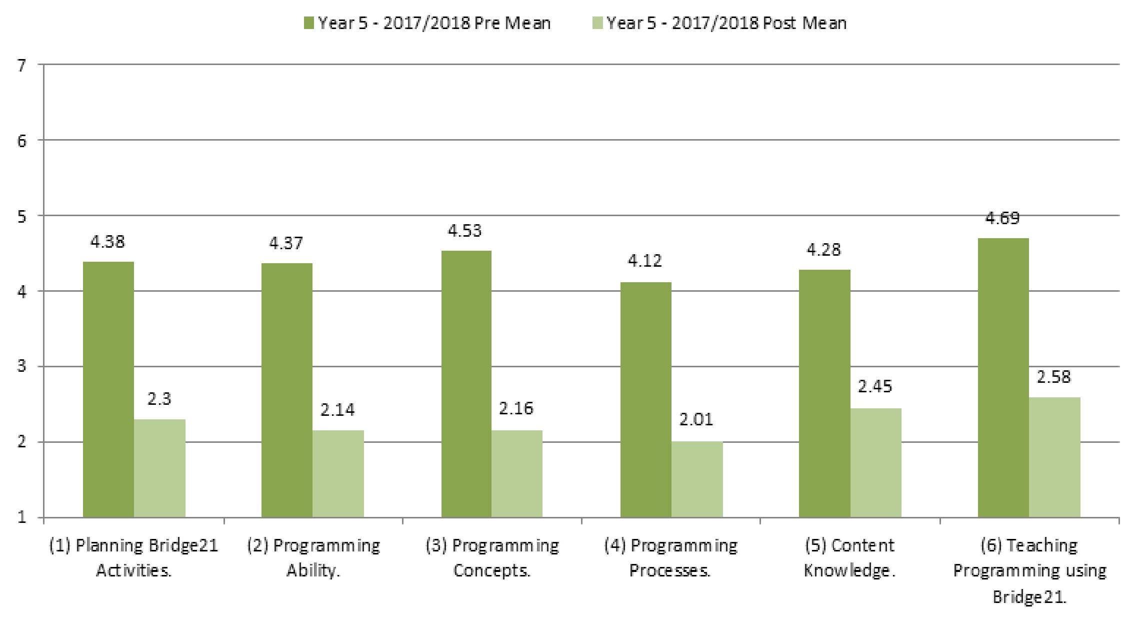
(4) Year 5 Learning Outcome Variables - 2017/2018

The 2017/2018 data set includes pre and post workshop responses from teachers attending the four year of the Post Graduate Certificate in 21st Century Teaching and Learning programme. These results provide further insight into teacher perceptions of their learning before and after the workshops.

Quantitative analysis (Figure 6-15) comparing pre and post results in Year 5 captured the biggest decrease in mean for programming concepts(-2.37), with 89% of teachers agreeing that they had a greater understanding of basic computer programming concepts, with 3 % neutral in their view, and 8% disagreement. Teachers also reported a decrease in mean for programming ability, (-2.23), with 93% of teachers agreeing that they were more confident in my ability to program, with 2% neutral in their view, and 5% disagreement. Moreover, 94% of teachers agreed that they had a greater understanding of basic computer programming processes after attending the workshops, with 1% neutral in their view, and 5% disagreement (-2.11). Also 84% of teachers agreed that they were more confident in their ability to teach computer programming using the Bridge21 model (-2.11). Teachers were also confident in their ability plan Bridge21 activities with 93% of teachers agreeing that they were confident in planning Bridge21 activities, 3% neutral in their view, and 4% disagreement (-2.08). Finally, 90% of teachers reported that they were more confident in their ability use their computing knowledge teach computing, with 10% disagreement (-1.83).

Figure 6-15 Learning Outcome Pre and Post-Workshop Variables Year 5

7 Point Likert Scale Results [1 = Strongly agree; 7 = Strongly disagree]



6.3.3.1 Summary

There were a number of factors, which were perceived to impact upon the learning outcomes data set over five-year research period. The post-workshop learning outcome questionnaire was phased into the workshop data collection model in 2014/2015, which corresponds with the first year of the CPD programme, to baseline teacher perceptions of their learning. Quantitative analysis shows that 85% of teachers agreed that they had a greater understanding of basic programming concepts, with 84% of teachers also agreeing that they had a greater understanding of computer programming processes. Finally, 77% of teachers reported that they were confident in their ability to teach computing using the Bridge21 model after attending the workshops.

Further collection of post-workshop learning outcome data in the second year of the certificate programme (2015/2016), provided further opportunity to compare two sets of post-test learning outcome data between certificate years one and two. Analysis showed that 78% of teachers reported that they were confident in their ability to program, with 79% of teachers reporting a greater understanding of basic programming processes and 71% of teachers agreed that they were confident in their ability to use the Bridge21 model in the CS classroom. Policy factors combined with assistance with implementations are potential factors influencing the slight increase of means.

The inclusion of pre-learning outcome questions into the workshop data collection model in the third year of the certificate CPD programme (2016/2017), complemented post-test learning outcome results. Comparing means between pre and post-test workshop learning outcome variables for 2016/2017 captured positive change in teacher attitudes (Figure 6-14, p. 165).The biggest drop in mean was reported for programming ability (decrease in mean of -2.68 to *m* = 2.75), with 75% of teachers agreeing that they were more confident in their programming ability after attending the CPD workshops. Teachers also reported an increase in confidence in teaching programming using Bridge21, with 70% of teachers confident in their ability use the Bridge21 model. Similar changes were reported for teacher confidence in their ability to apply their computing content knowledge, with 74% of teachers reporting that they gained in content knowledge after attending the CPD.

Finally, the analysis of pre and post learning outcome variables for the 2017/2018 academic year confirm teachers as very positive in their attitudes towards using the Bridge21 model, with 93% of teachers agreeing that they were confident in planning Bridge21 activities (*m* = 2.3), and 84% of teachers agreed that they were confident in their ability to teach computer programming using the Bridge21 model (*m* = 2.58). To conclude, 90% of teachers reported that they were confident in their ability use their computing content knowledge teach computing, which is demonstrated through a decrease in mean of from m = 4.28 to m = 2.45 between pre and post learning outcome results.

6.3.4 Phase 3 - Quantitative Analysis – Wilcoxon All Pre / All Post Results

Descriptive statistics covered in the last section highlight a change between pre-test and post-test scores, with teachers self-reporting changes in attitudes in learning outcome variables. However, further analysis was required to verify the 'validity' of this change. The researcher conducted a Wilcoxon (1945) test on pair matched pre and post learning outcome responses organised per workshop module type. Table 35 provides the number of pair matched learning outcome responses, which were included in the Wilcoxon analysis. As discussed in Section 5.7.1.1 Wilcoxon (1945) is used to calculate the statistical significance of an increase or a decrease between the results of a related or matched pairs of responses (Whitley & Ball, 2002). The tables supporting the following analysis are listed in Appendix 9.14.5.

Table 35 Module Workshop and Paired Matched Sub-Sample Size

Workshop Matched Pre and Post Learning Outcome Pairs per Module	Paired Matched Sample Size
TA21-Mod-2: Problem Solving in the 21st Century (Computational Thinking)	47
TA21-Mod-3: Introduction to programming (Scratch 1: Introduction & Animation)	89
TA21-Mod-4: Intermedia programming (Scratch 2: Game design)	35
TA21-Mod-5: Exploring Computer Systems (Raspberry Pi 1: Introduction)	16
TA21-Mod-6: Text-based Programming (Python 1: Introduction)	2
Total Matched Pre and Post Learning Outcome Pairs per Module	189

6.3.4.1 Limitations with the Data Set

There are a number of limitations with this data set. Teachers attended the same workshop at different times and teachers attended a variety of workshops. The sample is N = 189, as not all participants gave their consent to be involved in data collection and there were a number of pre and post responses which could not be matched. There were only two pair matched teacher responses for TA21-Mod-6: Text-based Programming (Python 1: Introduction) module. This sample is too small for analysis and is therefore excluded from reporting. Finally, the Wilcoxon (1945) Signed Rank Test is based on the null hypothesis that there is no statistical difference between two measurements (in terms of pre and post matched results), and this position is assumed in the following reporting.

(1) TA21-Mod – 2: Problem Solving in the 21st Century

The Wilcoxon (1945) test was carried out on a sample of *N* = 47 matched pre and post responses to determine statistical difference between results. Analysis was based on the outcome of results gained from the administration of a 7 point Likert scale arranged 1 = Strongly agree to 7 = Strongly disagree. Analysis of pre and post median scores shows that the post median scores were significantly lower than those measured before the workshop, and that this change is statistically significant (p<.005). The variable 'programming concepts' reported the largest drop in mean of three points between pre and post variables (-3). The remaining post-workshop learning outcome results confirm a two point drop in median scores between pre and post results (-2), with teachers reporting that they were more confident in their ability to program (-2); more confident in applying computing content knowledge (-2); had developed a greater understanding computing processes (-2); were more confident in planning Bridge21 activities (-2), and were more confident in using the model for teaching computing (-2).

(2) TA21-Mod- 3: Introduction to programming

Testing using Wilcoxon (1945) was carried out on a sample of *N* = 89 matched pre and post responses to determine statistical difference between results. Analysis was based on the outcome of results gained from the administration of a 7 point Likert scale arranged 1 = Strongly agree to 7 = Strongly disagree. The analysis of pre and post results show that the post median scores were significantly lower than those measured before the workshop, and that this change is statistically significant (p<.005). The variables 'programming concepts' (-5) and 'programming ability' (-4) demonstrated the biggest drop between pre and post-workshop median scores, indicating the largest growth in confidence. Each of the remaining variables also reported a drop in median scores between pre and post results of three points each, indicating that teachers were more confident in their ability to apply computing content knowledge (-3) and understand computing processes (-3), and were more confident in their ability to plan Bridge21 activities (-3) and use the model to teach computing (-3).

(3) TA21-Mod - 4: Intermediate programming

Testing using Wilcoxon (1945) was carried out on a sample of *N* = 35 matched pre and post responses. Analysis was based on the outcome of results gained from the administration of a 7 point Likert scale arranged 1 = Strongly agree to 7 = Strongly disagree. The analysis of pre and post results showed that the post median scores were significantly lower than those measured before the workshop, and that this change is statistically significant (p<.005). The learning outcome variables' 'teaching programming using Bridge21' (-4), 'programming concepts' (-4), 'programming ability' (-4), content knowledge (-4) and programming processes (-4) recorded the biggest decreases. These results suggest that teachers are gaining in confidence, through more exposure to the CPD content and methods. A smaller change in pre and post median scores was reported for planning Bridge21 activities (-2), with this need addressed through attending follow up assignment support sessions covering planning and implementation design. While a smaller median difference, this still indicates a growth in confidence.

(4) TA21-Mod- 5: Exploring Computer Systems

Testing using Wilcoxon (1945) was carried out on a sample of *N* = 16 matched pre and post responses. Analysis was based on the outcome of results gained from the administration of a 7 point Likert scale arranged 1 = Strongly agree to 7 = Strongly disagree. The analysis of pre and post learning outcome results showed that the post results were significantly lower than those measured before the workshop, and that this change is statistically significant (p<.005). The variables 'programming concepts' (-5.5), 'programming ability' (-5), and 'teaching programming using Bridge21' (-5), demonstrated the biggest decreases, indicating the largest growth in confidence and ability. Teachers also reported a growth in confidence in ability to use computing content knowledge (-4) and understanding of computing processes (-4) as well as planning to use the Bridge21 model to teach programming (-4).

6.3.4.2 Summary

In summary, this quantitative analysis confirms a median decrease across post-workshop learning outcome variables, which indicates an increase in teacher confidence. Statistical analysis using Wilcoxon (1945) Signed Rank Testing, confirms that teachers attending Raspberry Pi workshops gained most in programming content knowledge and confidence, as well as computing expertise in their preparation to use the Bridge21 model for teaching computing. This is reflected in a *five point decrease between pre and post-workshop median scores* (Appendix 9.14.5) which indicates that teachers were very confident in their ability to use the content and methods covered in the Raspberry Pi workshop. Teachers also reported increase confidence in their ability to implement Scratch content with median increases reported in teachers programming ability and understanding of programming concepts.

6.3.5 Phase 4 - Quantitative Analysis – Wilcoxon First Pre / Last Post Response

Different teachers attended different workshops at different times and not all teachers attended the same number of workshops. The previous section explored results organised by module, comparing pre- and post-workshop responses to explore the significance of the change in teacher attitude.

This section explores perceived change in teacher attitudes between their first and last learning outcome workshop questionnaire responses. Analysis of the first response and last response aims to capture the significance of the change in reported responses over time. There are less first pre and last post responses in the data set, given that not all teachers completed a pre or post responses during the workshops, Questionnaire completion in all workshops was optional.

6.3.5.1 Wilcoxon Results

Statistical analysis using Wilcoxon (1945) Signed Rank Testing confirmed an increase in teacher content knowledge and confidence across all learning outcome variables. The Wilcoxon test confirmed a drop in scores of four points between all pair matched pre and post-workshop responses across all learning outcomes. Table 36, p. 172 provides the pre workshop learning outcome median scores, and Table 37, p. 172 provides the corresponding post workshop learning outcome median scores. These results confirm that teachers reported that they were more confident in their ability to plan Bridge21 activities (-4), and that they perceived themselves as more confident in their programming ability (-4). Teachers also reported that they had a greater understanding of programming concepts (-4), and that teachers are more confident in their ability to teach programming using the Bridge21 model (-4). Teachers were also more confident in their ability to use computing knowledge (-4), and were more confident in understanding processes (-4).

Placing these results in context, the Wilcoxon (1945) Signed Rank Test provides evidence of a change in median between the pair matched pre and post reported results. Whitley and Ball (2002) advise that the strength in using the Wilcoxon (1945) test is that it offers the capacity to examine the significance of the increase or decrease in results between two variables, in relation to each other. In the context of the results reporting here, Wilcoxon (1945) shows a change in median from N = 6, to N = 2, which conforms a shift in response. These responses are relative to the scale which was used to explore learning outcomes (1 = Strongly agree, 7 = Strongly disagree), thus a shift in median to N =2, shows a change in a positive direction. To conclude, Wilcoxon calculates the statistical significance of the increase or the decrease in median to arrive at a statistic which determines the statistical significance of the change between two related or matched pairs of responses, which in this case shows a positive change, demonstrating an increase in teacher confidence and ability.

Table 36 Wilcoxon (1945) Signed Rank Test First Pre Learning Outcome Results						
Variable	Response	Mean	Likert Scale Minimum	Likert Scale Maximum	Median Result	P Value
Pre-workshop Results						
(1) Planning Bridge21 Activities	152	5.13	1	7	6	p<.005
(2) Programming Ability	153	5.13	1	7	6	p<.005
(3) Programming Concepts	153	5.38	1	7	6	p<.005
(4) Programming Processes	153	5.23	1	7	6	p<.005
(5) Content Knowledge	152	5.16	1	7	6	p<.005
(6) Teaching Programming using Bridge21	152	5.47	1	7	6	p<.005

Table 37 Wilcoxon (1945) Signed Rank Test Last Post Learning Outcome Results						
Variable	Response	Mean	Likert Scale Minimum	Likert Scale Maximum	Median Result	P Value
Post-Workshop						
(1) Planning Bridge21 Activities.	153	2.64	1	7	2	p<.005
(2) Programming Ability	153	2.71	1	7	2	p<.005
(3) Programming Concepts.	153	2.52	1	7	2	p<.005
(4) Programming Processes	151	2.57	1	7	2	p<.005
(5) Content Knowledge.	153	2.45	1	7	2	p<.005
(6) Teaching Programming using Bridge21	153	2.81	1	7	2	p<.005

6.3.6 Phase 5 - Qualitative Analysis

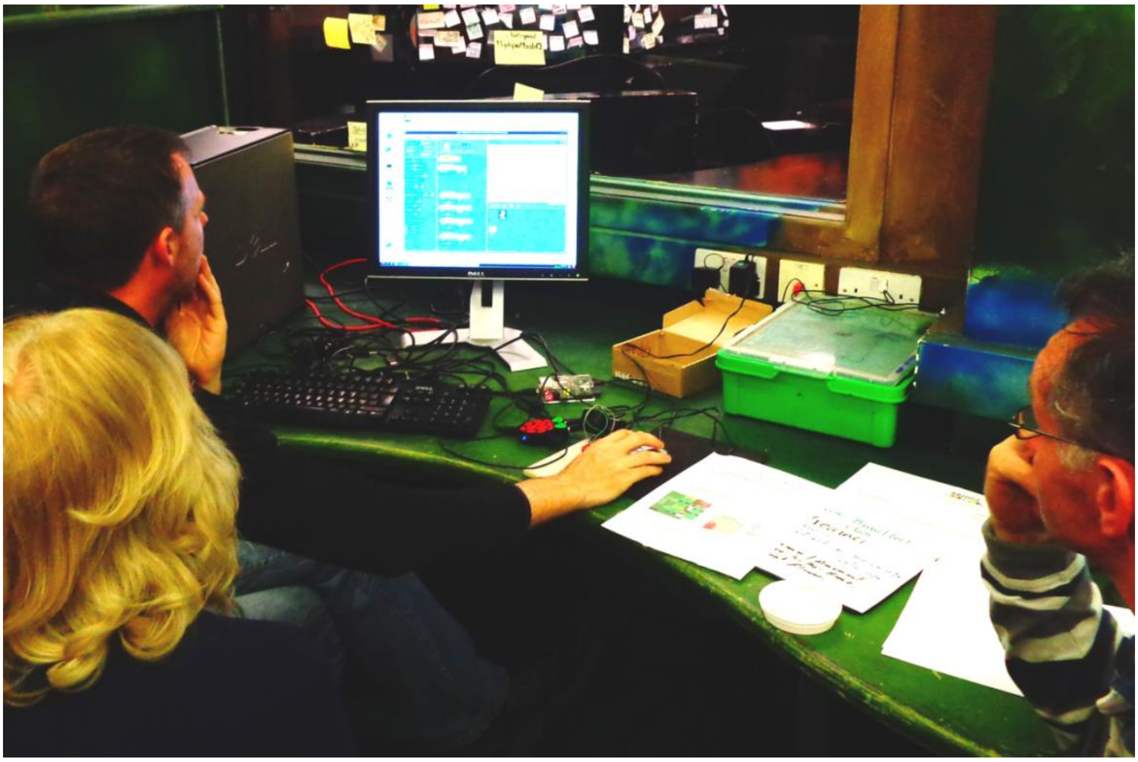
The previous section explored the quantitative leaning outcome data, using descriptive and significance testing. This section provides corresponding qualitative analysis, which explores content knowledge and methods developed by teachers. Qualitative responses were obtained from a self-selecting sample of *N* = 607 teachers who responded to the question: "*how might this Bridge21 computing workshop help you in your classroom teaching?* This question was adapted from an existing questionnaire (Kristiansen, 2007) and two themes emerged from coding the responses (Appendix 9.13.5). The first theme explored (1) programming content knowledge developed by teachers and the second theme examined (2) teacher perceptions of using the Bridge21 model.

(1) Theme 1 – Programming Content Knowledge

Teachers reported developing a better understanding of computing after completing the Bridge21 CPD workshops. For example one teacher reported that they had obtained a *"better understanding of python and similarities to scratch"* (CSCPD_3), while a further teacher reflected that they had enjoyed working through programming activities which had provided "*a path between languages"* (CSCPD_453). Learning programming through completing project with peers helped teachers make connections between concepts (Figure 6-16, p. 174). An example of this can see in the following response, where one teacher reports that they had developed a better understanding of programming and methods used to teach the content: I have a *"better grasp of Python (and a deeper) understanding of how it relates to my teaching subject"* (CSCPD_62). Working on programming projects also encouraged teachers to engage with physical computing, with one teacher reflecting that they had developed a *"basic understanding of Raspberry Pi inputs / outputs use"* (CSCPD_50). Peers played an important role in helping teachers' master basic computing content knowledge. For a one teacher, the opportunity to work on computing projects with peers had helped them to *"really consolidate an understanding of blocks, sequencing, and commands"* (CSCPD_457). The experience of project-based learning helped one teacher achieve *"a greater understanding of gaming and how to make a sample game"* (CSCPD_5), with another sharing *"that it's not just about learning how to use Scratch[5]; it's about the problem solving involved"* (CSCPD_570).

[5] *Block based visual programming language*

Figure 6-16 Teachers Programming in Scratch using the Raspberry Pi (Byrne, Fisher, & Tangney, 2015b)



Teachers enjoyed learning programming through using the Scratch programming language, with teachers reporting that they had learned a *“new function on scratch - "if" and "when" adding sounds”* (CSCPD_405) and *“how to use lots of aspects of Scratch, how to move sprites, broadcast messages”* (CSCPD_255). The experience of creating games and animations using the Scratch programming enabled teachers to explore functions, variables and other programming concepts, with one teacher reported that they had learned *“aspects of writing code - how to make the code sensitive / interactive (i.e. a game rather than an animation)”* (CSCPD_26). The CPD empowered one teacher with the confidence to try and simulate similar exercises in the classroom, e.g. *“I feel confident that I can create a game in scratch and help students do it too”* (CSCPD_290). The Scratch programming language exposed teachers to complex computing concepts, with one teacher reflecting that through using Scratch that they had *“learned more about global variable, events, concurrent bits of code”* (CSCPD_353). The experience of ‘learning by doing’ resonated with teachers with teachers reporting that they enjoyed the practical process of writing programs. One teacher reflected, “*I found it fascinating and want to learn more - a pleasant surprise!*” (CSCPD_164), with a further teacher reporting that through attending the CPD workshops they had learned that the “*end goal is the learning motivator (fun task leads to learning through doing)!”* (CSCPD_152).

(2) Theme 2 – Teacher Perceptions of using the Bridge21

Teachers planned on using the Bridge21 model for teaching programming. In the following example, one teacher reported that they had enjoyed the Bridge21 CPD workshops and that the *"Bridge21 model was very good"* (CSCPD_128). A further teacher reported that they planned to *"use the Bridge21 model for group projects in class"* (CSCPD_611), with one teacher planning to use the *"structure of Bridge21 as it suits classroom work on Scratch"* (CSCPD_525). The Bridge21 CPD workshops had also provided teachers with access to resources, and activities, which they could replicate in a classroom as well as providing a methodology for teaching the content. One teacher reported that the workshop experience had given them *"some good resources I can use, Bridge21 model"* (CSCPD_519). A further teacher reported that they planned to *"bring back Scratch skills to students and show them what I know and maybe learn more from them"* (CSCPD_129). Teachers had embraced the Bridge21 model and were planning learning experiences, which incorporated collaborative and project-based learning. One teacher reflected that the workshops had changed their perception of teaching and reported that *"student led learning is the way forward!"* (CSCPD_523). A further teacher planned on using the Bridge21 model to engage students through using programming tasks designed to *"motivate, stimulate and brighten up students learning"* (CSCPD_430). One teacher planned on using the Bridge21 model as a way to make *"lessons more creative, fun"* (CSCPD_391), while another teacher reported that they intended to use the model to give their students *"more freedom but with scaffolding outline"* (CSCPD_85).

The following examples provide further insight into the ways that teachers planned on using the Bridge21 model to teach computing, with teachers also considering how the model might be used more generally, to teach any subject. One teacher reflected that the rational for integrating the Bridge21 model into their teaching was that *"students need to rely more on themselves than teacher to achieve aims"* (CSCPD_531) while a further teacher intended to use the Bridge21 to *"allow for more peer learning and avoiding the teacher being used as a giver of information"* (CSCPD_17). Teachers also planned on using the Bridge21 pedagogy and process as an integrated approach which could be used to *"complement the things I already do in my teaching"* (CSCPD_563), and could also be used to *"try out more teamwork and self-directed teaching"* (CSCPD_576).

6.3.7 Discussion

The learning outcome findings were analysed to address research question Q1.2 what content knowledge did teachers learn? These findings captured feedback from teachers self-reporting on their experience of the CPD content and methods. A key finding is the reported positive change in teacher learning outcomes post-workshop completion. These findings resonate with the literature exploring collaborative learning theory as a method for teaching and learning computing. Luckin and Du Boulay (1999) argue that collaboration plays an important role in helping learners understand computing. Teague and Roe (2008) suggest that the reason that collaboration is encouraged in computing lessons is to help learners develop the confidence to engage with computing concepts, which learners may perceive as complex and difficult to learn. Tom (2015) continues that learners can experience anxiety when learning computing, and suggests using collaborative learning experiences as a way to make computing interactive, engaging, and fun. Fundamentally, Vygotsky (1978) suggests that learning collaboratively, and interacting with peers, plays a vital role in helping learners address initial knowledge gaps, with peers providing assistance and guidance to address questions, provide insights into methods and advise on practical strategies. The findings explored in this section demonstrate the strength of a collaborative approach to learning computing, which Teague and Roe (2008) argue is essential in developing a practical understanding of computing.

6.4 Strategies teachers intend using for teaching computing

The third level in the Kirkpatrick (1994) model explores ‘behaviour’. The content of the behavioural questionnaire was adapted from an existing instrument (Kristiansen, 2007) and was administered at the end of each CPD workshop to explore teacher intentions to use Bridge21 model and the workshop content. The data from this questionnaire was collected to answer the research question: Q1.3 what strategies did teachers intend using for teaching computing?

6.4.1 Data Set Limitations

Before exploring the behavioural findings the limitations or the data set are discussed. This data set contained one scale and one question. The researcher acknowledges that this is a small data set but it provides useful information, which captures teachers’ intent to use the content of the workshops including elements of the Bridge21 model. A further limitation is the later inclusion of this question into the data set which means that the number of responses is lower for this questionnaire than from that of other post-workshop CPD instruments. Thus, the results are used as indicators, and a Cronbach Alpha score was not generated. The following section provides analysis, which addresses teachers’ intentions to use the CPD, and includes examples showing strategies to engage students.

6.4.2 Quantitative Analysis – All Years

The post-workshop results in Table 38, arranged 1 = Strongly agree to 7 = Strongly disagree, confirm that teachers' intended using Bridge21 model in their teaching (*m* =1.83).

Table 38 Behavioural Intention Post-Workshop Variable All Years	
Cronbach's alpha = N/A for 1 scaled item	
7 Point Likert Scale [1 = Strongly agree; 7 = Strongly disagree]	
Measure of Agreement	
Variables	Intention to use Bridge21
Response	415
Mean	1.83

(1) Variable - Intention to Use Bridge21 – All Years

The intention results show (Table 39) that 91% of teachers intended using the Bridge21 model, with 4 % neutral in their view and the remaining 5 % of teachers' not intending to use the model.

Table 39 Intention to use Bridge21 All Years		
Question - I intend to use the Bridge21 model in my teaching computing on return to the classroom		
7 Point Likert Scale [1 = Strongly agree; 7 = Strongly disagree]	Sum of all Post-workshop Responses	
Measure of Agreement	Response	Valid Percent
1. Strongly agree	241	58 %
2. Agree	91	22 %
3. Somewhat agree	44	11 %
4. Neither agree nor disagree	16	4 %
5. Somewhat disagree	7	2 %
6. Disagree	6	1 %
7. Strongly disagree	10	2 %
Valid Response	415	100 %

6.4.3 Quantitative Analysis – Year on Year

The previous section provided an aggregated view of the 'intention' findings. Further analysis exploring the same data set, year on year (Table 40) provides the capacity to explore change over time. The following section explores the intention findings, covering three consecutive years.

Table 40 Post Workshop Intentions – Year on Year Results

Likert Scale Question: I intend to use the Bridge21 model in my teaching computing on return to the classroom

7 Point Likert Scale [1 = Strongly agree; 7 = Strongly disagree]

Measure of Agreement	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	-	-	-	-	84	58	109	58	48	58
2. Agree	-	-	-	-	25	17	39	21	27	33
3. Somewhat agree	-	-	-	-	11	8	27	14	6	7
4. Neither agree nor disagree	-	-	-	-	9	6	6	3	1	1
5. Somewhat disagree	-	-	-	-	4	3	2	1	1	1
6. Disagree	-	-	-	-	4	3	2	1	-	-
7. Strongly disagree	-	-	-	-	7	5	3	2	-	-
Valid Response	-	-	-	-	144	100	188	100	83	100

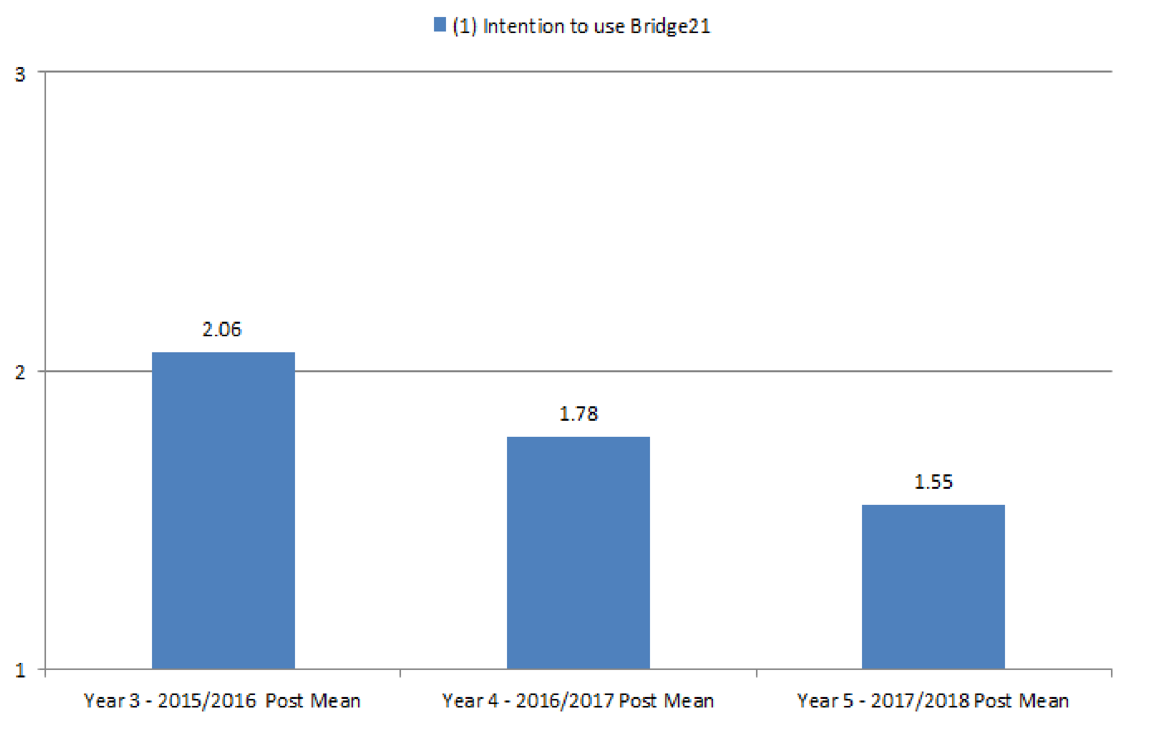
(1) Year 3, 4 and 5 Intention Variables Analysis

The intention questionnaire was phased into the workshop data collection model in the third year of CPD data collection. The post-workshop intention data set was collected to correspond with the collection of the learning outcome and reaction data at the end of computing workshops. The post-workshop intention data set includes responses from the last three cohorts of teachers attending the one year, part time Post Graduate Certificate in 21st Century Teaching and Learning programme.

Quantitative analysis (Figure 6-17) based on Likert scale results reported in Table 40, p. 178 found that in the 2015/2016 data set, that 83% of teachers agreed that they intended to use the Bridge21 model in their teaching computing on return to the classroom (m = 2.06), with 6% neutral in their view, and 11% not intending to use the Bridge21 model. More teachers planned on using the Bridge21 model in the 2016/2017 data set, with 93% of teachers planning on using the Bridge21 model, with 3% neutral in their view, and 4% not planning on using the Bridge21 model. Finally, analysis of the 2017/2018 data set confirmed that 98% of teachers intended using the Bridge21 model in their teaching on return to the classroom, with 1% neutral in their view, and 1% disagreeing that they intended to use the Bridge21 model. These results show a strengthening of teacher intention to use the Bridge21 model over the last three years of the CPD workshop analysis.

Figure 6-17 Intentions Post-Workshop Variables Year 3, 4 and 5

7 Point Likert Scale Results Clustered 1 = Strongly agree; 2 = Agree; 3 = Somewhat agree



6.4.4 Qualitative Analysis

This section provides the qualitative analysis, which adds further context to the quantitative result. The qualitative responses were obtained from a self-selecting sample of N = 646 teachers who responded to the question: "*how might this Bridge21 workshop help you in your classroom teaching of computing?*" This question was adapted from an existing questionnaire (Kristiansen, 2007) and one theme emerged from coding the responses (Appendix 9.13.5). This theme captures examples of (1) activities teachers planned on using in the classroom, with further analysis of field notes exploring (2) strategies for increasing student engagement (Appendix 9.13.6).

(1) Theme 1 – CPD Activities for Teaching CS

Teachers planned on using activities that they had experienced in the CPD workshops. For example one teacher planned on using the content of the Problem Solving / Computational Thinking workshop as a way to *"introduce programming holistically*" (CSCPD_15). Another teacher planned on using elements of the Raspberry Pi workshop content as a method to enable them to "*create a module for Transition Years and 1st Years on programming*" (CSCPD_163). One teacher reflected that the Scratch workshops (game design) had provided a "*good insight into how to create a group work activity and how you can get students to be more independent learners. Scratch could be used in I.T. classes or setting to up a coding club as an extra-curricular activity. (The workshop content) would also work well with a TY (Transition Year) class*" (CSCPD_9). Further examples of teachers intending to reusing CPD activities include the use of the "*Scratch to python -> scratch sheet as intro to python*" (CSCPD_600), (an activity in the python workshop). Another application that teachers planned to use with students was the Blockly Maze game (https://blockly-games.appspot.com/), with one teacher confirming that they "*will certainly use Blockly in the classroom. I will work through problem solving activities with my class focusing on how to solve the problem, different ways to solve the problem. Focusing on the process. This workshop has made me think more about teaching children problem solving skills and how this can be applied to various areas of the primary school curriculum"* (CSCPD_56).

Teachers also planned to use elements of the Bridge21 model as a method for teaching computing. Teachers perceived that a facilitator led approach to teaching as complementing student-centred learning, with teachers planning to use facilitation in combination with collaborative learning tasks. For example, one teacher reflected that the workshops had helped them develop the "*confidence as a facilitator for my students"* (CSCPD_131). A further teacher planned to "*act more as a facilitator in class"* (CSCPD _16), with another teacher reflecting that they felt prepared to "*actually facilitate a class so the kids can design a game through scratch*" (CSCPD_276).

Teachers observed that using facilitation required a change in teaching practice. One teacher observed that they needed to use "*less instruction"* in order to support *"more facilitated learning*" (CSCPD_405). A further teacher observed that the workshops has helped them to explore using a different method for teaching computing which was "*mainly about facilitating – no lecturing"* (CSCPD_415) with a further teacher planning to "*facilitate rather than teach – use students who have more experience as mentors*" (CSCPD_131). These examples demonstrate that teachers were motivated to use facilitation and that teachers saw the potential of using this strategy. However, teachers did not plan on using facilitation in isolation. Teachers intended using facilitation as part of a Bridge21 approach, with one teacher reflecting that using the Bridge21 model would; *"allow me to create lesson plans and facilitate young people using Scratch"* (CSCPD_87). In one instance, a teacher reported that they intended to use the Bridge21 model with facilitation to *"approach the planning of my lessons to maximise peer based, teacher facilitated opportunities for problem solving using computational thinking"* (CSCPD_36). While a further teacher reflected on the potential of using facilitation to make the teaching "*more interesting for students (and to) make it more relevant for students (where) teaching becomes facilitating"* (CSCPD_418). Figure 6-18 shows teachers working together to program in Scratch, generating commends to operate a Makey-Makey controller.

Figure 6-18 Teachers Operating Scratch using Makey-Makey Controllers

Finally, teachers were positive in their attitude towards using the CPD content, with one teacher reported that they were ready to help their students "*create programs"*

(CSCPD_269). A further teacher reflected that they were motivated to start using their technical skills with their students, reporting that it was "g*reat to bring tech skills to class as they are so necessary and ubiquitous in everyday / work / life now"* (CSCPD_242). One teacher was ready to "*inspire students to create Scratch games"* (CSCPD_363), while a further teacher planned on using puzzles and worksheets that they had used in the Problem Solving / Computational Thinking workshop so that *"I can give the students a task on what an algorithm is"* (CSCPD_278). A further teacher reported that they were prepared to start writing their own computer programs and use these examples, stating that they were confident in preparing *"simple programs and allow students to modify them"* (CSCPD_498).

(2) Theme 2 – Intended strategies for Increasing Student Engagement

Further analysis of field notes (Appendix 9.13.6) shows that teachers planned on using the Bridge21 model as a pedagogical process, to engage students in computing, with one teacher reporting that the process: "*will help me make classes I feel more interesting for students. I can also show colleagues how the Bridge21 model works*" (CSCPD_294). Teachers also planned on using the process to introduce activities which were designed to encourage students to engage with problem solving, with one teacher stating that they planned to "*introduce more problem solving to get students to think outside the box and more creativity"* (CSCPD_323), with a further teacher planning to use to use activities to *"help students to devise their own problem solving skills"* (CSCPD_333). A further teacher viewed the pedagogical process as providing the capacity to help students strengthen their ability to problem solve: "*I will allow my students more time to problem solve and have some independence. I can also introduce my colleagues to these topics and hope they will look into the area"* (CSCPD_136). Start-up activities and 'ice-breakers' were further components of the Bridge21 pedagogical process which teachers planned on using, with one teacher reflecting that they intended to use "*ice-breakers for motivation. I will group students to work together to solve problem"* (CSCPD_387). One teacher already used the Bridge21 model adding that they planned to use the CPD to disseminate computing expertise with peers: "*I could present this workshop to colleagues with less or limited technical ability"* (CSCPD_3), with a further teacher planning to practice *"what I have learned, together with a colleague, and I may then offer a short computing course to students"* (CSCPD_261).

Teachers also planned on using the Bridge21 model to teach Raspberry Pi and Python content, providing the following examples outlining how they planned to engage their students. One teacher reported that they would "*like to use the Raspberry Pi in a computer / Information Technology class to introduce students to the basics of hardware. We take this for* *granted so often"* (CSCPD_97), with a further teacher sharing that they were "*going to invest in a Raspberry Pi and Makey-Makey and start to develop my skills more and then I plan to hopefully implement the content with my Transition Year students first and then Junior Certificate short courses"* (CSCPD_274). One teacher planned on adapting the CPD content to design STEM learning experiences: "*I would like to introduce hardware aspects and perhaps try to build some of the hardware (e.g. the controls) as part pf the science /engineering project"* (CSCPD_44), while a further teacher planned to adapt Python and raspberry Pi content and activities to provide maker sessions in their school: "*I plan to teach Python, use the Raspberry Pi for wearable computing, knitting and computing , makerspace"* (CSCPD_160).

Finally, further examples of types of computing content which teachers planned to implement with students covered the use of the Scratch programming language. One teacher reported that they were going to start planning to use the Bridge21 model as a structure to introduce Scratch programming in their school: "*I would think about trying out Scratch with students. I'll probably be doing some online research on this area attending this workshop"* (CSCPD_122). Teachers were energised to teach Scratch using the workshop approach, with one teacher reporting that they were "*defiantly going to try and introduce scratch to students in my school"* (CSCPD_253), with a further teacher prepared to teach Scratch stating that "*I have two students interested in scratch so will be applying it in class from next week"* (CSCPD_471) and a third teacher prepared to use the pedagogical process "*to give s students an opportunity to use Scratch. I would be a facilitator"* (CSCPD_450). Teachers considered the types of Bridge21 activities they would use to engage students, with one teacher reflecting that they planned to use Scratch for "*teaching Irish - I will try to get students to write and perform a story using Scratch"* (CSCPD_13).

6.4.5 Discussion

The behavioural intention findings were analysed to address research question Q1.3 what strategies did teachers intend using for teaching computing? A key finding is the high level of intention to use the Bridge21 model for teaching computing, with teachers planning to use facilitation as a new teaching strategy.

The quantitative evidence reports confirms the trend of year on year intention to use the Bridge21 model as a method for teaching computing in schools (Figure 6-17, p. 179), with a change in mean from m =2.06 in 2015/2016, to m =1.55 in 2017/2018. The year 5 results report that 98% of teachers attending the computing workshops intended to use the Bridge21 model, with 1 % neutral in their view and only 1% reporting that they did not intend using the Bridge21 model.

Qualitative analysis adds further context, providing insights into the workshop content and methods that teachers planned on integrating into computing lessons on return to the classroom. Teachers first reported that they planned on using CPD activities, with teachers providing examples of programming activities combining different workshop elements. Teachers also provided examples of the types of activities that the planned on using, e.g. teachers referencing Blockly and Scratch as two which they planned on using with students. A further theme, which emerged through qualitative analysis, was teachers' intentions to use facilitation as a strategy for teaching computing. Teachers planned on using the strategy to encourage more student participation in computing lessons. Teachers also reflected that they felt prepared to use facilitation to help students explore concepts.

The analysis of quantitative and qualitative findings confirm that teachers intended using the Bridge21 model and the content of the workshop in teaching computing, as it equipped teachers with the practical expertise and confidence to introduce computing into schools. Lieberman (2001) suggests that using worked examples gives context to computing problems, with learners given the opportunity to explore the parts of a program and analyse how individual parts relate to a program as a whole. Teachers reported that they were confident in their ability to adapt particular workshop examples, with teachers for example, intending to use 'Scratch to Python' work sheets and the Blockly (https://blockly-games.appspot.com/) application as an introduction to visual programming. Computing problems, either in electronic or worksheet form, give learners the opportunity to explore particular elements and analyse how particular elements relate to other aspects of a programme. These findings show that teachers not only intended to use CPD activities and worked examples, some teachers planned on designing and then using their own examples.

Finally, one particular element of the Bridge21 model that teachers planned on using was facilitation. Vygotsky (1978) stresses that the process of peer collaboration, or the process of assisting others in extending and deepening their learning, plays an important role in helping learners address gaps in their own knowledge. Facilitators play a critical role in providing encouragement, as well as play a crucial role in offering support through sharing their content knowledge to help learners. However, a problem with using facilitation as a method of teaching is that teachers may lack the practical expertise in the technique. Indeed, Stronge (2018) argues that a main barrier to using faciltation is that teachers may lack the confidence to support students helping each other to learn. Hamlen et al. (2018) further add that teachers need practical assistance in using facilitation as a technique to encourage students to share their expertise, which can be problematic where students do not have the confidence or the experience to work with their peers. Bridge21 CPD workshops provide a structure which encourages collaboration between teachers, where teachers are given the time to learn together and share content knowledge as well as core practical expertise in computing which they are encouraged to use to help their students overcome confidence barriers. These findings confirm that teachers leaving the CPD were motivated to use the Bridge21 model to introduce facilitation into computing lessons. Teachers also reported that they were prepared to use the Bridge21 model and were confident in their ability to supervise students working collaboratively.

6.4.6 Limitations

The above examples cover computing methods and content that teachers planned on using to teach computing in schools using the Bridge21 model. Further analysis of the *N* = 438 teacher responses to the question: "*how will the CPD help you develop your students' learning further?*" identified issues, covering further support and resources, and further support with collaborative learning, to ensure equal participation (Appendix 9.13.7). One teacher reported that they would need further assistance with planning, after attending a one day workshop: *I will have to further explore some of the elements I encountered today in order to learn about them and figure out how best to apply them it the classroom"* (CSCPD_15), with a further teacher requesting assistance with finding resources: *"I would love to follow the lesson plan outline that was used on us. I just need some hands on training on how to do this with the resources my school has"* (CSCPD_52).

Teachers struggled with the emphasis on collaborative learning. One teacher reflected that they would "*have preferred to have covered more content in the workshop. Do not feel prepared to 'go it alone*" (CSCPD_156), A further teacher reported that they would have preferred a more direct approach: "*I thought today was very difficult and if there wasn't someone in my group who had a good idea of how to code I would have been lost and very frustrated. A more step by step-guided approach would have made me more confident trying this in school"* (CSCPD_71). A further teacher shared that they would have liked to interact with other groups suggesting, "*changing groups around so that everyone mixes and people can learn from each other. Sometimes members of team can overshadow others*" (CSCPD_62). A one day workshop required investment in terms of time, with one teacher reporting that the workshop pacing was two slow: *unfortunately the delivery method meant that it took far longer to write / understand Python code than other coding languages I have used in the past"* (CSCPD_58). Furthermore, teachers were concerned about access to computer rooms on return to schools: "*I will defiantly use the Bridge21 model in Transition Year if I can get access* *to a computer room"* (CSPD_125). These results show that changing methods and learning new content takes time, which is captured by the following teacher who shared that "a*ll these workshops help to influence you and transform your approach to teaching. I have a colleague interested in this course. I will make her aware that while the course is very beneficial"* (CSCPD_164).

6.5 Teaching Computing in Schools

The previous section explored teachers' perceptions of the CPD. However, Kirkpatrick (2007) advises that further analysis should include measures which understand "*how much transfer of knowledge, skills, and attitudes occur*" as a result of the training intervention (p. 6). Thus, in accordance with Level 4 analysis, the last phase of the research explored teachers' actual experiences of using the CPD, and, the results, in terms of teachers' use of the Bridge21 model for teaching computing.

6.5.1 Data Set and Limitations

The data set was generated from the administration of a mixed methods questionnaire exploring teachers' experiences of teaching computing in schools (Appendix 9.12). The content was adapted from an existing Bridge21 scale (Lawlor et al., 2018; Lawlor et al., 2016), barriers to implementing the CPD content scale (Kirkpatrick, 2007), other methods used for teaching CS scale (Sentance & Csizmadia, 2017a) and further CPD covering use of the Bridge21 model scale adapted from Lawlor et al. (2013). The data was collected to answer the research question: Q2.1 what elements of the Bridge21 model did teachers identify as most relevant for teaching computing in practice?

The data set was generated from a self-selecting sample of *N* = 64 teachers who had completed module TA21-Mod-1: Digital Media Literacy and 21st Century Learning, and at least one or more computing workshops over the past five years (Table 41, p. 187). This sample includes *N* = 6 teachers who attended all Bridge21 CPD workshops. Analysis of workshop attendance figures show that teachers attended TA21-Mod-3 (Scratch animation) the most number to times (28%), followed by TA21-Mod-2 (Problem Solving / Computational Thinking), with 25% of teachers reporting attending that workshop. Teachers self-reported attending the most number of workshops in the 2016-2017 academic year (*N* =51), with TA21-Mod-6 (Python) attended the least number of times.

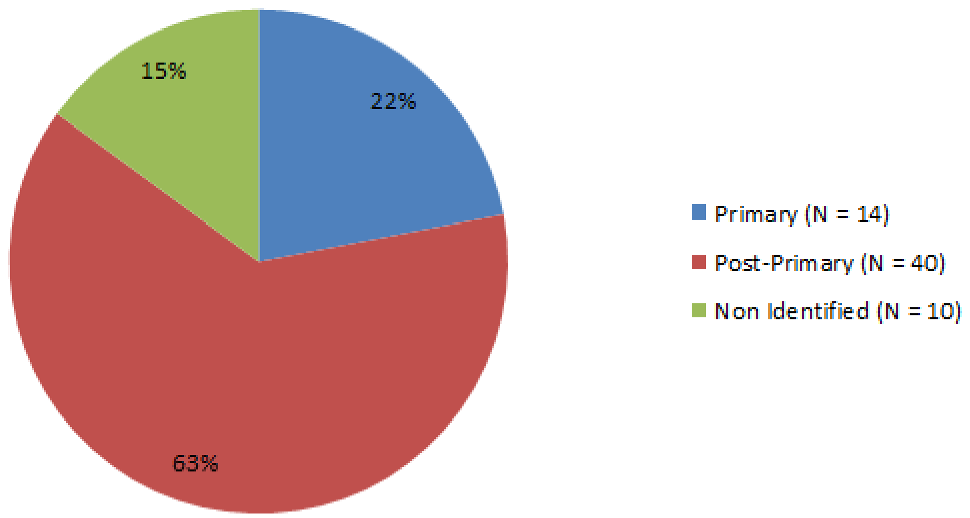
Table 41 Bridge21 CS CPD Computing Workshops Attended by CPD Teachers								
Workshop	2013/ 2014	2014/ 2015	2015/ 2016	2016/ 2017	2017/ 2018	Sub Total	Total	%
TA21-Mod-2	4	7	5	15	4	35	64	25
TA21-Mod-3	5	8	4	16	5	38	64	28
TA21-Mod-4	3	5	2	9	6	25	64	18
TA21-Mod-5	4	5	3	8	4	24	64	17
TA21-Mod-6	2	4	4	3	3	16	64	12
Total	18	29	18	51	22	138	-	100

There are limitations with the data set given that the researcher had limited opportunity to follow up with teachers in schools once they had left the CPD. *N* = 64 teachers responded to the teaching computing in schools questionnaire, and of these not all teachers completed every question. The researcher acknowledges that a small sample is a limitation in supporting generalisations. However, the responses provide a valuable insight into teachers' experiences of using the Bridge21 model and provide a platform for further research. The researcher agrees with Onwuegbuzie and Weinbaum (2017) in that while small samples limit the capacity to generalise, nevertheless the analysis plays an important role in obtaining insights into 'phenomenon, individuals, or events, as is most often the case in interpretivist studies.'

6.5.2 Demographics

Analysis of participant profile data (Appendix 9.14.6.1) found that the sample contained primary and post-primary teachers (Figure 6-19, p. 188). A total of *N* = 14 teachers identified as teaching at primary level with *N* = 40 reporting that they were involved in teaching at post-primary level. The remainder (*N* =10) chose not to self-identity their status. The post-primary sample includes a total of *N* = 9 teachers who self-reported teaching computing at second level.

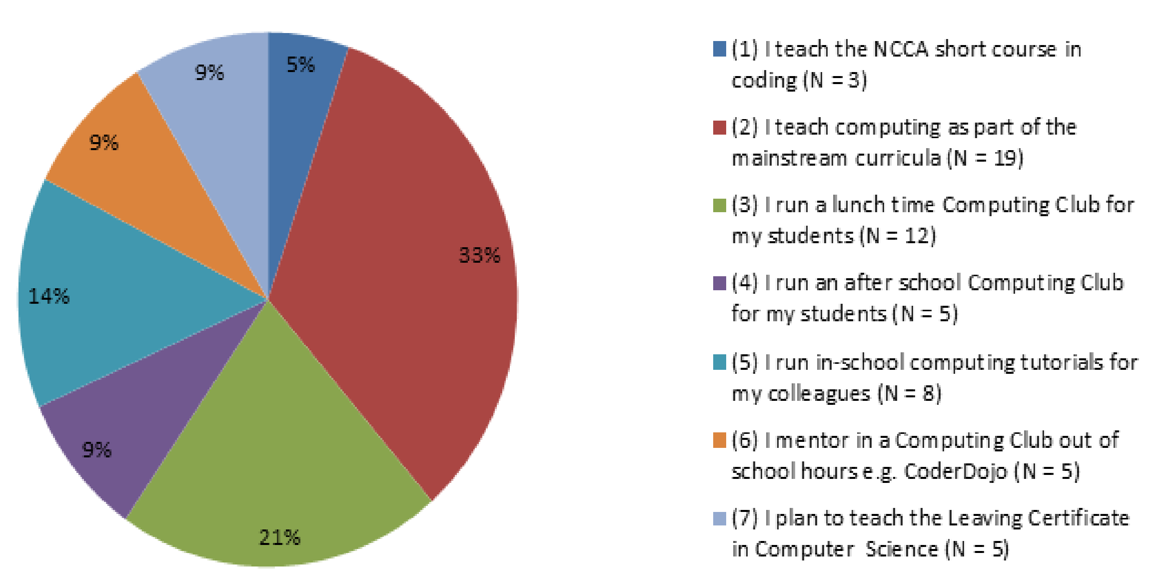
Figure 6-19 Breakdown of Primary and Post-Primary Teacher Sample



Further analysis (Figure 6-20, p. 189) shows that 33% of teachers self-reported that they taught computing as part of the mainstream curricula (*N* = 19), with 21% of teachers running clubs at lunch times (*N* = 12), and 14% of teachers volunteering to run tutorials with peers (*N* = 8). A further 9% of teachers reported that they were active in running after school computing clubs (*N* =5) with 9% also devoting their own time to mentoring in clubs run external to their school (*N* =5). To conclude 9% of teachers reported planning to teach the Computer Science leaving certificate (N = 5). The data supporting this analysis is listed in Appendix 9.14.6.2.

Field note analysis captures some of the reasons why numbers for teaching the NCCA short course in Coding are reported as low. The analysis shows the inclusion of primary school teachers in the sample, which may impact upon the percentage of the sample teaching the NCCA short course. In terms of the post-primary teachers, the reasons put forward for not offering the course include *"my school is not offering short courses yet"* (TCIN_2), "*no timetable allocation yet but it is being discussed next year*" (TCIN_5), computing is "*not part of the school's curriculum*" (TCIN_23), and "*timetable does not allow it*" (TCIN_36). These examples show some of the complexities in integrating new courses, into the curricula.

Figure 6-20 Teaching Computing in Schools Profile



6.5.3 Road Map

In accordance with the Kirkpatrick (1994) model, the remaining sections cover level four which examines the results as they relate to a work place context. This section is organised into four sub-sections. The first section explores elements of the Bridge21 model used for teaching computer science. Section two explores barriers to implementing the Bridge21 model in a computing context and section three examines other methods used for teaching computer science enabling comparison between Bridge21 and other computing teaching methods. Section four explores requirements for further CPD using the Bridge21 model. Further qualitative analysis examines two themes. The first theme explores teacher use of the Bridge21 model to teach computing with teachers providing demonstrating use of the Bridge21 model to enhance student engagement in computer science. Discussion of the results follows with analysis covering limitations with implementing Bridge21 and a summary concludes this chapter.

6.6 Elements of the Bridge21 model used for teaching computing

The following results contained in the subsequent sections were produced through descriptive reporting. Cronbach's (1951) Alpha reliability coefficient values were calculated for Likert scaled items and LeCompte and Schensul's (1999) logical framework was used to govern the process of coding and creating themes generated through pattern coding. The following section explores elements of the Bridge21 model that teachers reported using for teaching computing in schools.

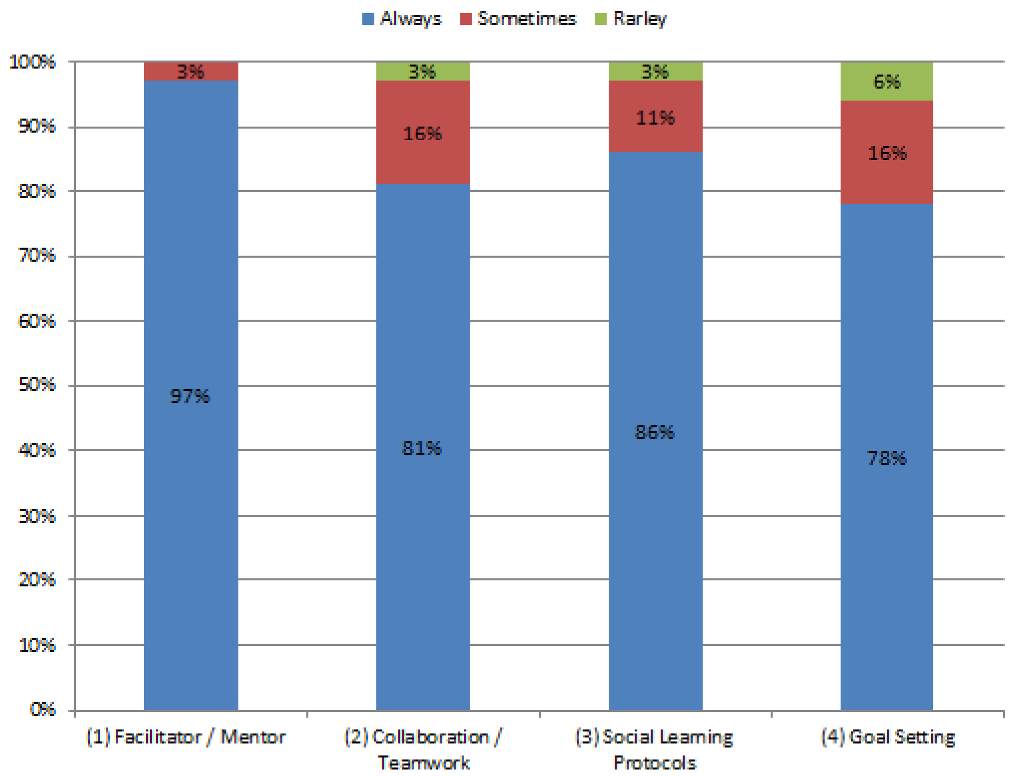
6.6.1 Quantitative Analysis Scale 1 – Bridge21 Elements

Analysis of Bridge21 element variables – Table 42, arranged 1 = Almost never to 5 = Almost always, (to comply with other teaching in school Bridge21 scales), confirm that to a large extent teachers used a facilitator approach to teaching computing (*m* = 4.47) and they incorporated student collaboration through teamwork into lessons (*m* = 4.31). The data indicated that teachers also used social learning protocols (techniques to nurture student self-confidence and develop learner autonomy) (*m* = 4.22), reflection activities (*m* = 4.17), and goal setting (*m* = 4.17) with equal frequency. There was also extensive use of skills focused tasks (*m* = 4.14). Teachers endeavoured to configure classrooms to support students working in teams (*m* = 3.96) and integrated technology-mediated activities into computing lessons (*m* = 3.92). Cronbach alpha for this scale is .834 indicating a strong degree of reliability or consistency with this scale. Appendix 9.14.6.3 provides the Likert scales used in this analysis.

Table 42 Bridge21 elements (Lawlor et al., 2018; Lawlor et al., 2013; Lawlor et al., 2016)				
Here is a list of the elements of the Bridge21 model. For each one, please identify the extent to which you use it for teaching computing?				
Cronbach's alpha = .834 indicating a strong degree of reliability or consistency with this scale				
5 Point Likert Scale [1 = Almost never, 5 = Almost always]				
Measure of Frequency				
Variables 1	(1) Teacher as Facilitator / Mentor	(2) Collaboration / Teamwork	(3) Social Learning Protocols	(4) Goal Setting
Post CPD Response	36	36	36	36
Mean	4.47	4.31	4.22	4.17
Variables 2	(5) Reflection	(6) Skills Focused Tasks	(7) Consideration of the Learning Space Design	(8) Technology-Mediated Learning
Post CPD Response	35	36	36	36
Mean	4.17	4.14	3.96	3.92

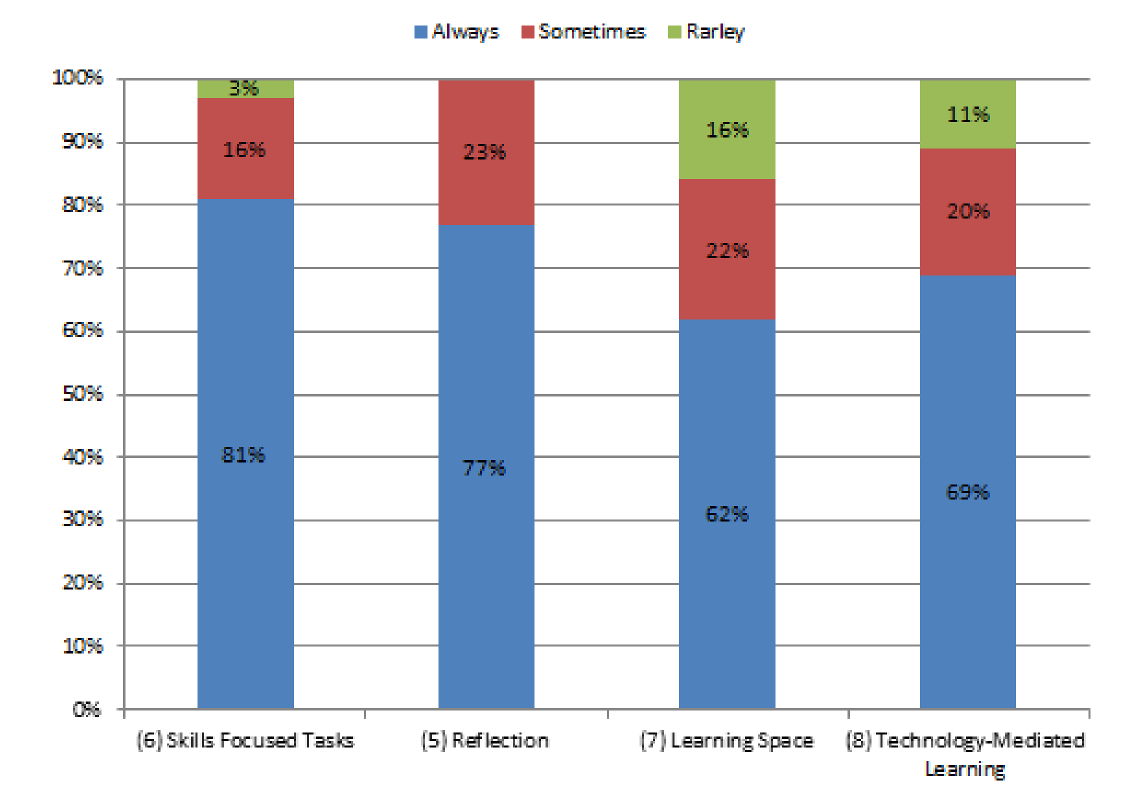
Analysis of the first four Bridge21 element variables (Figure 6-21) show that 97% of teachers used facilitation to teach computing, with the remaining 3% indicating that they only 'sometimes' used the approach in a classroom context. Next, 81% of teachers reported using collaborative learning, including teamwork for teaching computing, with 16% sometimes using collaboration, and 3% reporting that they rarely used the method. A further 86% of teachers often or almost always use social learning protocols, and that just over one-tenth (11%) sometimes using protocols and the remaining 3% rarely using the approach. Social learning protocols are techniques that teachers use to encourage confident, learner autonomy and strong personal growth and personal development. Finally, a total of 78% of teachers reported they often or almost always encouraged goal setting, with 16% of teachers reporting that they sometimes used goal setting, and the remaining 6% rarely using goal-setting tasks in a CS context.

Figure 6-21 Analysis of the First Four Bridge21 Elements used for Teaching CS



Analysis of the remaining four Bridge21 element variables (Figure 6-22, p. 192) confirm that 81% of teachers implemented skills focused tasks, with 16% sometimes incorporating tasks and 3% stating they almost never used the approach and that 77% of teachers often or almost always used reflection exercises, with 23% of teachers indicating that they sometimes used reflection. Furthermore, 62% of teachers reported changing the structure of their classrooms to support group working, with 22% sometimes changing the physical arrangement of their classrooms and the remaining 16%, rarely or almost never rearranging the infrastructure of their classrooms, or moving desks to support group work. Those who reported rarely changing the layout of their classroom may have included teachers who already have classrooms set up in a group structure (for example in primary schools) or who already have classrooms configured to support group work in their school. Finally, 69% of teachers almost always, or often, used technology in computing lessons, with 20% sometimes using technology and 11% rarely or "almost never" using technology. These results report that a lack of use of technology, merits further investigation. One factor may be that teachers are using CS unplugged (https://csunplugged.org/en/) and similar approaches which do not rely on technology. CS unplugged provides access to resources, which teachers can use to support problem solving and computational exercises, which do require access to a computer.

Figure 6-22 Analysis of the Second Four Bridge21 Elements used for Teaching CS



Qualitative analysis exploring teacher responses to 'other' uses of the Bridge21 model captured the two examples concerned with the physical layout of classrooms. In the first example, one teacher reported that there are restrictions in organising rooms to support collaborative computing: "*I only teach first years coding at present. On top of this the room we are in is not ideal, and we are awaiting a new school, where I would prefer a pod based work system*" (TCIN_2). In the second example, one teacher reporting that the set-up of a computer room makes group work difficult to implement and that class size and behaviour are also factors: *"the learning space design is difficult in most Computer rooms due to layout of tables and wiring etc. All the above (Bridge21) strategies are dependent on the class group in terms of ability, behaviour, and class size"* (TCIN_13). Having explored which elements of the Bridge21 model teachers use in teaching computing, the next section examines particular barriers to using the Bridge21 model.

In summary, analysis of the Bridge21 element variables and supporting qualitative texts confirm that teachers were most confident in using a facilitator/mentor approach to teaching computing (97%), with teachers also confident in their use of collaborative learning and team working in a computing context (81%). Furthermore, 86% of teachers also reported designing computing lessons to encourage students to communicate, share ideas and support fellow students (social learning protocols), which teachers providing lessons which an emphasis on learning computing skills (81%), and, goal setting (78%), with students creating plans and taking control of the design process. Teachers used reflections (77%), technology-mediated learning (69%), and configuration of the learning space design, with evidence provided in the qualitative results, which confirm difficulties in gaining access to computer rooms and timetable restrictions impeding on classroom delivery.

6.6.2 Quantitative Analysis Scale 2 - Barriers to Using the Bridge21 Model

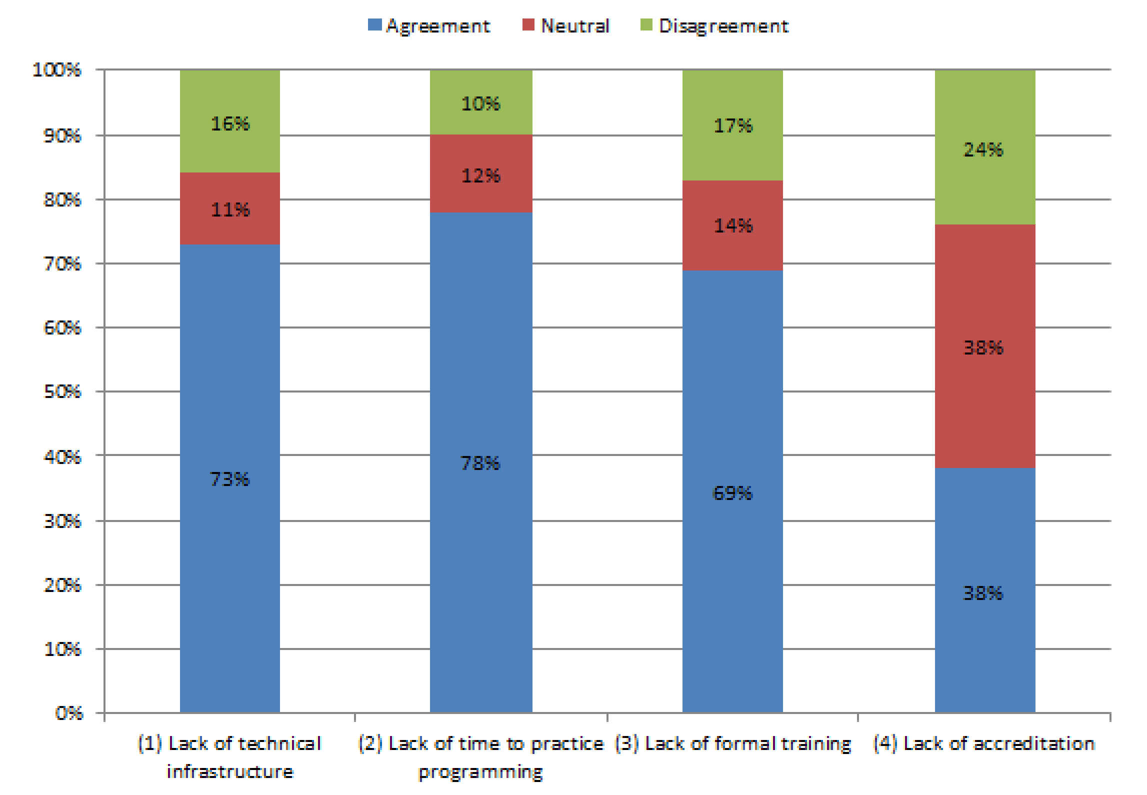
Understanding barriers to using new methods of teaching and learning enable CPD designers to include activities which can help equip teachers with strategies to overcome barriers experienced in school contexts (Kennedy, 2011). There are also technical barriers which can make teaching computing problematic (Sentance & Csizmadia, 2017a). Thus, it is important to establish what barriers teachers perceive as inhibiting use of the Bridge21 model as a method for teaching CS.

The analysis of post CPD barrier variables in Table 43 p. 194 (arranged 1 = Strongly disagree; 5 = Strongly agree) confirm that the lack of a suitable technical infrastructure (m = 3.89) and a lack of time to developing programming skills (m = 3.89) are the main barriers to using Bridge21. Teachers next perceived lack of access to formal training in CS (m = 3.72), and lack of access to accreditation in CS (m = 3.24), as well as timetable restrictions (m = 3.22), as further barriers to using the Bridge21 model. Teachers were less concerned that they lacked peer support (m = 3.14) or lacked the time to implement the Bridge21 model (m = 2.61), with teachers citing practical incompatibility with the timetable (m = 2.38) as well as a lack of confidence (m = 2.35) as barriers to using the Bridge21 model. The analysis also shows that a lack of demand from students (m = 2.27), and a lack of support from senior management (m = 2.84) as well as Bridge21 implementation problems (m = 1.76) were lesser barriers. Each variable is now explored in turn. Cronbach alpha for this scale is .811 indicating a strong degree of reliability or consistency with this scale.

Table 43 Barriers Post CPD Variables (Hogarty, Lang, & Kromrey, 2003; Kirkpatrick, 2007)				
What barriers do you face when considering using the Bridge21 model to teach computing?				
Cronbach's alpha = .811 indicating a strong degree of reliability or consistency with this scale				
5 Point Likert Scale [1 = Strongly disagree to 5 = Strongly agree]				
Measure of Agreement				
Variables	(1) Lack of technical infrastructure	(2) Lack of time to practice programming	(3) Lack of formal training	(4) Lack of accreditation
Post CPD Response	37	36	36	37
Mean	3.89	3.89	3.72	3.24
Variables	(5) Timetable restrictions	(6) Lack of peer support	(7) I haven't found the time to implement Bridge21	(8) It isn't practical for my situation
Post CPD Response	37	37	36	37
Mean	3.22	3.14	2.61	2.38
Variables	(9) Lack of confidence	(10) Lack of demand from students	(11) Lack of support from senior management	(12) I tried it and it didn't work
Post CPD Response	37	37	37	37
Mean	2.35	2.27	2.84	1.76

Analysis of the first four barrier variables (Figure 6-23, p. 195) shows that 73% of teachers perceived that the lack of a suitable technical infrastructure was a barrier to teaching computing using Bridge21 model, with 11% neutral in their view and a further 16% of teachers disagreeing this was a barrier. Furthermore, 78% of teachers agreed that a lack of time to practice programming was a barrier, with 12% neither agreeing nor disagreeing, and a further 10% disagreeing that a lack of time was a barrier. In addition, 69% of teachers agreed that a lack of formal training was a perceived barrier, with 14% of teachers were neutral in their view, with 17% of teachers disagreeing that a lack of formal training in computing was a perceived barrier to using the Bridge21 model. Teachers were also asked about the importance of accreditation, and analysis shows that 38% of teachers agreed that a lack of accreditation was a barrier to further Bridge21 model use. However the same amount of teachers (38%) neither agreed nor disagreed that a lack of appropriate accreditation was a barrier, with just under a quarter (24%) disagreeing that a lack of accreditation was a barrier to Bridge21 model use. Appendix 9.14.6.4 contains the Likert scale tables supporting the analysis.

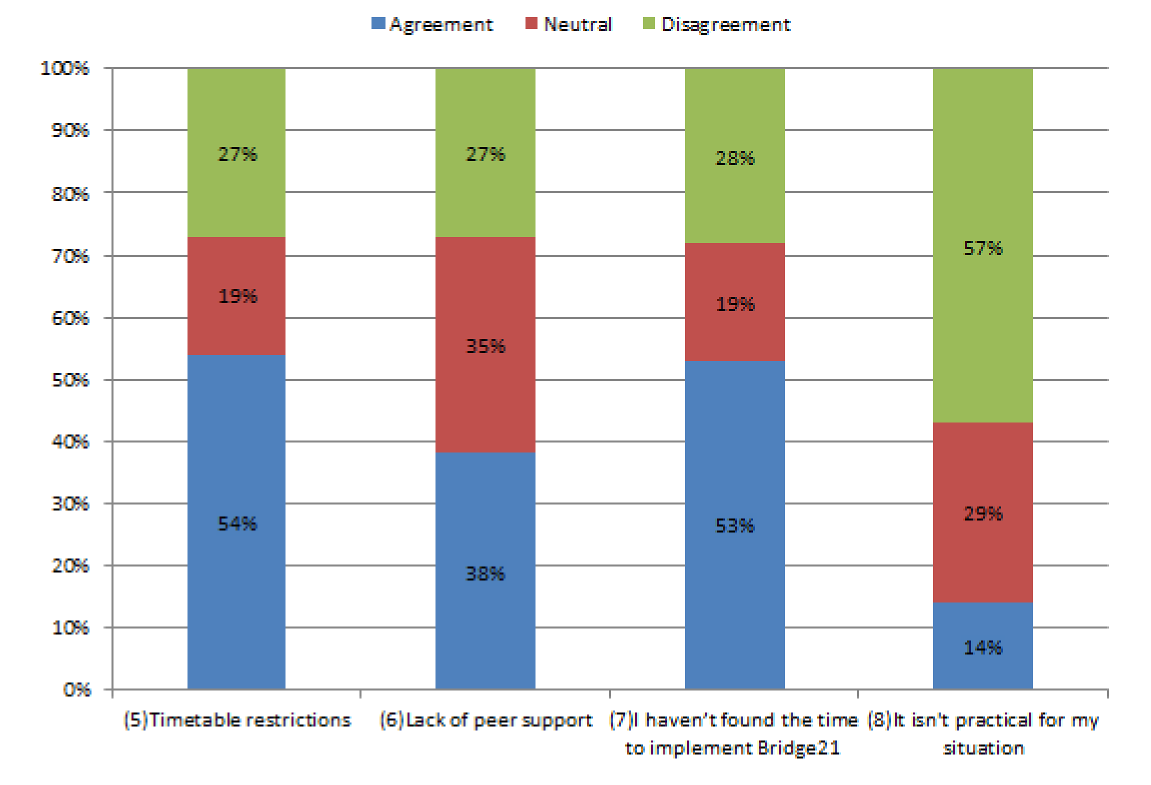
Figure 6-23 Barrier Variable Analysis – First Four Variables



Teachers also reported ‘other’ barriers to using the Bridge21 model. Qualitative analysis of text responses adds context to the quantitative results. A consideration was the difficulty that teaching computing requires a range of methods, with the Bridge21 model providing a methodology, which brings together different approaches, joined together through collaborative learning. As a consequence of this, teachers need assistance in keeping skills up to date, and access to lesson planning to work through different combinations of lessons. This issue is expressed by the following teacher who reported that "*don't think it's a case of one system fits all, teachers need to have a range of methodologies to hand and adapt to the group as they see fit*" (TCIN_47).

Analysis of the second four barrier variables (Figure 6-24) shows that 54% of teachers agreed that timetable restrictions were a perceived as a barrier to using the Bridge21 model, and that 19% presented a neutral view, the remaining 27% of teachers, somewhat disagreeing that the school timetable restricted use of the Bridge21 model. Furthermore, 38% of teachers agreed that a lack of peer support was a barrier, with 35% neither agreeing nor disagreeing, and 27% disagreeing that a lack of peer support was a barrier. Furthermore, 53% of teachers agreed that they had not found the time to implement the Bridge21 model, with 19% somewhat neutral in reporting on their implementation, with the remaining 28% of teachers reporting that finding the time to implement. Finally, 14% of teachers agreed that there were perceived practical barriers to implementing the Bridge21 model in contrast to 57% of teachers disagreed that there were practical barriers to implementing the Bridge21 model in their teaching, with 29% neither agreeing nor disagreeing. Appendix 9.14.3.4 contains the Likert scale tables supporting the analysis.

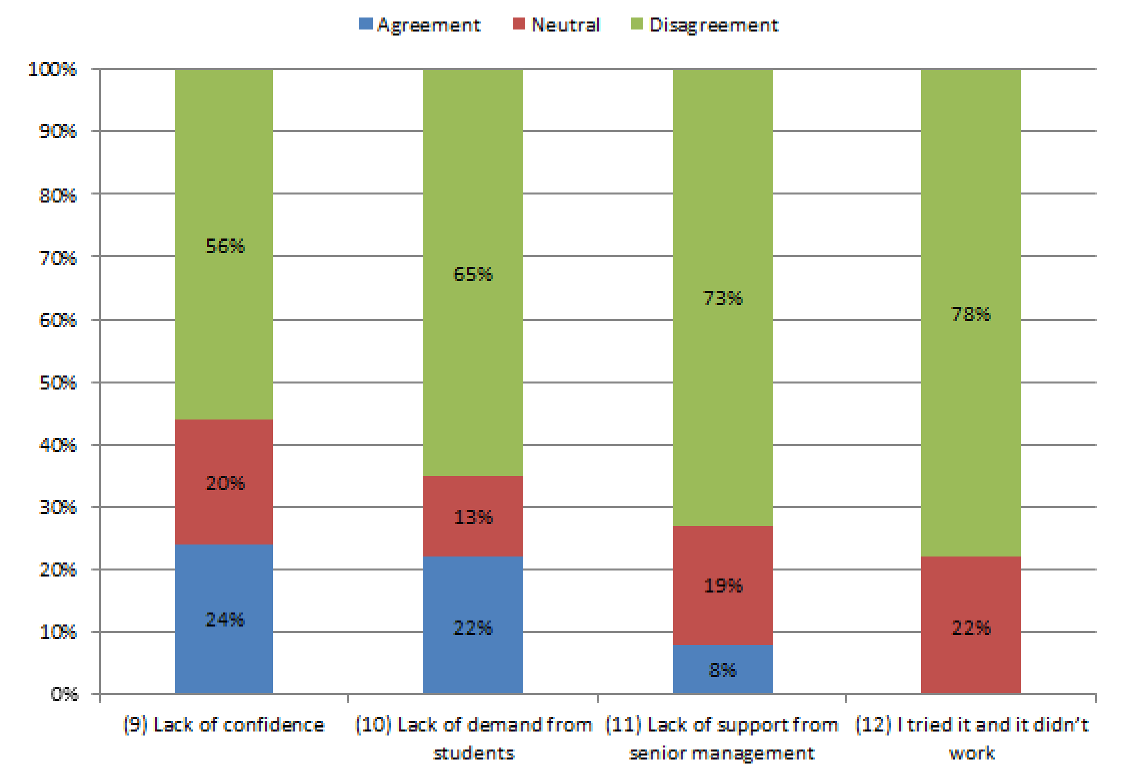
Figure 6-24 Barrier Variable Analysis – Second Four Variables



Complementing the quantitative results are comments from teachers reporting on timetable restrictions and the need for technical support to implement the Bridge21 model in schools. For example, teachers also reported other barriers to using the Bridge21 model. One teacher reported that they experienced the problem losing momentum when using the Bridge21 model: *"It can be time consuming and the gaps between class contact times require the need to continuously remind/repeat instructions. Also, lack of interest by a large cohort of students can make the teaching and learning tough for both students and teacher"* (TCIN_4). Another teacher reported the value of having a technology expertise in the school to help with troubleshooting technical problems: "*the school has a specific IT expert teacher in situ"* (TCIN_28).

Finally, analysis of the four remaining barrier variables (Figure 6-25) confirms that 56% of teachers disagreed that a lack of confidence was a barrier to using the Bridge21 model. Also 65% of teachers disagreed that a lack of student demand influenced their decision to use the Bridge21 model. A further 73% of teachers disagreed that management influenced their decision to use of the Bridge21 model, with 8% reporting a lack of support from senior management was an issue in implementing the Bridge21 model in a computing capacity. Finally, 78% of teachers successfully implemented the Bridge21 model, with 22% of teachers neutral in their view, with no failures reported by teachers who tried the Bridge21 model and it did not work in a computing context.

Figure 6-25 Barrier Variable Analysis – Remaining Four Variables



Complementing the quantitative results are comments from teachers reporting on management issues and constraints with implementing the Bridge21 model. An early career teacher lacked the support to continue using the Bridge21 model in both a classroom and a school context: *"this is my first year in this (*name omitted*) school. Behaviour management is of primary concern among staff. Students fail to work in pairs and quickly become distracted when I attempt to be a mediator, as opposed to leading. Most senior students became dispirited and began playing online games"* (TCIN_50). Finally, organising traditional rooms into a group space was perceived as a problem to fully implementing the Bridge21 model, with on teacher reporting *"I find the lack of space a problem. I would love if there was a collaborative space in my school but there just isn't unfortunately"* (TCIN_64).

In summary, analysis of the first four barrier variables (Figure 6-23, p. 195) confirm that a lack of technical infrastructure (73%) is a perceived barrier to using the Bridge21 model in a computing context, with one teacher raising the need for technical support to support implementations. Furthermore, teachers also identified a lack of time to practice programming as a further barrier to teaching computing (78%), with teachers also reporting on the importance of training (69%) as factor in preparing teachers for teaching computing using the Bridge21 model in a school context. In contrast, 38% of teachers agreed that a lack of accreditation was a barrier to using the Bridge21 model to teach computing with the same number neutral in their view (38%), and 24% disagreeing that accreditation was a barrier.

Analysis of the second four barrier variables (Figure 6-24, p. 196) confirm that time table restrictions in terms of the amount of time given to each class (54%), was a barrier to Bridge21 model use. While 53% of teachers agreed that it was difficult to find the time to implement the Bridge21 model, compared to 28% who confirmed that they had made time to implement the model for teaching computing on a regular basis. A lack of peer support was somewhat perceived as a barrier to using the Bridge21 model, with 38% of teachers agreeing that lack of collegial support was viewed as a barrier to implementing the Bridge21 model. In contrast to these results, 57% of teachers agreed that the Bridge21 model was appropriate as a practical methodology for introducing computing and for teaching the content within their school context.

Finally, analysis of the remaining four barrier variables (Figure 6-25, p. 197) confirm that 56% of teachers disagreed that they lacked the confidence to implement the Bridge21 model in a computing context, with a further 65% of teachers disagreeing that a lack of interest from students was a barrier to teaching computing in their school. Furthermore, teachers disagreed that a lack of support from senior management (for example school principals) was a barrier to using the Bridge21 model (39%), with 78% of teachers confirming that they successfully implemented the Bridge21 model in a computer science context. These initial results confirm teacher confidence in using the Bridge21 model, with teachers' timetable, and peer support as well as well as lack of access to the necessary infrastructure and resources as barriers to using the Bridge21 model to teach computing.

6.6.3 Quantitative Analysis Scale 3 – Other Methods Used for Teaching CS

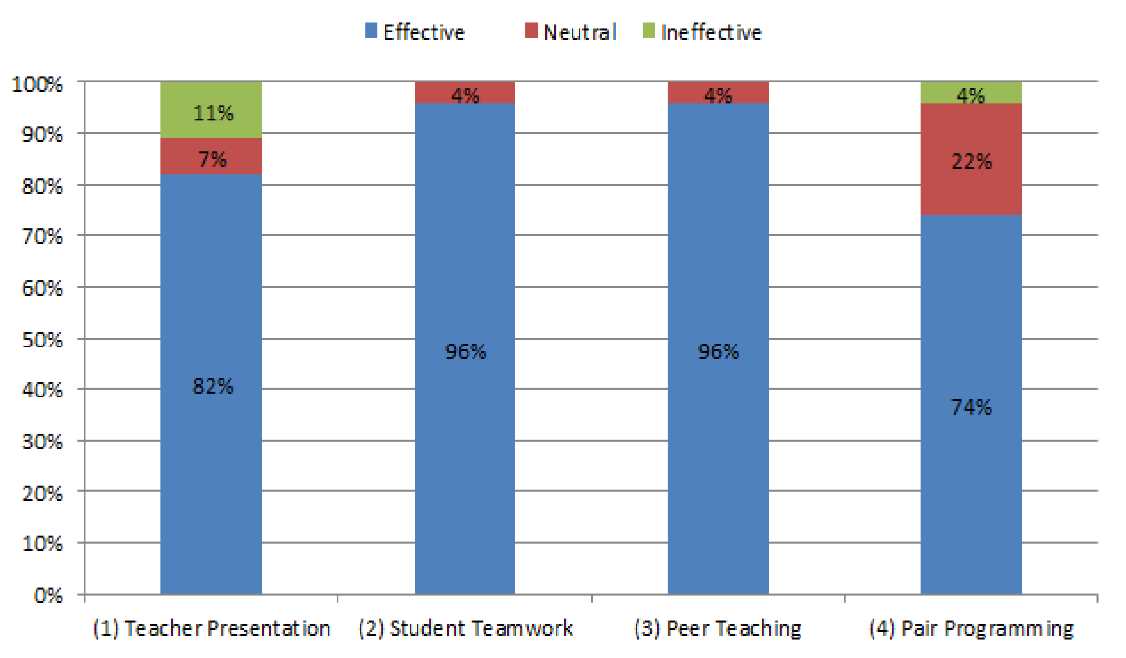
The previous sections explored teacher use of the Bridge21 model and barriers to its use in teaching context. Lister et al. (2007) argues that lectures and self-directed learning play an important role in helping learners grasp core content knowledge, while Sentance and Csizmadia (2017b) suggest using a blend of methods to teach computing. Thus, further research was needed to explore what other teaching methods teachers used in the context of teaching in addition to using the Bridge21 model.

Analysis of other methods used for teaching computer science (Table 44, p. 200), arranged 1 = Very effective to 5 = Very ineffective, confirms that teachers most used peer teaching as a strategy for teaching computer science in the classroom *(m* = 1.36). Teachers next reported using student teamwork /collaborative working (*m* = 146), followed by the use of instructional games (*m* = 1.53), and project work (*m* = 1.53). Teachers next reported using pair programming as a strategy with students (*m* = 2.00), with teachers also using presentations (*m* = 2.03). Teachers reported using online tutorials (*m* = 2.21), and printed worksheets least in their teaching (m = 2.64). Cronbach alpha for this scale is .705 indicating a strong degree of reliability or consistency with this scale.

Analysis of the first four methods variables (Figure 6-26, p. 200) show that 82% of teachers agreed that the use of presentations in teaching computing had a positive impact, with 7% neutral in their view, and the remaining 11% disagreeing that presentations were an effective strategy for teaching computing. Furthermore, 74% of teachers reported pair programming was an effective strategy in the computer science classroom, with 22% neutral in their view, and 4 % starting that the method was 'ineffective' as a teaching strategy. In contrast, 96% of students agreed that teamwork / collaboration had a positive impact and was an effective strategy, with 4% neutral in their view. Similarly, 96% of teachers agreed that peer teaching had a positive impact on student learning, with 4% neutral in their view. Appendix 9.14.6.5 provides the Likert scale tables supporting this analysis.

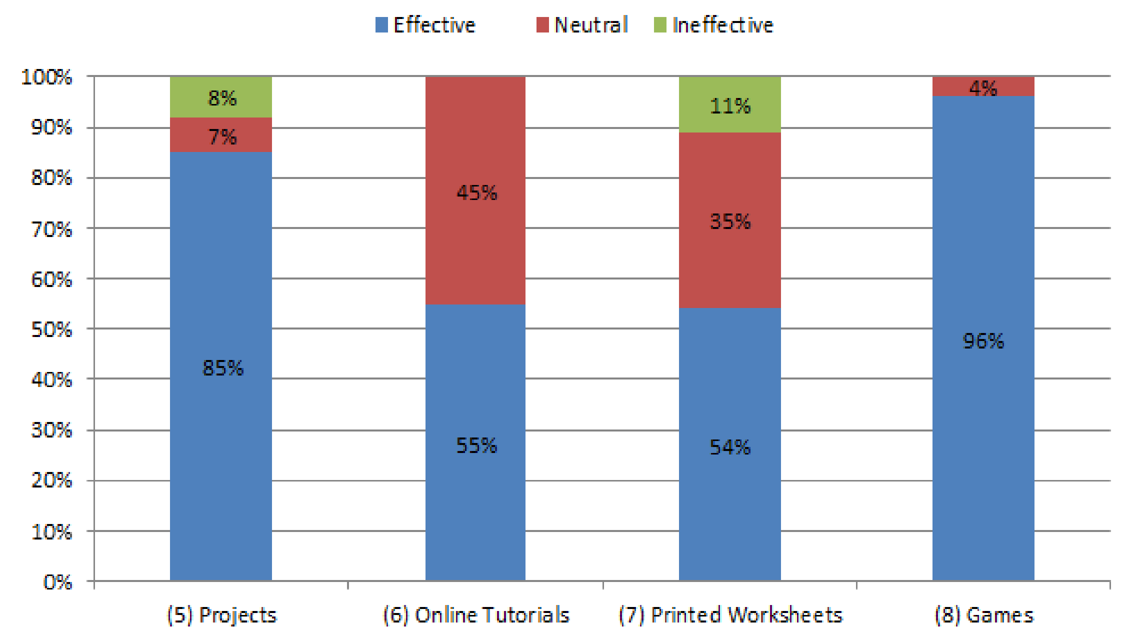
Table 44 Other Methods for Teaching Computing (Sentance & Csizmadia, 2017a)				
Question: Here is a list of strategies used for teaching computing. For each one, please identify the extent to which you use it for teaching computing?				
Sum of all Post-CPD Responses				
Cronbach's alpha = .705 indicating a degree of reliability or consistency with this scale.				
5 Point Likert Scale [1 = Very effective, 5 = Very ineffective]				
Measure of 'Effectiveness/ Impact'				
Variables	(1) Teacher Presentation	(2) Student Teamwork	(3) Peer Teaching	(4) Pair Programming
Response	28	27	28	23
Mean	2.29	1.38	1.24	1.81
Variables	(5) Projects	(6) Online Tutorials	(7) Printed Worksheets	(8) Games
Response	26	22	26	25
Mean	1.76	2.20	2.52	1.52

Figure 6-26 Other Methods Variable Analysis – First Four Variables



Analysis of the second four methods variables (Figure 6-27) show that 85% of teachers agreed that a project-based approach was effective in helping students learn computing, with 7% neutral in their view and a further 8% of teachers disagreeing that a project-based approach was effective as core strategy for teaching computing. In contrast, teachers confirmed that the use of online tutorials (55%) and printed worksheets (54%) were somewhat less effective as a strategy used for teaching computing, with 45% of teachers neutral in their view on the use of online tutorials, and 35% neutral in their view of printed worksheets, with 12% not satisfied with worksheets. Finally, 96% of teachers reported that games were an appropriate approach for teaching computer science, with only 4% neutral in their view. Appendix 9.14.3.5 provides Likert scale tables supporting the analysis.

Figure 6-27 Other Methods Variable Analysis – Remaining Four Variables



Complementing the quantitative results are comments from teachers reporting on the use of other methods of teaching used in the computing classroom. One teacher reported that they “*felt that the effectiveness of the strategies is very group dependent*” (TCIN_17), while a further teacher reported that they would like to develop further expertise in learning to use “*more fun methodologies which would be more beneficial for my younger age group*” (TCIN_37). One teacher shared the example of using pair programming adding: “*a note on pair programming – I have tried this – but I found that the stronger student worked on. Students have to be taught how to teach. Entering Transition Year students into SciFest (see: https://scifest.ie/) - concentrates the mind. Presentation, communication, design. Mentors – I have 3rd level 3rd Year Comp Science students act as mentors with my students when*

Transition Year is being taught python and raspberry pi" (TCIN_41). This last comment demonstrates peer mentoring used to help develop confidence.

In summary, analysis of the other methods for teaching computing variables confirm that teachers most used peer teaching (96%) and student-led teamwork (96%), with teachers also using games (96%), and projects (85%) as core strategies for teaching computing. Least used were online tutorials (55%) and printed worksheets (54%), with teachers also reporting that pair programming (74%), and presentation led teaching methods (82%) were used to a lesser extent in the computing classroom. In comparison, 11% of teachers reported that worksheets were not helpful in teaching computing, with a further 11% of teachers reporting that presentations had limited impact in teaching computing. These results compare with Bridge21, in so far that they capture teachers using projects, games, teamwork and peers teaching as strategies to assist students engage with computer science. Further qualitative analysis of text responses, exploring other methods used to support the quantitative variables shows that the use of a particular strategy is context specific, depending on the group of students involved in a particular lesson. Analysis has also revealed a need for activities for use in primary schools, with primary teachers sharing strategies to support collaboration.

6.6.4 Quantitative Analysis Scale 4 - Further CPD Using Bridge21 Model

A key step in ensuring that CPD programmes meet teacher's needs, is to ask teachers for suggestions which can be included in future CPD offerings (Wood et al., 2017). The researcher adapted questions from an existing instrument (Lawlor et al., 2013) to explore teacher requirements for further CPD in using the Bridge21 model.

Bridge21 CPD variables were adapted from prior research which identified seven areas in teachers indicated they might require further assistance to enhance teaching using the Bridge21 model (Lawlor et al., 2013). The researcher adapted these themes into variables (Table 45, p. 203) and teachers were asked to rank in order the area of the Bridge21 model that most required further CPD assistance. The results are discussed below, with the most requested CPD topic listed first.

Analysis of the Bridge21 CPD variables confirm that teachers most needed CPD in designing learning challenges (*N* = 19 teachers), with (*N* = 18 teachers) teachers requesting more CPD to learn and develop programming skills. Learning challenges are particular activities set by teachers, which involve students completing tasks such as problem solving. Additional CPD was sought covering the planning of ideas for use in computing projects (*N* =16 teachers) with just under a fifth of teachers indicating that they required further assistance with preparing student assessment materials (*N* =14 teachers). Further analysis indicates that teachers were more confident in using group work and facilitation, which are reflected in the low number of responses requesting assistance with integrating group work (*N* =9 teachers) and using facilitation in a computing context (*N* =6 teachers). The area which teachers expressed the least need for CPD was with assigning students roles and then organising students into teams (*N* =3 teachers).

Table 45 Further CPD in using the Bridge21 Model (Lawlor et al., 2013)		
Question: If you are interested in using the Bridge21 model which areas do you feel you require further CPD?		
Select each option that applies to you	Sum of Post CPD Responses	
Measure of Frequency	Response	Valid Percent
(1) Designing learning challenges	19	22 %
(2) Learning programming	18	21 %
(3) Planning computing projects	16	19 %
(4) Preparing student assignments	14	16 %
(5) Integrating group work into computing activities	9	11 %
(6) Using facilitation	6	7 %
(7) Assigning my students roles	3	4 %
Total	85	100 %

Finally, teachers made other CPD suggestions for topics, which were not included in the scale. Appendix 9.13.9 provides the corresponding qualitative analysis supporting this discussion. For example, teachers requested assistance with covering particular programming topics. One teacher called for more 'hands on' time in professional development sessions: "*as regards learning programming, I feel I would need more hands on approaches to JavaScript, as I learned that (1) a few years ago and (2) at a higher level"* (TCIN_12). Other examples include assistance with designing assessments for group work and help with developing additional resources. One teacher reflected that discussion around assessing group work, especially in the context of evaluating individual contributions is a potential area to cover in the computing workshops: "*assessing group work (how to identify the individual's contribution and the overall group effort)"* (TCIN_15). A further teacher reflected that they would like more time exploring the features of the Scratch to provide suggestions and guide students who have completed particular tasks: "*I would appreciate extra options for Scratch. Students become bored with its use. I used PowToon to continue interest*" (TCIN_50).

6.6.5 Qualitative Analysis

This section provides the qualitative analysis, which adds context to the quantitative results. The qualitative responses were obtained from a small self-selecting sample of N =33 teachers who responded to the question: "*what changes in classroom behaviour have you observed through implementing the Bridge21 model?*" This question was adapted from an existing questionnaire (Kristiansen, 2007) and two themes emerged from coding the responses (Appendix 9.13.8). The first theme investigated (1) teacher use of the Bridge21 model to teach computing; the second theme includes reports by teachers who (2) observed an increase in student engagement through teaching computing using the Bridge21 model.

(1) Theme 1 - Using the Bridge21 Model to Teach Computing

Qualitative analysis reveals that teachers are using the Bridge21 model to change from teacher-centric to student-directed learning methods. One teacher reported that they had used the Bridge21 model as a methodology to manage the "*gradual release of responsibility*" (TCIN_19), while a second teacher reported that an outcome of using the Bridge21 model had been "*more inquiry based and student led classes*" (TCIN_24). The Bridge21 model had given teachers a process and a pedagogy to integrate more project-based learning and student-led discussion into a class setting, "*allowing more opportunities for group work and discussion of ideas*" (TCIN_44). One teacher reported that they felt confident in giving students more responsibility for their learning, through "*stepping back to allow students explore what they know and learning by doing. Encouraging them to work out certain solutions on their own and to help and show each other how to do things*" (TCIN_62).

A further teacher agreed that they also felt more confident in facilitating students learning through working in teams, making time in the curricula to give students the opportunity to work together: *"I am sitting back more, and allowing students to investigate the tools. I encourage peers to teach each other. When helping students I deliberately make mistakes at times so students can try and remedy them/prove learning taking place"* (TCIN_54). A further teacher commented that the CPD experience had provided an opportunity to expand their 'repertoire,' reporting that they now used "*collaborative work online, encouraging students to present projects using ICT, more computer based projects, building up my own resource repertoire*" (TCIN_32). These results provide examples of teachers changing their practice to move to a more student-centred, collaborative approach.

(2) Theme 2- Using Bridge21 to Increase Student Engagement

Qualitative analysis also revealed that teachers observed an increase in student engagement, through using the Bridge21 model to teach computing. One teacher reported that "*collaboration evident. More discussions. More creativity and inventiveness. Better language skills"* (TCIN_5); with a further teacher observing that *"students behaviour improves if they are engaged and focused, project and group work helps to achieve this, I find it very good as a revision method or as a means for identifying prior learning"* (TCIN_62). A third teacher reported that *"students are more willing to ask questions and lose the fear of being wrong"* (TCIN_42), while a further teacher observed that *"there is more of a buzz and students are more relaxed"* (TCIN_37); with a further teacher reported that they had observed that "*students take more ownership of their learning and therefore things like engagement and classroom behaviour are improved"* (TCIN_24). However there were also some instances where there were "*no obvious changes in behaviour seen"* (TCIN_60), while a further teacher was concerned that *"the good students have a great time"* (TCIN_58), but that sometimes students who are not 'engaged' require additional encouragement to take part.

Further qualitative analysis confirms that teachers observed that using a collaborative approach to teaching motivated students and created computing learning experiences, which were perceived as 'industrious.' One teacher noted that students were more enthusiastic in their learning and had noticed a "*more animated learning environment is apparent*" (TCIN_25). A further teacher reported that they had observed a more "*busy, happy, industrious atmosphere*" (TCIN_22); while another teacher reported that they had observed "*busyness with a purpose*" (TCIN_4). Industrious, busy and happy students, was a theme shared by other teachers, with one teacher stating that they had observed "*engagement, interest, motivation all positive. Ownership of learning*" (TCIN_37); with a further teacher reporting, "*learners are more engaged*" (TCIN_34); and that there is *"more interaction. More noise. Fun*" (TCIN_50); also that "*students are more engaged in lessons and enjoy them more*" (TCIN_19). One teacher noticed a "*willingness to try things out & make mistakes, increased cooperation & communication*" (TCIN_30); with another teacher observing that students were "*more reliant on each other and less so on teacher*" (TCIN_43). On teacher reported a change in their own confidence through using the Bridge21 model and shared: "*I am not scared of technology so I am more likely to use it in my teaching*" (TCIN_18), with one computing teacher reporting that they had observed "*more interest from students in using IT than sitting in regular classrooms*" (TCIN_33) with a further teacher reporting that for the Bridge21 CS session they ran in their school, they observed, "*students volunteered to participate after school*" (TCIN_46). Teachers also observed that students' enjoyed taking the lead in their learning, investing energy in creating their work.

6.6.6 Discussion

Analysis of demographic data from the teaching computing in schools questionnaire confirms that the sample contained a mix of primary and post-primary teachers. While 15% of respondents did not identify their professional context, 63% of respondents identified as teaching at post-primary level, with a further 22% teaching at primary level. Further analysis of the sample explores the context in which computing is taught in schools. Findings show that 33% of teachers reported teaching computing as part of the curricula, with 5% self-reporting that they taught the NCCA short course in coding. A further 9% of post-primary teachers planned on teaching the NCCA leaving advanced certificate course in computer science. A further 14% of teachers reporting that they were active in providing assistance to peers who wanted to learn more about computing. Having examined who taught computing, and where in the curricula computing is taught, further analysis explored which elements of the Bridge21 model teachers incorporated into computing lessons. Furthermore, teachers also provided evidence demonstrating that integrating computing into the curricula is complex, with schools taking time to plan for and resource computing courses.

The Bridge21 pedagogical model contains eight elements (Table 2, p. 7), which combine to support collaborative teaching and learning experiences. A typical Bridge21 learning experience seeks to encourage students to work collaboratively to complete tasks. Bridge21 learning experiences also involve teachers switching role from that of leader, to the role of facilitator and guide, with students encouraged to take individual responsibility for their learning.

Analysis of teacher data reveals that teachers most used a facilitator approach to teaching computing. Teachers were also proactive in encouraging collaborative learning in computing lessons, with teachers organising tasks to encourage students to learn and explore computing concepts together with their peers. Teachers also reported that they used open-ended questioning as a technique to encourage students to think about the answers to questions. Teachers also incorporated student reflections and included tasks, which encouraged students to set their own learning goals, as well as incorporating tasks focused on developing students' practical skills.

Investigating barriers experienced by teachers involved in implementing CPD content and methods, enables CPD designers to include activities which help teachers develop strategies for removing potential barriers (Kennedy, 2011). In particular there is a need to explore barriers to teaching computing (Ko et al., 2004) so that CS CPD designers can construct activities which help teachers develop strategies to address these barriers. Thus, analysis was undertaken to explore what methodological and content related barriers teachers experienced through implementing the Bridge21 model. These results confirm that teachers perceived a lack of an appropriate technical infrastructure as the main barriers to using the Bridge21 model to teach computing, as well as a lack of time to practice programming. Further barriers identified by teachers included lack of access to formal training, lack of accreditation and timetable restrictions where next perceived as barriers, with teachers wanting recognition for attending computing CPD, as well as more flexibility with timetables to be able to implement practical work. These results confirm that teachers were active in implementing Bridge21 but that deeper implementation was restricted in part due to a lack of technical infrastructure and time to develop teaching materials.

Having explored barriers to using the Bridge21 model, further analysis explored other methods, which teachers used in the computing classroom. Sentance, Barendsen, et al. (2018) stress that there are a number of different methods teacher can use to teach computing. Examples include completing individual tasks in computing laboratories or computer rooms (Chamillard & Braun, 2000), pair-programming where two students work together to complete tasks (Dybå et al., 2007); project work where students work in small groups (N. M. Webb et al., 1986); and lectures or teacher led presentations (Matthíasdóttir & Arnalds, 2015). Having identified different methods for teaching computing, research exploring other CS methods was designed to explore what approaches other than the use of the Bridge21 model, teachers used in the classroom. Results show that teachers most used peer teaching and student teamwork, and integrated games into computing lessons. Teachers next reported using project work as well as teacher led-presentations, pair programming, online tutorials, and worksheets to a lesser extent. These results compare with the approach provided through the Bridge21 model, with teachers embracing a collaborative, peer-led approach to learning computing, involving students working on projects.

Understanding teachers future CPD requirements is essential in programmes where teachers are involved in designing activities which students will use to enhance their learning (Voogt et al., 2015). Identifying which areas teachers require further assistance with, enables CPD designers to include new content, spend more time exploring methods or give teachers further opportunity to practice developing skills for use with students in the classroom (Hargreaves & O'Connor, 2018). The practical nature of Bridge21 learning experiences called for further research exploring which elements of the Bridge21 model teachers required further

CPD assistance with, in adapting for use in their classrooms. Analysis of Bridge21 CPD requirements confirm that teachers requested most assistance in developing learning challenges, covering problem solving activities, puzzles, brain storming tasks and research and development activities, which were used to encourage students to take the lead in their learning. Teachers next requested further time spend on developing programming competencies, with suggestions for extending the time spent on mastering basic skills. Teachers also requested assistance in organising project planning, exploring strategies, and techniques to support students working on projects as well as assistance with preparing assessment rubrics for grading individual and group contributions to projects. Finally, teachers reported that they were most confident in organising students into groups and facilitating collaborative working.

Qualitative research reveals that teachers integrated the Bridge21 model into teaching computing, with teachers observing an increase in student engagement and perceived motivation to complete computing projects. Student engagement is a theme shared by other research exploring the use of the Bridge21 model, with Byrne et al., (2018) providing examples of students self-organising to complete tasks during 'hackathon' activity, and Sullivan et al., (2015) providing examples of students working independently on projects. Another observation is that teachers planned on creating their own computing content as well as adapting the CPD content for use in teaching. Indeed, Hubbard (2018) provides further examples demonstrating that teachers are growing in confidence to develop their own computing content for use in teaching (Hubbard, 2018). Another theme, which emerged through qualitative analysis, was that teachers reported using facilitation as a method for teaching computing, with teachers providing examples of learning from students and from students learning from each other. Finally, teachers were energised in both build computer programmes, which could be used to demonstrate concepts, with teachers integrating technologies to create learning experiences with involved students working with technology and developing technological skills.

6.7 Limitations with Implementing Bridge21

The re-coding and analysis of N =1,972 CPD qualitative responses revealed that 2 % of teachers (or N = 40 individual respondents) raised limitations with the CPD (Appendix 9.13.10). This is a very small percentage and demonstrates that the overwhelming majority of respondents were very positive towards the overall experience. On those with reservations one teacher suggested that short beginners' courses would be good for teachers with no or limited knowledge in computing, with a further teacher concerned that "*there is no way I can do an* *assignment to the level of the technology teachers. (The workshop) teacher was good. The content was too difficult"* (TCIN_1). This concern was shared with another teacher who reported that they did not "*have the basic computing ability for this assignment*" (TCIN_ 28). A further teacher reported that they require more basic tuition asking for *"a foundational course on the absolute basics"* (TCIN_ 2). Some teachers just wanted *"more hands on training to become competent before starting to engage my students"* (TCIN_ 45). One teacher didn't think they were ready yet to use the Bridge21 approach, "*I don't think that it would currently help, as my knowledge is still quite new*" (TCIN_24), with a further sharing the concern that they still lacked in confidence and required further CPD "*(I lack the) confidence to implement today's material unless I got more CPD*" (TCIN_22). Finally, one teacher reflected that for the Bridge21 approach to teaching computing to be successful the "*follow up on this workshop needs considerable investment for schools*" (TCIN_12). These comments show that a small number of teachers required further CPD supports, either through exploring further content or through practicing computing through practical and activity based tasks to gain further confidence. This is in contrast to 91% of teachers (Table 39, p. 177) who agreed that they intended using the Bridge21 model on return to teaching computing in their schools.

6.8 Summary

This chapter explored the research findings and analysed the CPD and school data sets. The CPD findings indicate that teachers' perceptions of collaborative, project-based teaching and learning were positive. Teacher reactions to a facilitator led approach to collaborative learning were also positive, and teachers enjoyed the experience of working with peers in teams to develop computing content knowledge (section 6.2). The Bridge21 model also played a 'significant' role in assisting teachers' master computing content knowledge and the confidence to teach computing (section 6.3). The findings also report an increase in teacher content knowledge, confidence, and ability to teach computing, with teachers intending to use Bridge21 model as a teaching strategy (section 6.4).

The school findings reported the inclusion of primary and post-primary school teachers in the sample (section 6.4), with teachers involved in teaching computing within the curricula as well as supporting students learning computing in clubs, and sharing computing knowledge with peers in schools. Further analysis shows that teachers used with most frequency the Bridge21 elements of facilitation in combination with collaborative, group based learning and 'social learning protocols' in their subsequent teaching of CS in schools (section 6.5). Teachers also reported using teamwork, peer learning, and project work as core methods, which they used with their students for teaching computing. Barriers to Bridge21 model implementation included a lack of an appropriate technical infrastructure as well as time to practice programming and a lack of formal training. Teachers also identified the need for additional CPD assistance in designing Bridge21 learning challenges, enhancing programming content knowledge and project planning, and design.

7 Discussion

The previous chapter explored the impact of using the Bridge21 model in a CPD and school context. This chapter starts by revisiting the challenges that teachers face in preparing to teach computing. The aims and objectives of the Bridge21 CS CPD programme are revisited in light of the research findings. The research questions are then used to structure the discussion, and the next section explores teachers' perceptions of the Bridge21 model as a method of CPD. This discussion is followed by a further discussion, which examines teachers' experiences of using the Bridge21 model to teach computing. The research limitations are then discussed, and this is followed by the conclusions. A final section explores the research contributions and examines areas for further research.

7.1 Teaching CS in a 21st Century Context

Teaching in the 21st century is complex, which Voogt, Erstad, Dede, and Mishra (2013) attribute to designing lessons which encourage students to play a greater role in their learning. One of the factors that make teaching today different from teaching in the past is that teachers face increasing pressure to integrate technology into teaching and learning experiences (Zawacki-Richter & Latchem, 2018). In addition to this, European Union (EU) member states and national governments worldwide are involved in implementing new curricula in Computer Science (CS) (Makki, O'Neal, Cotten, & Rikard, 2018). Ireland is one EU member state which is actively involved in integrating computing as a main stream subject across primary and post-primary education (NCCA, 2018a, 2018e). Indeed, the Department of Education and Skills (DES) in Ireland are engaged in the process of rolling out a Coding syllabus for primary schools and an advanced level Computer Science Subject for upper secondary students. There is also a short course in Coding for lower secondary students (NCCA, 2014a) so the subject is now present in all levels of the Irish education system.

The requirement to teach computing in schools has created a demand for teachers qualified to teach the subject (E. Roberts, 2018). However, teaching and learning computing is difficult. A key difficulty is that students' experience in learning computing is that of applying concepts from one context to another (Du Boulay, 1986). Du Boulay (1986) further advises that students have difficulty with linking concepts, making generalisations without understanding basic concepts, and solving computational problems. Teachers face similar difficulties. Robins et al. (2003) report that teachers face difficulties with developing relevant content knowledge, designing learning experiences, mastering sufficient methodological expertise, and developing the confidence to help students with problem solving in a computing context. This has prompted teachers to source professional development programmes to develop the necessary content knowledge, computing skills and methodological expertise to teach these new subjects (Neutens & Wyffels, 2018). Papadakis (2018) argues that there is no one approach for helping teachers prepare for teaching computing. Rather, as Webb et al. (2017) suggest what teachers need is the opportunity to develop "*subject knowledge alongside their pedagogical knowledge in order to respond to rapid curriculum change*" (p. 465). Furthermore, the fundamentals of computing may not have changed (Yeh, Good, & Musser, 1973), but the context in which they are applied has changed (Mouza, Yadav, & Ottenbreit-Leftwich, 2018).

CPD programmes are emerging which aim to provide learning experiences designed to give teachers the opportunity to develop pedagogical content knowledge, and to explore teaching strategies which encourage more classroom discussion, project-based and student-centred learning (Hargreaves & O'Connor, 2018). Collaborative professional development programmes provide learning experiences which encourage teachers to play an active role in their learning, with tasks designed to encourage teachers to share their content knowledge as well as their professional expertise (Kennedy, 2011). Collaborative programmes in computing are built around project-based and problem solving tasks, and involve teachers working in teams to build computing artefacts, giving teachers the opportunity to develop practical expertise in computing and programming (Sentance & Csizmadia, 2017a). A collaborative, and project-based approach to professional learning aims to encourage teachers to explore strategies and discuss ideas as well as put into practice concepts which they intend using with students in a classroom context (Walker, 2018).

In a CS context, Hamlen et al. (2018) propose that collaborative, project-based and technology mediated professional learning experiences offer the potential for computer science teachers to not only develop a practical understanding of computing, but also develop relevant strategies for teaching the content. Recent examples of collaborative, project-based CS CPD programmes are provided by Sentance and Csizmadia (2017b)[6] and Hazzan et al. (2014). The programmes described by Sentance and Hazzan call for an integrated and collaborative approach to professional development, to equip teachers with the confidence, conent knowledge and the professional expertise to introduce computing learning to increase student engagement in CS.

Sentance and Csizmadia (2017b) propose a particular model of CS CPD, which combines activities to encourage teachers to work towards a common goal including sharing

---

[6] *See also Sentance et al. (2013); Sentance et al. (2012)*

experiences, expertise and content knowledge. The model is also practice based, with a focus on exploring pedagogical issues in a collaborative context. A peer teaching / coaching approach to professional learning encourages teachers to share practical knowledge, with teachers sharing experiences, practices and strategies. A further core element of the programme is the facilitation of teachers in research, with teachers provided with an academic infrastructure to oversee the design, implementation, and evaluation of computing learning experiences implemented in schools. The CPD aims to create a 'holistic' approach to CS CPD, with teachers encourage to 'cascading' knowledge, and further contribute to the professional development of colleagues, who share the same pedagogical values and beliefs. A final and important part of the CPD is the link between the assignments and accreditation, giving teachers the professional acknowledgement and status that they need to implement computing in schools. Sentance and Csizmadia (2017b) provide a blended and self-directed approach to CS CPD. In contrast, the B21 CS workshop modules offer student-centred learning experiences, with teachers working in teams during computing workshops to cover the pedagogy and processes as well as the content knowledge that are needed to teach computing.

The rational for designing collaborative and project-based CPD comes in response to system wide changes in education, with Papadakis (2018) suggesting that traditional teaching approaches, including lectures and self-directed study, are unable to contribute substantially to the development of the necessary cognitive models by the students. Furthermore, Phillips et al. (2017) argue that teachers need access to CPD which provides the opportunity to develop alternative teaching approaches, which help students develop strong cognitive models, which can aid problem solving. Rather, Crook (2018) further recommends supporting teachers master strategies for facilitating collaborative activities, give students the opportunities to discuss and critique what they have learned. While Haduong and Brennan (2018) offer specific examples such as using 'debugging' as a collaborative activity to introduce students to domain based skills such as solving skills. Teague and Roe (2008) further add that collaborative activities are essential in helping students develop the confidence and practical expertise to write computer programmes, with Luckin and Du Boulay (1999) reporting that collaborative learning experiences create the opportunity for more able others to help colleagues explore computing concepts, assisting with developing strategies and meaning making.

The above examples demonstrate the benefits of collaborative learning in a CPD context and propose that teachers need access to CPD, which equips them with the pedagogical content knowledge, practical skills, and confidence to facilitate students working on projects, and completing tasks independently of the teacher. Brown (2006) continues that teachers need assistance in developing knowledge, as well as the technical and methodological expertise for "*liaising with groups, looking at what issues are unfolding, and occasionally interrupting the class to address a particular issue*" (p. 19). Furthermore, there are those who argue that CS CPD programmes should provide teachers with the opportunity to develop their "*subject knowledge alongside their pedagogical knowledge in order to respond to rapid curriculum change*" (M. Webb et al., 2017, p. 465). However Gretter and Yadav (2016) continue that it is essential that teachers are also provided with the means to develop the knowledge and expertise to oversee the "*integration of 21st century skills into the classroom*" (p. 19), in accordance with structural changes in post-primary education

Bridge21 is a social constructivist model of 21st Century Teaching and Learning which is used by post-primary across teachers across Ireland to encourage collaborative, team-based, technology-mediated learning in their subject teaching (Lawlor et al., 2018). Trinity College Dublin (TCD) is using the Bridge21 pedagogical model as the underlying pedagogical approach for a CS CPD certificate. The Bridge21 CPD programme is designed for in-service post-primary teachers planning to teach computing at second and primary level (see http://bridge21.ie/cs/). Teachers can select computing workshop modules covering computing concepts including Computational Thinking, Computer Programming, and Hardware. Each workshop uses the Bridge21 pedagogical model, with teachers learning computing through collaborating on projects, and completing technology-mediated activities supported by their peers (TCD, 2017).

Inspired in part by Vygotsky's (1978) ideas of social constructivism in a peer learning context, the Bridge21 pedagogical model contains eight elements (Table 2, p. 7), which combine to create collaborative teaching and learning experiences. A Bridge21 learning experience seeks to encourage students to work collaboratively to complete tasks, which aim to nurture learner autonomy, with facilitators working with teams to encourage equal participation in project-based tasks. Bridge21 learning experiences involve teachers switching role from that of leader, to the role of mentor and guide, with students encouraged to take individual responsibility for their learning. Implementing the Bridge21 model requires teachers to make a step change in teaching practice, with teachers using the elements of the Bridge21 model to encourage student-centred learning, with teachers facilitating students, and students encouraged to engage with teachers and share their learning.

To revisit Vygotskian theory (1978), meaningful learning, demands collaboration, with the Zone of Proximal Development (ZPD) providing a lens to explore the potential of working in a team, assisted by peers to deepen knowledge and expertise, in context. Finally, Bridge21 learning experiences are structured to support the social construction of knowledge through peer collaboration. The results presented in this thesis confirm the importance of peer collaboration in helping teachers construct knowledge and develop the expertise to teach computing. To conclude, Vygotsky (1978) suggests that learning collaboratively, and interacting with peers, offers the potential to help learners address knowledge gaps and develop practical strategies.

7.2 Bridge21 CS CPD Programme Aims and Objectives

The Bridge21 CPD programme aligns with literature, which argues for a collaborative approach to teaching and learning. The Bridge21 CPD workshops use what Olsen (2015) calls a '*knowledge construction model*' which means that learning is supported through completing social and practical tasks which encourage the construction of knowledge. Furthermore Du Boulay (1986) cautions that learning computing can be disorientating for those with no prior experience in the domain. While Papadakis (2018) argues that traditional teaching approaches are limited in helping learners develop experience and explore relationships between computing concepts. Indeed Mishra and Henriksen (2018) suggest that teamwork combined with project-based learning provides a way to help students develop solutions to domain based problems. Mishra and Henriksen (2018) and Ben-Ari (2004) as well as Zendler (2018) argue that using teamwork in CS lessons creates opportunities for students to develop the confidence and skills to share understandings, knowledge and expertise.

The Bridge21 CPD programmes consists of five computing modules, which when followed in sequence, are designed to help teachers develop a practical understanding of the content knowledge and pedagogical methods required to teach computational thinking, visual and text based programming and hardware. Each of the Bridge21 CPD workshops are designed to provide teachers with a practical understanding of computing and engage teachers in mastering a number of practical tasks which they can use with students in a classroom context. The need to equip teachers with a practical understanding of computing is reflected in the CPD aims. First, the CPD aims to provide "*in-service teachers with the requisite knowledge, skills and competence to support the development of an innovative learning culture within schools, which is team-based, technology-mediated, project-focused and cross curricular"* (TCD, 2018, p. 1). Second, the CPD aims "*to enhance the expertise of participant teachers in new models of teaching and learning with particular emphasis on Science Technology Engineering Maths/Computer Science"* (TCD, 2018, p. 1). Each of these aims combine to support teachers develop a practical understanding of computing designed for classroom teaching.

The Bridge21 CPD computing workshop modules use learning experiences designed to assist teachers construct knowledge and expertise through social interaction. Darling-Hammond (2017) reminds us that an important function of CPD programmes is to help teachers develop confidence in the methods which are designed to work in harmony with the system in which they will be used. The Bridge21 CS CPD workshop modules emphasise learning by doing (Hardy, 2012) to give teachers the opportunity to develop practical experience in collaborative, project-based methods which align with reform spanning the Irish education system.

7.3 Revisiting the Research Questions

Collaborative activities incorporating collaboration, projects and teamwork, and a facilitation led approach to teaching are perceived play an important role in helping teachers' master theory and processes that are used to teach computing concepts (Guzdial, 2016). Moreover, learning experiences which encourage peer collaboration and the sharing of professional expertise creates an environment which is designed to help the learner 'succeed' (Ridgway & Passey, 1991). The adaptation of the Bridge21 model as a CS CPD method generates an opportunity to investigate what impact a collaborative, project-based and technology mediated approach to professional learning plays in equipping teachers with the content knowledge, confidence and expertise to teach computing. Two research questions were designed to explore these issues. The first question explores what are teachers' perceptions of the Bridge21 model as a method of CPD (section 7.4). The second question examines what are teachers' experiences of using the Bridge21 model to teach computing (section 7.5).

7.4 Teachers' perceptions of the Bridge21 model

This research set out to explore teacher perceptions of a model of professional development designed to support teachers prepare for teaching computing within the context of Irish educational. Research was designed to understand what role a collaborative, project-based, and technology-mediated approach to teaching and learning (the Bridge21 pedagogical model) played in assisting teachers develop the confidence, content knowledge and the methodological expertise to teach computing. Understanding teacher perceptions of educational models and their impact on teaching offer the potential to inform policy and provide direction in the design of CPD programmes. For example Day and Leitch (2001), de Vries, Jansen, and van de Grift (2013) as well as Kennedy (2011) argue that research exploring teacher perceptions give valuable insights into teachers beliefs and motivations for changing their teaching methods. Moreover, research exploring teacher perceptions can reveal personal, school-related and system wide factors impacting upon education (McMillan, McConnell, & O'Sullivan, 2016). These studies argue that research exploring teacher perceptions gives new insights into teaching, and captures the teachers' perspective for transforming methods.

There are a number of studies, which have implemented research designs to explore teacher perceptions of professional learning interventions. Naugle et al. (2000) call for research exploring teacher perceptions of their own teaching performance, and Coldwell and Simkins (2011) highlight the need to examine teacher perceptions of their learning outcomes. Rubin (2018) further suggests exploring teacher perceptions of the CPD content, intentions to use the CPD content in their teaching. In addition, Carl (2009) suggests that an essential part of the CPD evaluation process is to ensure that teachers' voices are heard, and that their CPD needs are considered. Further research by de Paor and Murphy (2018) argue that listening to teachers is essential in preparing CPD which meets teacher needs as it provides feedback which can help designers adjust content to ensure that it is relevant to practice. These examples argue for exploring teachers' perceptions of CPD interventions, to uncover rationales, motivations and intentions to use CPD content and methods in a practical context. The following discussion explores teacher perceptions in relation to the Bridge21 CS CPD programme structured according to the research questions (Table 1, p. 5), with the first section (Section 7.4.1) exploring teacher reactions to the CPD workshop content.

7.4.1 Q1.1 what are teachers' reactions to the CPD workshop content?

The first step in the CPD analysis was to explore teacher reactions to the CPD, which, according to Kirkpatrick (2007) provides a 'measure of customer satisfaction' as well as indicators in terms of "*how the attendees feel about the program*" (p. 1). Reaction research, conducted at the end of learning interventions, enables teachers to express their initial feelings towards new content and methods (McMillan et al., 2016). Designing tools to explore teacher reactions also provides an opportunity to explore teacher attitudes towards new content and using new teaching methods.

A key finding is the level of positive agreement across all reaction variables over time, which confirms that teachers enjoyed the CPD. A further finding is diversity in age, and workshop representation demonstrates the depth of professional experience and variety in computing workshop content, which was covered in the CPD sample over five years. These findings are discussed in the following analysis that revisits the reaction results reported in section 6.2.

Initial quantitative analysis (section 6.2.2) exploring teacher reactions captured feedback which revealed elements of the workshop content, teaching methods and activities that teachers enjoyed most or found challenging. This analysis proved useful in finding out which elements of the CPD teachers enjoyed as well gathering feedback on methods and examples which teachers found most useful. Analysis of all reaction variables confirmed that 93% of teachers agreed that a facilitator lead approach to teaching gave them the time to direct their learning. Also 91% of teachers agreed that the workshop examples were helpful 91%, with a further 90% agreeing that the pedagogical process was satisfactory. The depth and range of programming activities were also perceived as appropriate (88% agreement), with 84% of teachers also satisfied with the curricula content and its level of difficulty. Finally, 89% of teachers confirmed that the CPD met their expectations.

Further qualitative analysis complementing the quantitative analysis conducted to add context to the quantitative results show that teachers enjoyed the one day workshop experience, confirming that a facilitator led CPD approach gave them the freedom to choose topics to research and explore, supported by their peers. Teachers also reported that the workshops provide a safe environment to share, discuss, and explore methods and content used for teaching computing lessons. Teachers provided examples, which demonstrated that the pedagogical process provided a structure to introduce problem solving into computing lessons, with teachers giving further examples, which demonstrate how computing activities could be adapted to give students a contextual view of computing, with the workshops providing resources for teachers.

A second wave of quantitative analysis explored teacher reactions to the CPD, organised by year (Section 6.2.3). Analysis of the year on year results confirms that teacher reactions to the workshops remained positive across time, with teachers in year 1 agreeing that the workshops were worth attending compared to year 5. Similar patterns were reported for a facilitator approach to CPD, and the quality of examples used in the CPD. Teachers also confirmed that the pedagogical process was satisfactory and that the programming activities were sufficient for supporting teachers in their preparation for teaching computing. Teachers self-reported that they were very satisfied with the curricula, and agreed that the level of difficulty was appropriate. Finally, comparing CPD expectation variables between years 3 and years 5 reveal that in teachers agreed that the workshops continued to meet with their CPD expectations. These comparative results show strong positive reactions to the Bridge21 CS CPD workshop model, with sustained teacher satisfaction, across time.

Deeper qualitative analysis (section 6.2.4) exploring teachers self-reporting on key learnings they took from the CS CPD workshops confirmed that teachers reacted positively to a collaborative approach to learning computing with their peers and facilitator-mentoring approach to teaching. Exploring collaboration first, the analysis showed that teachers enjoyed the experience of learning computing through collaboration. For example, teachers reported that group work made learning computing less stressful and that groups are good for exploring difficult concepts. Furthermore, the analysis shows that teachers sought out expertise within their peers groups, with teachers also reporting that they learned from listening to others and through sharing their ideas and practices with other teachers.

In summary, the analysis shows that teachers perceived that learning in a team is fun and instructive. Teachers also reacted positively to a facilitator led approach to collaborative leaning. For example, one teacher observed the technique of stepping away with a further teacher observing the practice of giving learners time to think about the content that they were exploring. Teachers also observed facilitators using techniques to encourage team members to explore new concepts and were encouraged to give something a go even if they did not feel that confident. Teachers observed facilitation and saw how they could encourage their own students to construct knowledge rather than being reliant on the teacher providing answers. The one day computing workshops were perceived as a safe space where teachers could work alongside colleagues, with from different subject areas and experiences, to develop projects and tasks for students, which promoted a team-based approach to teaching and learning. Teachers also reported that they were equipped with the programming expertise and the outline of a pedagogical process, which they could use for structuring the integration of computing activities into the curriculum.

These results are similar, and compare with other studies which suggest using collaborative, project-based and technology mediated workshops to help teachers engage with computing and develop strategies for teaching the content (Cutts, Brown, Kemp, & Matheson, 2007; Dorling, 2016). Cordingley, Bell, Evans, and Firth (2005) suggest that collaborative, project-based and technology mediated professional learning programmes are linked with positive outcomes; in that they provide a form of CPD which helps teachers develop ownership and control in personalising their learning.

Finally, collaborative professional learning programmes provide teachers with the opportunity to have learning conversations with peers and explore how new methods and content relate to classroom practice. However reaction analysis is criticised for providing 'superficial feedback' at the end of professional learning interventions (K. G. Brown, 2005; Sitzmann, Brown, Casper, Ely, & Zimmerman, 2008). Champion (2003) provides a counter argument stating that reaction data can provide useful data that can be used to "*gauge participant reactions during professional development programs help leaders spot trouble areas and know when and where to adjust the program midstream. Participant reaction data also can help validate the program's design*" (p. 75). Furthermore Ekmekci et al. (2018) argue that reaction analysis can also provide a way to respond to and address teacher's needs. Crook (2018) suggests that assessing teacher reactions is particularly important as "*we still have only a limited insight into what defines 'effective' collaboration*" (p. 121).

To conclude, the reaction findings generated in this study match with Kirkpatrick's aims for reaction analysis in that the data confirms that teachers reacted positively to the training intervention and were satisfied with the CPD. They show that teachers responded positively to a collaborative, project-based and facilitator led approach to learning computing, with teams playing a vital role in helping teachers explore concepts and practice computing tasks supported by their colleagues. These results compare with Melcer and Isbister (2018) who suggest that practical tasks involving making computing artefacts have a greater positive impact on professional learning, The reaction findings also confirm that a collaborative, project-based and facilitator led approach to learning computing gives teachers the confidence to take part in computing activities. Again Melcer and Isbister (2018) observed similar results in that collaborative working can help to reduce programming anxiety. These results fall in line with supporters of a collaborative, project and practical approach to teaching and learning computing (Ben-Ari, 2001; Ridgway & Passey, 1991; Sentance, 2018b); who agree that learning computing with peers plays an crucial role in helping teachers develop the content knowledge, and the computing expertise as well as confidence to teach computing.

7.4.2 Q1.2 what content knowledge did teachers learn?

There are a number of barriers to learning computing, and programming which professional development programme need to address, to help teachers prepare for teaching the subject. Grover et al. (2018) report that a core problem that learner, new to programming face, is that they do not understand the meaning of codes that are used in programming languages while Samurcay (2013) suggests that learners struggle with mastering problem solving methods. Ko et al. (2004) reports that learners can also struggle with understanding design issues surrounding the construction and implementation of computer programs while Du Boulay (1986) continues that learners can struggle with misunderstand the nature of programming and lack the content knowledge to understand relationships between code, programmes and machines. Pea (1987) reports that a core barrier to learning programming relates to learning how to transfer concepts from one context to another.

One way to explore knowledge gain within a CPD context is to put in please measures, which explore teacher perceptions of learning outcomes. Acknowledging the constraints of designing research around the measurement of learning outcomes (section 2.5.2), the testing of pre and post workshop responses to outcomes linked to programme objectives provide indicators demonstrating teacher perceptions of acquired knowledge and professional expertise. Guskey (2000) and Kirkpatrick (1994) both agree that measuring learning outcomes provides useful information in terms of enabling training participants to report on their perceptions of their learning as they relate to the training intervention aims and overarching objectives. Guskey (2000) continues that learning outcome analysis plays an important role in helping to confirm a perceived increase in content knowledge and subject expertise. While Kirkpatrick (1994) further argues that 'measuring learning outcomes' is an important phase in professional development analysis because if learners do not meet their learning 'objectives' then 'you can't expect a change in behaviour.' Following Guskey (2000) and Kirkpatrick (1994), the researcher designed pre and post-tests learning outcomes measures to explore teacher attitudes to the workshop content and teacher confidence in using the workshop methods. Furthermore, the addition of pre-test learning outcome measures enabled the researcher to explore the changes reported by teachers in the post-test results (Bonate, 2000).

The following analysis explores teacher perceptions of learning computing knowledge, processes, concepts, as well as the practical ability to plan and prepare for implementing the Bridge21 model in a classroom context. A key finding is that teachers reported gains in content knowledge, confidence, and understanding of computing concepts and processes, with teachers also providing examples of the impact that a collaborative approach to learning had on developing computing content knowledge they reported they could apply in a classroom context (section 6.3).

Initial quantitative analysis (section 6.3.2) explored content knowledge developed by teachers using scales which were designed to capture a change between pre-test and post-test learning outcome variables developed from the Certificate programme handbook. The outcomes are statements of achievement that teachers are expected to achieve on completion of each computing module. The researcher adapted each module outcome to explore teacher perceptions of their learning in relation to each computing module aims and objectives. Analysis of the pre-workshop learning outcome variables shows that 58% of teachers agreed that they were not confident in their ability to plan Bridge21 activities prior to the workshop in contrast to 76% of teachers who reported that they were more confident in the post-test results gathered after the workshop. Furthermore, 65% of teachers reported that they were not confident in their ability to program before the workshop in contrast to 80% who agreed that they were more confident after the CPD workshop. Similarly, 59% of teachers reported that they did not have a good understanding of processes before the workshop, compared to 83% after the CPD. Before the workshops, 65% of teachers indicated that they did not have basic understanding of computer concepts. After the workshops 79% of teachers agreed that they had developed a greater understanding of basic concepts. Prior to the CPD, 67% of teachers reported that they did not feel confident in their ability to teach computer programming using the Bridge21 model compared to 75% of teachers who agreed that they were more confident after completing the workshop. Finally, 60% of teachers stated they did not have adequate computing content knowledge before the workshops compared to 75% of teachers agreeing that they had gained in content knowledge. These results show a shift in teacher perception of their learning, capturing gains in confidence, ability and knowledge and methods.

Further qualitative analysis complementing initial learning outcome analysis reports that pedagogical process provided tasks and activities, which motivated teachers, to keep on task and to finish projects. The workshop provided a mix of activities and task as well as a process that teachers reported that they were confident to use in the classroom, which teachers sharing that they were more confident in completing programming tasks. While teachers reported that the tasks were complex, colleagues provided peer support and encouragement, offering help and guidance to complete tasks. Teachers reported learning a process for teaching computing, while also developing and strengthening their content knowledge and skills to assist their students with problem solving.

Subsequent quantitative analysis exploring year on year change across learning outcome variables (section 6.3.3) provides further insight into teacher perceptions of their learning. Post workshop results for 2014/2015, shows that 85% of teachers agreed that they had a greater understanding of basic programming concepts, with 84% of teachers also agreeing that they had a greater understanding of computer programming processes. Finally, 77% of teachers reported that they were confident in their ability to teach computing using the Bridge21 model after attending the workshops. This trend continues in 2015/2016, with post workshop results confirming that 78% of teachers reported that they were more confident in their ability to program, with 75% of teachers reporting a greater understanding of basic programming concepts and 71% of teachers agreed that they were confident in their ability to use the Bridge21 model in the CS classroom. The introduction of pre-tests in 2016 generated the opportunity to explore differences between pre and post results. Examples include 75% of teachers agreeing that they were more confident in their programming ability, with 80% reporting that they had a greater understanding of programming processes after attending the CPD workshops. Teachers also reported an increase in programming content knowledge (74%) as well as an increase in confidence in teaching programming using Bridge21, with 70% of teachers confident in their ability use the Bridge21 model.

Similarly, analysis of pre and post-test results for the 2017/2018 academic year confirm teachers as very positive in their attitudes towards using the Bridge21 model. For example, 93% of teachers agreeing that they were confident in planning programming activities using the Bridge21 model, and 84% of teachers agreed that they were confident in their ability to teach computer programming using the Bridge21 model. Furthermore, 94% of teachers reported that they had gained a greater understanding of programming processes, with 90% of teachers reporting that they had gained in computing content knowledge. These results confirm a change between pre and post results, however further analysis was required to verity the statistical significance of this change.

A further wave of quantitative analysis (section 6.3.4) explored the statistical significance of the change reported in post-test learning outcome results. The results confirm that teachers attending Raspberry Pi workshops gained most in programming content knowledge and confidence, as well as expertise in their preparation to use the Bridge21 model. This is reflected in a five point change across all pre and post-workshop median scores. Teachers reported increased confidence in their ability to teach Scratch with a median change of four points in programming content knowledge and four points for programming ability. Teachers reported that they were confident in planning Bridge21 python exercises, which is demonstrated through the biggest median change of three and a half points. These results provide a representation of teacher perceptions and confirm that teachers perceived themselves as confident in applying what they had learned.

Additional quantitative analysis (section 6.3.5) investigated the statistical significance of the change over time, analysing teacher first pre-test, and last post-test scores. The researcher used a Wilcoxon (1945) Signed Rank Test on a sub-set of responses which compares two medians to determine whether they are statically different enough to show actual difference between two populations. The analysis confirmed a four point change between pair matched pre-test and post-test median scores (section 6.3.5.1). This change in median score between two related samples confirms a gain in teacher knowledge, expertise, and confidence to use the CPD content and methods. Moreover, these results indicate that teachers were more confident in their ability to programme, and had a greater understanding of concepts. The results also confirm that teachers were more confident in their ability to plan Bridge21 activities, and to teach programming using Bridge21 the model.

Deeper qualitative analysis (section 6.3.6) asking teachers to reflect on what they might now do differently having attended the CPD, explored teacher perceptions of using the Bridge21 model in a teaching context. Exploring programming content knowledge first, the analysis shows teachers developed a basic understanding of programming languages. For example, teachers reported that they understood computing concepts, and could relate what they had learned to their teaching. Earlier examples were linked to later activities and projects provided teachers with the opportunity to consolidate their learning. Teachers covered core computing concepts topics such as concurrency, and Boolean Operators. Moreover, teachers also reported that they had developed the confidence to teach the content of the workshops to their students. For example, one teacher reflected that they had developed the confidence to help their students with a further teacher reporting that working in teams to complete projects provided the motivation to learn by doing. These examples show that the CPD workshops had a positive impact on teacher learning, with teachers reporting that they were confident in teaching computing through the Bridge21 model.

These results confirm teachers developed basic content knowledge to assist them in introducing computing into schools. The quantitative results report a change in pre and post-test results and qualitative data analysis provides insights into teacher attitudes, knowledge and skills. Kirkpatrick (2007) reports that establishing that learning has taken place, gives CPD provider's important information highlighting what worked and what did not work in CPD offering. The results confirm that a collaborative, and project-based workshop in computing gives teachers the time to develop skills and knowledge in a form which they perceive that they can practically apply in the classroom. These results compare with other studies which argue that collaboration plays an important role in enabling teachers to share experiences and knowledge, gaining expertise to teach new content, as well as supporting others to put into practice ideas and concepts in a peer supported environment (Crook, 2018; Voogt et al., 2015). Moreover, a collaborative approach to CPD enables teachers to direct their learning (Kennedy, 2011), empowering teachers to ask questions or seek guidance to solve problems which are particular to their needs and specific to their teaching contexts (Sentance & Humphreys, 2018). These results confirm that collaborative professional development provides a platform for teachers to meet, share experiences, explore problems, and developing teaching strategies through completing projects (Cutts et al., 2017; Sentance et al., 2014).

The learning results also confirm a change in teacher confidence, after attending the CPD. Paraskeva et al. (2008) argue that building confidence is key to the success in computing with Pareja Roblin et al. (2018) highlighting that one of the key barriers that teachers can face when using technology is a lack of confidence. Mishra and Henriksen (2018) suggest that building confidence should be an outcome of computing learning experiences, as computing is a creative process. Indeed Beecher (2018) argues that without confidence, computing professionals can find it difficult to explain problems or communicate ideas to project teams or colleagues. What the above results confirm is that not only did teachers increase in their confidence; but that teachers also reported that they had a greater understanding of computing, as well as the practical expertise to teach computing on return to the classroom. What this means is that the CPD workshops provided teachers with an experience which enabled them to build in confidence and knowledge, as well as developing expertise in facilitating activities designed for the classroom.,

Finally, these examples shine light on the strategies developed by teachers to address some of the computer programming difficulties raised by Du Boulay. For example, teacher gave examples demonstrating the use of problem solving to overcome computing problems, showing understanding of "*what kinds of problems can be tackled and what the eventual advantages might be in expending effort in learning the skill*" (Du Boulay, 1986, p. 57). Furthermore, teachers also shared examples demonstrating connections between hardware and operating systems, demonstrating an understanding of "*the general properties of the machine that one is learning to control*" (Du Boulay, 1986, p. 57). Teachers provided examples of computer programmes, and described modifications, changes and plans for adapting the content, demonstrating mastery of the "*mastering the syntax, and the underlying semantics*" of programming (Du Boulay, 1986, p. 57).

To conclude the learning results show that teachers perceived that they were more knowledgeable, more confidence and better equipped with the expertise to teach computing on their return to the classroom. In response to the question what content knowledge did teachers learn, the answer is that teachers reported that they were more confident in their ability to teach computing, teachers perceived that they were more knowledgeable in computing, and that teachers were more prepared the expertise and methods to teach computing on return to the classroom. Finally, the Bridge21 model had a positive impact on teacher learning, with teachers prepared to use facilitation with collaborative, project-based methods in their own teaching. Teachers reported enjoying learning computing with peers. These results are similar to other studies (see Crook, 2018; Hargreaves & O'Connor, 2018) who argue for using a collaborative, project-based CPD approach.

7.4.3 Q1.3 what strategies did teachers intend using for teaching computing?

Having established that the CPD workshops provided teachers with baseline knowledge and expertise in using collaborative teaching methods, further research explored strategies that teachers intended to use to teach computing on return to the classroom. The third step in the analysis examined what strategies teachers intended to use for teaching computing. Kirkpatrick (2007) reports that while the learning analysis confirms what trainees learn, further analysis is needed to capture 'change in behaviour'. Kirkpatrick continues that it can be difficult to 'predict when a change in behaviour will occur' and suggests exploring trainees intentions before they leave training programme. Capturing intentions enables CPD providers to identify which elements of the training that participants plan use in practice (section 6.4).

Initial quantitative analysis (section 6.4.2) exploring teacher intentions to use the CPD content confirmed that teachers intended using the Bridge21 model in their teaching. The intention results show that 91% of teachers intended using the Bridge21 model, with 4 % neutral in their view and the remaining 5 % of teachers' not intending to use the model.

Further quantitative analysis (section 6.4.3) exploring year on year teacher intention to use the Bridge21 model (Figure 6-17, p. 179), confirmed in a change in mean from m =2.06 in 2015/2016, to m =1.55 in 2017/2018. The year 5 results report that 98% of teachers attending the computing workshops intended to use the Bridge21 model, with 1 % neutral in their view and only 1% reporting that they did not intend using the Bridge21 model.

Initial qualitative analysis of teacher responses (section 6.4.4) shows that teachers planned on using facilitation as a method to support their students learn. Teachers also indicated their support for an element of peer-teaching with one teacher planning to encourage students 'who have more computing experience' to act as mentors to other students during computing class, with a further teacher planning to 'maximise peer learning.' Teachers also planned to use worked examples to help give students a contextual view of computing, and planned on using problems solving to help students develop practical expertise. Teachers also planned on using more facilitation in the computing classroom, giving students more opportunities to discuss and share their learning.

Deeper qualitative analysis (section 6.4.5) explored strategies teachers planned on using to increase student engagement in computing, and examples of the CPD content teachers planned on using in teaching, Teachers shared examples which described using more facilitation and group work in the computing classroom, to give students the opportunity for autonomy and self-directed learning, supported by peers. Teachers also shared plans to use the Bridge21 model as a structure to introduce students to physical computing, using Raspberry Pi and Makey-Makey activities as ways to help students work through the set-up, and program tasks supported by these platforms. Moreover, teachers also planned on using examples they had experienced in the CPD context, using worked examples in Scratch which students could adapt and change to practice programming skills. To conclude, these results confirm that teachers intended using the Bridge21 model to teach computing. Teachers planned to adapt the CPD content and use facilitation as a teaching strategy. A collaborative approach to teaching and learning computing created opportunities for teachers to share professional expertise and develop strategies before implementing concepts in the classroom (Ryoo, Goode, & Margolis, 2015). Working in teams helped teachers explore the potential of collaborative learning, with teachers sharing experiences, and planning strategies to use a facilitator driven, collaborative and project-based approach to teaching computing. The results show that the CPD equipped teachers with the pedagogical skills and knowledge, as well as the confidence to teach computing. These findings confirm that teachers leaving the CPD were motivated to use the Bridge21 model to introduce computing into schools. These results compare with research by Papadakis (2018), and Webb et al (2017) who argue that more needs to be done to support teachers improve and deepen their 'pedagogical skills and their CS content knowledge'.

7.5 Teacher experiences of using the Bridge21 model

The previous section explore teacher perceptions of the CPD, in relation to reactions to the CPD content and methods, teacher perceptions of their learning and intentions to use the content of the CPD to introduce students to computing in schools. While these results confirm that the Bridge21 workshops provided teachers with a contextual view of computing, using projects and team-based activities to assist teachers develop knowledge and skills for teaching computing, further research was needed to explore teachers use of the CPD in a school context. Kirkpatrick (2007) suggests conducting further analysis to explore 'trainees' experience of using the training content in the context of their work. Kirkpatrick continues researchers should "*allow time for behavioural change, for some programmes, two or three months after the training is a good rule of thumb*" (p. 7), and suggests that "*six months, is* *more realistic (as it gives) trainees time to get back to the job, consider the new suggested behaviour, and try it out*" (p. 7). Exploding behavioural change is complex (Davies, Nutley, & Walter, 2008), and it is difficult to measure the transfer of knowledge from one context to another (Perkins & Salomon, 1988; Woodworth & Thorndike, 1901). Indeed Perkins (1986) suggests that the transfer of computer programming knowledge is particularly 'fragile'. Thus there is a need to explore what Mayer (1975) calls 'resilience' in terms of exploring what strategies teachers gained in a CPD context and applied in a school context. The following section explores what are teachers' experiences of using the Bridge21 model to teach computing.

A key finding in the demographic data is the level of representation of primary school teachers in the follow up sample, with 22% of respondents in the teaching computing in schools sample self-identifying as primary school teachers, and 63% of teachers self-identifying as teaching at post-primary level, with a further 15% choosing not to self-identify their status. Further analysis of the demographic data reveals that 33% of teachers' self-reported teaching computing as part of the main steam curricula. A further 21% of teachers reported spending time during the school day to supervise or facilitate lunchtime clubs with students in their schools. In addition, 5% of teachers self-reported that they taught the NCCA short course in Coding, with a further 9% reporting that they planned to teach the new NCCA advanced certificate in Computer Science. These finds confirm that the CPD appeals to primary and post-primary teachers, and with teachers attending to support planning and implementations within the school day, but also to support extra curricula computing activities. Having explored initial demographic data, the following section, examines which elements of the Bridge21 model teachers used for teaching computing. Also covered are barriers to using the Bridge21 model and other methods which teachers reported using in their teaching, as well as suggestions enhancing future Bridge21 CPD offerings. The section ends with discussion exploring teacher experiences of using the Bridge21 model to increase student engagement in computing.

7.5.1 Q2.1 what elements of the Bridge21 model did teachers identify as most relevant for teaching computing in practice?

Initial quantitative analysis (section 6.6.1) exploring elements of the Bridge21 model used for teaching computing confirm that 97% of teachers agreed that they most used the facilitator approach. Furthermore, 81% teachers reported that they incorporated student collaboration through teamwork into computing lessons. Data analysis indicated that 86% of teachers in the sample also incorporated 'social learning protocols' into their classroom teaching, which are techniques for nurturing learner autonomy. Teachers next reported that they used student reflection activities (77%) and goal setting (78%) with near equal frequency. Furthermore, 81% of teachers implemented skills focused tasks, such as problem solving. Finally, 62% of teachers endeavoured to configure classrooms to support students working in teams, with teachers reporting that some classrooms were already configured to support group work. These results confirm that the CPD teachers used the Bridge21 model, incorporating Bridge21 elements to teach computing.

Further quantitative analysis (section 6.6.2) exploring barriers to using the Bridge21 model for teaching computing confirmed that the lack of a suitable technical infrastructure (73%) and a lack of time to practice developing programming skills (78%) and a perceived lack of access to formal training in CS (69%) were core barriers to implementing Bridge21. Teachers were less concerned that a lack of access to accreditation (38%) was a perceived barrier to using the Bridge21 model. Further analysis shows that teachers cited practical incompatibility with the timetable (54%), as well as a lack of time to implement the Bridge21 model as further barriers (53%) to using the Bridge21 model. The analysis also shows that a lack of peer support (38%); incompatibility with the timetable (14%); lack of confidence (24%); lack of demand from students (22%) and lack of support from senior management (8%) as lesser barriers to implementing the Bridge21 model. In contrast, teachers reported a 78% success rate in using the Bridge21 model, with teachers trying the model and confirming its compatibility with their teaching. These results confirm that teachers perceived a lack of access to suitable technology and time to keep their programming skills up to date as key barriers.

Additional quantitative analysis explored other methods teachers used for teaching computing, providing comparative data to the Bridge21 model (section 6.6.3). This analysis explored other methods, which teachers used for teaching computing in addition to or in tandem with the Bridge21 model. Analysis of the results, confirm that teachers mostly used student teamwork (96%) and peer teaching (96%) as alternative methods for teaching computing. Teacher next reported using teacher presentations when teaching computing (82%) as well as pair programming (74%), as teaching strategies. Games were also used to teach computing, with 96% of teachers using game based activities, which encouraged game play as a strategy for learning computing. Similarly, a project-based approach to computing was also used in the classroom (85%) as a scaffold to structure computing work. Teachers least used online tutorials (55%) and printed worksheets (54%), showing that teachers were confident in using methods which involved interacting with students learning computing. Analysis of supporting qualitative responses provide examples of teachers reporting that they used a blended approach to teaching computing, using facilitation and projects, as well as problem solving with short presentations and games to give students a contextual view of CS.

Final quantitative analysis (section 6.6.4) examined teacher requirements for further CPD in using the Bridge21 model confirm that 22% of teachers most needed CPD specialising in assistance with designing learning challenges, with 21% of teachers also requesting more CPD to learn and develop programming skills for use in teaching. Additional CPD was sought covering the planning of computing projects (19%), with 16% of teachers indicating that they required further assistance with preparing student assessment materials. Further analysis indicated that 11% of teachers were confident in using group work and that 7% were confident in using facilitation, which is reflected in the low number of responses requesting assistance with integrating group work and using facilitation in a computing context. The area which teachers expressed the least need for CPD was with assigning students roles and then organising students into teams (4%).

Deeper qualitative analysis (section 6.6.5) investigated teacher use of the Bridge21 model to teach computing and reports by teachers who observed an increase in student engagement through teaching computing using the Bridge21 model. Initial analysis exploring teaching computing using the Bridge21 model show that teachers used the Bridge21 model for encouraging students to take more responsibility for their learning and to give students more freedom in designing projects. The Bridge21 model empowered teachers in terms of providing a structure, which enabled students to take the lead in their learning, with teachers using worked examples to engage students in problem solving conversations. Teachers reported that they were using collaborative work more, with students taking the lead in designing their projects. Further analysis, exploring teacher observations of student behavioural change, captured examples, which demonstrate an increase in student engagement. The Bridge21 model provided a method to structure lessons, with teachers observing an increase in student engagement.

Teachers provided the following examples that they had used to increase student engagement in computing. Key findings include the gradual handover of responsibility for learning from the teachers to the student, with teachers putting in place activities, which encourage self-directed learning, supported by peers for particular tasks. Teachers also reported increasing the use of group work, with students encouraged to discuss, explore, and share computing ideas with peers. Worked examples were used to give students the freedom to work though tasks in groups, developing problem solving skills within a context. Furthermore, teachers reported that they enjoyed collaborating with students, engaging students in discussion on their projects, with students sharing expertise with the teacher, and teamers gaining insights into computing from students. A key finding that teachers reported was the energy and enthusiasm, which students applied to their work, with teachers facilitating discussions and students leading the computing design process.

Analysis of the use of the Bridge21 model confirms that teachers almost always used facilitation and collaborative project-based learning, with teacher's also incorporating problem solving methods in their computing lessons. Teachers perceived that the main barriers to the continued use of the Bridge21 model was a lack of a suitable technical infrastructure as well as a lack of time to practice developing computing and programming skills. Teachers asked for further CPD specialising in the design of learning challenges for students, with teachers also reporting that they were confident in their use of the Bridge21 model and that teachers observed that the Bridge21 model provided a structure which engaged students and supported them taking the lead in designing and creating computing projects.

Teachers also discussed limitations in using the model (section 6.7). The computing workshops were designed to support teachers with broad based computing knowledge and skills strengthen existing content knowledge and deepen expertise. Requests were made for introductory sessions to cover basic computing concepts and processes prior to attending the workshops. Further requests were made for more time spent practicing programming, in addition to contributing to collaborative project work. Finally, implementing the Bridge21 model was perceived to require investment from schools in technical infrastructure to support and sustain computing in schools.

Robins et al. (2003) suggests that there are four key themes that impact upon teaching, and the teachers' preparation for teaching computer science. First, Robins et al. (2003) suggests that it is essential that teachers are supported in developing the relevant content knowledge which they can apply in their teaching. Second, teachers need assistance with designing authentic or contextual learning experiences, which help students, develop a contextual view of computing. Thirdly, teachers need assistance in developing appropriate methodical approaches, which are compatible with teachers' beliefs but are also providing techniques for facilitating students collaborating and completing project-based learning in a computing context. Finally, it is essential that teachers are equipped with the confidence to engage with students, and assist with solving computing problems.

Responding to each of these points, the above results confirm that teachers gained in content knowledge and confidence to use a methodology designed to introduce computing into schools, which supported a collaborative, technology-mediated and project-based approach to student learning. Moreover, the CPD provided teachers with experience of computing tasks and activities, which they were encouraged to adapt for use in classrooms, with teachers completing programming tasks, which they could use as frameworks for teaching computing. The Bridge21 model workshops provided an integrated approach to teaching and learning computing, with teachers experiencing project-based learning as well as involved in completing shorter tasks designed to support divergent thinking and enhance problem solving skills in context. Teachers provided evidence reporting on using Bridge21 in a school context, with teachers teaching the CPD content and adapting Scratch, the Raspberry Pi, Python programming and computational thinking activities for use with their students. Finally evidence from teachers reporting on their experience of teaching computing in schools shows that teachers are using the Bridge21 model and that teachers are confident in their ability to engage with students, and confident in sharing their CS expertise.

Donaldson and Cutts (2018) suggest that teachers need access to structures which enable them to break down tasks into concepts. Using the Bridge21 model, teachers were able to break down the structure of computing lessons into group-based activities, enabling students to share their knowledge and expertise. Teachers observed that students take ownership, liaising with peers and completing projects, with teachers interjecting when assistance was required. Teachers also reported that the Bridge21 model provided a strategy to 'reduce teacher control' (Powerll, 1988).

Finally, the Bridge21 model provided teachers with a framework for introducing computing into schools. Teachers adopted the role of facilitator, stepping back from direct teaching to let students work collaboratively with their peers. The results provide evidence of teachers adopting a 'facilitator role' which Major et al. (2012) suggest help students to explore and analyse the links between concepts, especially where teachers ask questions and require students to articulate responses. A facilitator led approach can also help to address issues ranging from 'anxiety and fear to boredom' (Tom, 2015) especially where tasks are complex or where students are experiencing difficulty. Finally facilitation engages teachers in conversations with students, creating opportunities for students to impart knowledge and for teachers to share experience (Guzdial, 2016). The results concur with these findings, in that teachers reported that using facilitation in combination with other elements of the Bridge21 model to create engaging and 'fun' leaning experiences enjoyed by teachers.

To conclude, the Bridge21 model provided a methodology to engage students in computing. While the evidence on student engagement is based on teacher observations, the results indicate that students responded positively to the Bridge21 approach to learning computing. The examples provided by teachers confirm that students worked in teams to solve computing problems, with students consulting with peers when they encountered difficulties. Moreover, collaborative activities, gave students the opportunity to discuss and critique what they have learned (Crook, 2018). Teachers encouraged students to complete tasks in teams, which ensured that students had support networks available to assist with developing computing skills (Haduong & Brennan, 2018). Finally Teague and Roe (2008) suggest that tangible collaborative activities are essential in helping students develop the confidence and practical expertise to write computer programmes. The results confirm that the Bridge21 model and associated learning activities provided a context, which enabled students to work on projects, leveraging peer and teacher expertise when they needed assistance, rather than being directed by the teacher.

7.6 Research Limitations

The researcher acknowledges the following limitations with this research.

The first limitation relates to data collection. The data set exploring teacher experiences is smaller than the data set exploring teacher perceptions of the CPD. Thus, there is an imbalance between teachers reporting on their workshop experiences and teachers reporting on using the workshop content in schools.

A second limitation concerns the aggregation of results. CPD quantitative and quantitative data sets aggregate teacher perception and experience data, thus providing a summary view. Further analysis is needed to provide a more detailed view on a module-by-module basis, and year-by-year basis in relation to CPD performance.

The third limitation concerns the use of a hypothetico-deductive logical framework to drive data analysis. The researcher used a logical framework to map teacher responses to the research questions. There is an opportunity to reanalyse the data and explore further synergies between the data sets, given that the results presented in this thesis present one of many possible interpretations and are structured according to a particular framework.

In terms of the participants, teachers were invited to opt into the research process and thus there is bias within the data set, with responses weighted in favour of teachers who could be considered as more pre-disposed to the Bridge21 model. The teacher samples are opportunistic and response are therefore not representative of all teachers who took part in the Bridge21 CS CPD programme. The majority of data was gathered post-workshop and post CPD, therefore research claims that are based on reported increases in content knowledge, confidence, and changes in attitudes which may not be representative, but rather provide indicators of teacher perceptions of change.

In terms of the student experience, the evidence included in this thesis is self-reported with teachers reporting observations of student engagement with the Bridge21 model. While teacher accounts of the impact of the CPD on their students provides valuable insights into student engagement with Bridge21 learning experiences, further research is needed to explore the student experience, to understand what supports they need to further engage with computer science.

7.7 Research Conclusions

This research confirms that teacher perceptions of the CPD, involving the use of the Bridge21 model as a method for learning computing, were positive. Teachers were very satisfied with a collaborative, project-based and technology mediated approach to CPD. The reaction results show that a collaborative, project-based and technology mediated approach to teaching computing played a key role in encouraging teachers to develop strategies for overcoming challenges and problems in teaching computing. These results confirm that not only were teachers satisfied with a facilitator driven approach to learning computing but that teachers enjoyed the experience of learning with peers. Teachers enjoyed the experience of learning computing following a collaborative project-based approach, which enabled teachers to design projects and implement their ideas, supported by peers. The CPD also provided teachers with the opportunity to explore lesson planning and design, with the Bridge21 model providing a structure to bring computing to the classroom. The Bridge21 CPD workshops provided experiences, which not only enabled teachers to take the lead in their learning; they empowered teachers to lead the creative process.

Teachers also reported that the CPD met their expectations and that they intended using the Bridge21 model to introduce computing into schools. Teachers reported that they intended to use facilitation as a teaching method in the computing classroom. Teachers self-reported that the workshop experience had led to an increase in computing content knowledge, and that teachers were more confident in their ability to plan and orchestrate Bridge21 learning experiences in a computing context. Teachers also reported that they were empowered in their ability to use activities, which they had experienced in a CPD context, with their students. The CPD provided teachers with a process to structure computing lessons, which they could use in a classroom context.

A core component of CS CPD experiences should be to equip teachers with pedagogical content knowledge to teach the content. The results show that teachers used the Bridge21 model for teaching computing in schools, with facilitation, collaboration, and the use of social learning protocols used to encourage student engagement and nurture learner autonomy. This research also confirmed that teachers experience of teaching computing, following the Bridge21 model, led to an observed increase in student engagement in computing. Teachers observed an increase in student autonomy, with students taking the lead in computing projects, assisting peers, and working together to share computing knowledge and expertise.

This research also explored barriers to using the Bridge21 model revealing that, as found by Ott et al. (2018), teachers can experience isolation, a lack of support from peers, and resistance from management. In terms of barriers to using the Bridge21 model, teachers reported that the lack of a suitable technical infrastructure, time to practice programming as well as lack of access to formal training were barriers to the future use of the Bridge21 model. This research confirms that teachers need access to robust technical infrastructure to enable teachers to deliver computing learning experiences, which include tasks and activates which leverage computing resources. Teachers also reported using a blend of methods for teaching computing, involving students working in teams and completing projects as well as completing games, which Sentance and Csizmadia (2017b) advise as critucal to helping students gain access to computing concpets.

Finally, teachers reported that they most needed access to further CPD in designing learning challenges for use with students, with more time spend on learning and enhancing existing programming skills, and further assistance with planning and structuring project ideas and concepts for use by students. Teachers reported that they were very confident and empowered in using the Bridge21 model for teaching computing. Overall one could argue that the CPD did more than support "*teachers integrating technology into their pedagogy*" (Mishra & Koehler, 2006, p. 1017). Rather the results suggest that the Bridge21 CS CPD programme provided "*appropriate pedagogies for delivering a new subject, particularly in those aspects of computer science that relate to algorithms, programming and the development of computational thinking skills"* (Sentance & Csizmadia, 2017a, p. 469).

To conclude Sentance and Csizmadia (2017b) and Hazzan et al. (2014) and others (see Cabrera et al., 2018; Cutts et al., 2017; S. Davis, Ravitz, & Blazevski, 2018) argue that a new type of CPD is needed to prepare teachers with strategies and knowledge to support their students apply their thinking in a practical setting. In the examples provided by Sentance and Csizmadia (2017b) and Hazzan et al. (2014). CS CPD programmes need to be practical, and involve teachers in collaborative, project-based and technology-mediated work to help teachers develop the confidence and capacity to teach computing. CS CPD programmes should also be linked to accreditation, giving teachers the recognition of the time and resources spent in developing computing knowledge and expertise. Sentance and Csizmadia (2017b) suggest that CS CPD programmes which take an activity and project-based approach provide an opportunity to help teachers construct learning experiences, which help students, apply their learning. The Bridge21 CS CPD programme helped teachers develop in confidence and competence to teaching computing. Teachers reported that they were confident in their ability complete computing projects, and felt empowered to teach computing in schools. The next step is to assist teachers explore ways to further increase student engagement and support the sharing of expertise with others who are teaching CS today.

7.8 Research Contributions

There are three contributions from this research which are designed to respond to the pedagogical issues raised earlier by Sentance and Csizmadia (2017b) and Hazzan et al. (2014).

The first emerged through constructing an evidence base from research conducted with a self-selecting sample of *N* = 1,215 in-service teachers over a five year period. The evidence base included statistics and teacher accounts which explored the use of a collaborative, project-based and technology mediated approach to teaching used in a professional learning context. The results confirm that a workshop-based approach to professional learning plays an important role in equipping teachers with content knowledge, the technical skills and the confidence to teach computing. The CPD research findings confirm that teachers perceived that learning computing through a collaborative, project-based approach to professional development was a positive experience.

The second contribution emerged from research conducted with a self-selecting sub-sample of *N* = 64 teachers who completed a post CPD questionnaire. The results provide evidence of change in teacher attitudes and classroom practice, with teachers using the Bridge21 model elements of facilitation, collaborative learning, and contextualised learning tasks to engage students in computing and teachers confident and motivated to teach computing. These results provide insights into barriers (lack of technical infrastructure and further assistance with lesson planning) as well as the successes, with teachers self-reporting an observed increase in student learning and engagement though applying the Bridge21 model in teaching computing. Furthermore, these results provide examples of the collaborative strategies that teachers used with students, with teachers reflecting on the social strategies that students used to engage with computing.

Finally, a third contribution comes through adapting Kirkpatrick's (1994) work to create a theoretical framework to explore teacher reactions and perceptions of their learning and the impact that the CPD had on teaching computing in schools. The researcher adapted each of Kilpatrick's four levels as units of data analysis, which underpinned the research methodology that was used to explore teacher perceptions and experiences of the Bridge21 CS CPD programme.

7.9 Areas for Further Work

There are a number of areas of further work arise from this research.

The first, and most important, is the need to explore student's experiences of learning computing using the Bridge21 model. The preliminary findings report that teachers observed an increase in student motivation and engagement but this needs to be followed up with more in-depth research.

Second, there is a need for forming and sustaining a Computer Science Community of Practice in Ireland. Communities of practice play an important role in providing assistance and support to teachers embarking on lone implementations (Tytler et al., 2018). Delaney et al (2018) argue that communities which share practices play an essential role in helping teachers evaluate outcomes. Teachers support of organisations such as the Computers in Education Society of Ireland (http://www.cesi.ie/) and the Computer Science Teacher Association of Ireland (https://www.facebook.com/ComputerScienceTeachersAssocationIreland/) are strong advocates for supporting teachers plan, implement and disseminate details of CS lesson plans. These should be supported.

8 References

Abernethy, K., & Treu, K. (2009). Teaching computing soft skills: An experiential approach. *Journal of Computing Sciences in Colleges, 25*(2), 178-186. doi:dl.acm.org/citation.cfm?id=1629065

Acton, H. B. (1951). Comte's positivism and the science of society. *Philosophy, 26*(99), 291-310. doi:10.1017/S0031819100021719

Adelson, J. (2017). *Inventing adolescence*. New York, NY: Routledge.

Admiraal, W., Louws, M., Lockhorst, D., Paas, T., Buynsters, M., Cviko, A., . . . Kester, L. (2017). Teachers in school-based technology innovations: A typology of their beliefs on teaching and technology. *Computers & Education, 114*, 57-68. doi:10.1016/j.compedu.2017.06.013

Ajzen, I. (1991). The theory of planned behavior. *Organizational Behavior and Human Decision Processes, 50*(2), 179-211. doi:10.1016/0749-5978(91)90020-T

Alkin, M. C. (2012). *Evaluation roots: A wider perspective of theorists' views and influences* (2nd ed.). Thousand Oaks, CA: SAGE Publications, Inc.

Allen, M., Webb, A. W., & Matthews, C. E. (2016). Adaptive teaching in stem: Characteristics for effectiveness. *Theory Into Practice, 55*(3), 217-224. doi:10.1080/00405841.2016.1173994

Alliger, G. M., & Janak, E. A. (1989). Kirkpatrick's levels of training criteria: Thirty years later. *Personnel Psychology, 42*(2), 331-342. doi:10.1111/j.1744-6570.1989.tb00661.x

Aluko, F. R., & Shonubi, O. K. (2014). Going beyond kirkpatrick's training evaluation model: The role of workplace factors in distance learning transfer. *Africa Education Review, 11*(4), 638-657. doi:10.1080/18146627.2014.935007

Amory, A. (2018). Use of the collaboration-authentic learning-technology/tool mediation framework to address the theory–praxis gap. In T.-W. Chang, R. Huang, & Kinshuk (Eds.), *Authentic learning through advances in technologies. Lecture notes in educational technology* (pp. 61-73). Singapore, SG: Springer Nature Singapore Pte Ltd.

Andersen, P. B. (1992). Computer semiotics. *Scandinavian Journal of Information Systems, 4*(1), 3-30. doi:aisel.aisnet.org/sjis/vol4/iss1/1

Anderson, J. R. (1993). *Rules of the mind*. New York, NY: Lawrence Erlbaum Associates, Inc.

Angeli, C. (2013). Examining the effects of field dependence–independence on learners' problem-solving performance and interaction with a computer modeling tool: implications for the design of joint cognitive systems. *Computers & Education, 62*, 221-230. doi:10.1016/j.compedu.2012.11.002

Argyris, C., & Schön, D. A. (1974). *Theory in practice: Increasing professional effectiveness*. San Francisco, CA: Jossey-Bass.

Armstrong, S. L., Davis, H. S., & Paulson, E. J. (2011). The subjectivity problem: Improving triangulation approaches in metaphor analysis studies. *International Journal of Qualitative Methods, 10*(2), 151-163. doi:10.1177/160940691101000204

Arrowsmith, V., Lau-Walker, M., Norman, I., & Maben, J. (2016). Nurses' perceptions and experiences of work role transitions: A mixed methods systematic review of the literature. *Journal of Advanced Nursing, 72*(8), 1735-1750. doi:10.1111/jan.12912

Asghar, A., Ellington, R., Rice, E., Johnson, F., & Prime, G. M. (2012). Supporting stem education in secondary science contexts. *Interdisciplinary Journal of Problem-based Learning, 6*(2), 85-125. doi:10.7771/1541-5015.1349

Aslam, F., Adefila, A., & Bagiya, Y. (2018). Stem outreach activities: An approach to teachers' professional development. *Journal of Education for Teaching, 44*(1), 58-70. doi:10.1080/02607476.2018.1422618

Astbury, B., & Leeuw, F. L. (2010). Unpacking black boxes: Mechanisms and theory building in evaluation. *American Journal of Evaluation, 31*(3), 363-381. doi:10.1177/1098214010371972

ATECI. (2018). Association of teachers/education centres in ireland - cpd courses. *Web Page Accessed 10th May 2018.* Retrieved from https://www.ateci.ie/cpd-courses.html

Atherton, P. (2018). *50 ways to use technology enhanced learning in the classroom: Practical strategies for teaching*. London, UK: Learning Matters.

Atkins, S., Lewin, S., Smith, H., Engel, M., Fretheim, A., & Volmink, J. (2008). Conducting a meta-ethnography of qualitative literature: Lessons learnt. *BMC Medical Research Methodology, 8*(1), 1-10. doi:10.1186/1471-2288-8-21

Avidov-Ungar, O., & Forkosh-Baruch, A. (2018). Professional identity of teacher educators in the digital era in light of demands of pedagogical innovation. *Teaching and Teacher Education, 73*, 183-191. doi:10.1016/j.tate.2018.03.017

Avramides, K., Hunter, J., Oliver, M., & Luckin, R. (2015). A method for teacher inquiry in cross-curricular projects: Lessons from a case study. *British Journal of Educational Technology, 46*(2), 249-264. doi:10.1111/bjet.12233

Awad, N., & Barak, M. (2018). Pre-service science teachers learn a science, technology, engineering and mathematics (stem)-oriented program: The case of sound, waves and communication systems. *Eurasia Journal of Mathematics, Science and Technology Education, 14*(4), 1431-1451. doi:10.29333/ejmste/83680

B21. (2018). Computing workshops for teachers. *Web Page Accessed 4th July 2018.* Retrieved from http://bridge21.ie/cs/

Baggini, J., & Fosl, P. S. (2003). *The philopher's toolkit - a compendium of philisophical concepts and methods*. Malden, UK: Blackwell Publishing Ltd.

Bailey, K. D. (1994). *Typologies and taxonomies. An introduction to classification techniques*. Thousand Oaks, CA: SAGE Publications, Inc.

Baird, T. J., & Clark, L. E. (2018). The 'look-ahead' professional development model: A professional development model for implementing new curriculum with a focus on instructional strategies. *Professional Development in Education, 44*(3), 326-341. doi:10.1080/19415257.2017.1308424

Ball, S. J. (1993). Heuristic representation of the ethnographic research process *in* self-doubt and soft data: Social and technical trajectories in ethnographic field work. In M. Hammersley (Ed.), *Educational research - current issues* (pp. 34). London: The Open University Press.

Banks, F., & Barlex, D. (2014). *Teaching stem in the secondary school - helping teachers meet the challenge* Abingdon, Oxon, UK: Routledge.

Barak, M., & Assal, M. (2018). Robotics and stem learning: Students' achievements in assignments according to the p3 task taxonomy—practice, problem solving, and projects. *International Journal of Technology and Design Education, 28*(1), 121-144. doi:10.1007/s10798-016-9385-9

Barkway, P. (2001). Michael crotty and nursing phenomenology: Criticism or critique? *Nursing Inquiry, 8*(3), 191-195. doi:10.1046/j.1320-7881.2001.00104.x

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to k-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48-54. doi:10.1145/1929887.1929905

Baskerville, R. L. (1999). Investigating information systems with action research. *Communications of the Association for Information Systems, 2*(19), 1-32. doi:https://aisel.aisnet.org/cais/vol2/iss1/19/

Bates, R. (2004). A critical analysis of evaluation practice: The kirkpatrick model and the principle of beneficence. *Evaluation and Program Planning, 27*(3), 341-347. doi:10.1016/j.evalprogplan.2004.04.011

Bauer, C., Devitt, A., & Tangney, B. (2015). The alignment of cmc language learning methodologies with the bridge21 model of 21c learning. In F. Helm, L. Bradley, M. Guarda, & S. Thouësny (Eds.), *2015 eurocall conference; padova, italy; research-publishing.Net* (pp. 44-50). Padova, Italy: Research-Publishing.net.

Beecher, K. (2018). *Bad programming practices 101: Become a better coder by learning how (not) to program*. Berlin, GE: Apress.

Beijaard, D., Meijer, P. C., Morine-Dershimer, G., & Harm, T. (2005). *Teacher professional development in changing conditions*. Dordrecht, NL: Springer Netherlands.

Bell, T., & Newton, H. (2013). Unplugging computer science. In D. M. Kadijevich, C. Angeli, & C. Schulte (Eds.), *Improving computer science education* (pp. 66-82). New York, NY: Routledge.

Ben-Ari, M. (1998). Constructivism in computer science education. *29th Technical Symposium on Computer Science Education; Atlanta, Georgia, USA; ACM, 30*(1), 257-261. doi:10.1145/273133.274308

Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching, 20*(1), 45-73. doi:learntechlib.org/primary/p/8505

Ben-Ari, M. (2004). Situated learning in computer science education. *Computer Science Education, 14*(2), 85-100. doi:learntechlib.org/p/109461

Berger, P., & Luckmann, T. (1966). *The social construction of reality: A treatise in the sociology of knowledge*. London, UK: Penguin Books.

Bernthal, P. R. (1995). Evaluation that goes the distance. *Training & Development, 49*(9), 41-46. doi:https://eric.ed.gov/?id=EJ511215

Bevan, S., Brinkley, I., Cooper, C., & Bajorek, Z. (2018). *21st century workforces and workplaces: The challenges and opportunities for future work practices and labour markets*. London, UK: Bloomsbury Business

Bhanji, F., Gottesman, R., de Grave, W., Steinert, Y., & Winer, L. R. (2012). The retrospective pre–post: A practical method to evaluate learning from an educational program. *Academic Emergency Medicine, 19*(2), 189-194. doi:10.1111/j.1553-2712.2011.01270.x

Biggs, J. (1999). What the student does: Teaching for enhanced learning. *Higher Education Research & Development, 18*(1), 57-75. doi:10.1080/0729436990180105

Black, A. M., & Earnest, G. W. (2009). Measuring the outcomes of leadership development programs. *Journal of Leadership & Organizational Studies, 16*(2), 184-196. doi:10.1177/1548051809339193

Black, J., Brodie, J., Curzon, P., Myketiak, C., McOwan, P. W., & Meagher, L. R. (2013). Making computing interesting to school students: Teachers' perspectives. *18th Conference on Innovation and Technology in Computer Science Education; Canterbury, UK; ITCSE*, 255-260. doi:10.1145/2462476.2466519

Blömeke, S., & König, J. (2012). Does greater teacher knowledge lead to student orientation? The relationship between teacher knowledge and teacher beliefs. In J. König (Ed.), *Teachers' pedagogical beliefs. Definition and operationalisation - connections to knowlege and performance - development and change* (pp. 15-35). Münster, GE: Waxmann.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of education objectives* New York, USA: David McKay Company, Inc.

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational Psychologist, 26*(3-4), 369-398. doi:10.1080/00461520.1991.9653139

Blumer, H. (1969). *Symbolic interactionism: Perspective and method*. Berkeley, CA: University of California Press.

Bolam, R., & McMahon, A. (2005). Literature, definitions and models: Towards a conceptual map. In C. Day & J. Sachs (Eds.), *International handbook on the continuing professional development of teachers* (pp. 3-33). Maidenhead, UK: Open University Press.

Bonate, P. L. (2000). *Analysis of pretest-posttest designs*. Boca Raton, FL: Chapman and Hall/CRC.

Booth, A., Sutton, A., & Papaioannou, D. (2016). *Systematic approaches to a successful literature review*. London, UK: SAGE Publications Ltd.

Borko, H., Whitcomb, J., & Liston, D. (2009). Wicked problems and other thoughts on issues of technology and teacher learning. *Journal of Teacher Education, 60*(1), 3-7. doi:10.1177/0022487108328488

Bosse, Y., & Gerosa, M. A. (2017). Why is programming so difficult to learn? Patterns of difficulties related to programming learning mid-stage. *ACM Special Interest Group in Software Engineering Notes, 41*(6), 1-6. doi:10.1145/3011286.3011301

Boulton, H. (2017). Introducing digital technologies into secondary schools to develop literacy and engage disaffected learners: A case study from the uk. In A. Marcus-Quinn & T. Hourigan (Eds.), *Handbook on digital learning for k-12 schools* (pp. 31-44). Cham, CH: Springer International Publishing Switzerland.

Bouslama, F., Lansari, A., Al-Rawi, A. M., & Abonamah, A. A. (2003). A novel outcome-based educational model and its effect on student learning, curriculum development, and assessment. *Journal of Information Technology Education: Research, 2*, 203-214. doi:learntechlib.org/p/111473/

Boyas, E., Bryan, L. D., & Lee, T. (2012). Conditions affecting the usefulness of pre-and post-tests for assessment purposes. *Assessment & Evaluation in Higher Education, 37*(4), 427-437. doi:10.1080/02602938.2010.538665

Boylan, M., Coldwell, M., Maxwell, B., & Jordan, J. (2018). Rethinking models of professional learning as tools: A conceptual analysis to inform research and practice. *Professional Development in Education, 44*(1), 120-139. doi:10.1080/19415257.2017.1306789

Bray, A. (2015). Collaborative, contextual, and technology-mediated mathematics learning activities: Design heuristics and effects on student engagement. *Ph.D Thesis, Trinity College Dublin. Ireland*, 1-230. doi:http://hdl.handle.net/2262/77601

Bray, A., & Tangney, B. (2013). Mathematics, technology interventions and pedagogy-seeing the wood from the trees. *5th International Conference on Computer Supported Education; Aachen, Germany; ICCSE*, 57-63. doi:10.5220/0004349100570063

Bray, A., & Tangney, B. (2017). Technology usage in mathematics education research–a systematic review of recent trends. *Computers & Education, 114*, 255-273. doi:10.1016/j.compedu.2017.03.013

Brito, M. A., & de Sá-Soares, F. (2014). Assessment frequency in introductory computer programming disciplines. *Computers in Human Behavior, 30*, 623-628. doi:10.1016/j.chb.2013.07.044

Broad, M. L. (1997). *Transferring learning to the workplace*. Alexandria, VA: ASTD - American Society for Training and Development.

Brown, J. S. (2006). New learning environments for the 21st century: Exploring the edge. *Change: The Magazine of Higher Learning, 38*(5), 18-24. doi:10.3200/CHNG.38.5.18-24

Brown, K. G. (2005). An examination of the structure and nomological network of trainee reactions: A closer look at" smile sheets". *Journal of Applied Psychology, 90*(5), 991.

Brown, T., & Wyatt, J. (2010). Design thinking for social innovation. *Development Outreach, 12*(1), 29-43. doi:10.1596/1020-797X_12_1_29

Bryant, A. (2017). *Grounded theory and grounded theorizing: Pragmatism in research practice*. New York, NY: Oxford University Press.

Bujaki, M. L., & Richardson, A. J. (1997). A citation trail review of the uses of firm size in accounting research. *Journal of Accounting Literature, 16*, 1.

Bukoski, W. J., & Korotkin, A. L. (1976). Computing activities in secondary education. *Educational Technology, 16*(1), 9-23. doi:jstor.org/stable/44418053

Burch, P., & Heinrich, C. J. (2016). *Mixed methods for policy research and program evaluation*. Thousand Oaks, CA: SAGE Publications, Inc.

Burke, A., & Burke, S. P. (2018). Developing maker activity models across the curriculum. *29th Society for Information Technology & Teacher Education International Conference; Washington, DC, USA; SITE*, 1244-1246. doi:learntechlib.org/primary/p/182686/.

Bybee, R. W. (2010). Advancing stem education: A 2020 vision. *Technology and Engineering Teacher, 70*(1), 30-35. doi:eric.ed.gov/?id=EJ898909

Byrne, J. R. (2018). The bridge21 activity model. *Web Page Accessed 4th July 2018.* Retrieved from http://tft-project.eu/index.php/the-bridge21-model/

Byrne, J. R., Fisher, L., & Tangney, B. (2015a). Computer science teacher reactions towards raspberry pi continuing professional development (cpd) workshops using the bridge21 model. *10th International Conference on Computer Science & Education; Cambridge, UK; IEEE*, pp. 267-272. doi:10.1109/ICCSE.2015.7250254

Byrne, J. R., Fisher, L., & Tangney, B. (2015b). Empowering teachers to teach cs — exploring a social constructivist approach for cs cpd, using the bridge21 model. *2015 Frontiers in Education Conference; El Paso, Texas, USA; IEEE*, pp. 1-9. doi:10.1109/FIE.2015.7344030

Byrne, J. R., Fisher, L., & Tangney, B. (2016). A 21st century teaching and learning approach to computer science education: Teacher reactions. In S. Zvacek, M. T. Restivo, J. Uhomoibhi, & M. Helfert (Eds.), *Computer supported education* (pp. 523-540). Cham, CH: Springer International Publishing Switzerland.

Byrne, J. R., O'Sullivan, K., & Sullivan, K. (2017). An iot and wearable technology hackathon for promoting careers in computer science. *IEEE Transactions on Education, 60*(1), 50-58. doi:10.1109/TE.2016.2626252

Byrne, J. R., Sullivan, K., & O'Sullivan, K. (2018). Active learning of computer science using a hackathon-like pedagogical model. *2018 Constructionism; Vilnius, Lithuania; Research Council of Lithuania*, 138-150. doi:http://www.constructionism2018.fsf.vu.lt/file/repository/Proceeding_2018_Constructionism.pdf

Cabrera, K., Morreale, P., & Li, J. J. (2018). Computer science + education: An assessment of cs professional development. *Journal of Computing Sciences in Colleges, 33*(3), 141-147. doi:dl.acm.org/citation.cfm?id=3144719

Campbell-Kelly, M., & Aspray, W. (2018). *Computer: A history of the information machine* (3rd ed.). New York, NY: Routledge.

Caracelli, V. J., & Greene, J. C. (1993). Data analysis strategies for mixed-method evaluation designs. *Educational Evaluation and Policy Analysis, 15*(2), 195-207. doi:10.3102/01623737015002195

Carl, A. E. (2009). *Teacher empowerment through curriculum development: Theory into practice* (3rd ed.). Cape Town, SA: Juta and Company Ltd.

Caspersen, M. E. (2018). Teaching programming. In S. Sentance, E. Barendsen, & C. Schulte (Eds.), *Computer science education. Perspectives on teaching and learning in school* (pp. 109-131). London, UK: Bloomsbury Academic.

Caspersen, M. E., Gal-Ezer, J., Nardelli, E., Vahrenhold, J., & Westermeier, M. (2018). The cece report: Creating a map of informatics in european schools. *2018 Technical Symposium on Computer Science Education; Baltimore, MD, USA; ACM*, 916-917. doi:10.1145/3159450.3159633

Chalkin, S. (2003). The zone of proximal development in vygotsky's analysis of learning and instruction. In A. Kozulin, B. Gindis, V. S. Ageyev, & S. M. Miller (Eds.), *Vygotsky's educational theory in cultural context* (pp. 39-64). Cambridge, UK: Cambridge University Press.

Chamillard, A., & Braun, K. A. (2000). Evaluating programming ability in an introductory computer science course. *ACM Special Interest Group on Computer Science Education Bulletin, 32*(1), 212-216. doi:10.1145/331795.331857

Champion, R. (2003). Taking measure. The real measure of a professional development program's effectiveness lies in what participants learn. *Journal of Staff Development, 24*(1), 75-76.

Charmaz, K. (2000). Grounded theory: Objectivist and construcitivst methods. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of qualitative research* (pp. 509-535). Thousand Oaks, CA: Sage Publications, Inc.

Chelimsky, E., & Shadish, W. R. (1997). *Evaluation for the 21st century: A handbook*. Thousand Oaks, CA: SAGE Publications, Inc.

Chen, C. (2017). Science mapping: A systematic review of the literature. *Journal of Data and Information Science, 2*(2), 1–40. doi:https://doi.org/10.1515/jdis-2017-0006

Cheng, E. W., & Hampson, I. (2008). Transfer of training: A review and new insights. *International Journal of Management Reviews, 10*(4), 327-341. doi:10.1111/j.1468-2370.2007.00230.x

Clark, R. E. (1985). Confounding in educational computing research. *Journal of Educational Computing Research, 1*(2), 137-148. doi:10.2190/HC3L-G6YD-BAK9-EQB5

Clarke, A., & Dawson, R. (1999). *Evaluation research: An introduction to principles, methods and practice*. London, UK: SAGE Publications Ltd.

Clarke, D., & Hollingsworth, H. (2002). Elaborating a model of teacher professional growth. *Teaching and Teacher Education, 18*(8), 947-967. doi:10.1016/S0742-051X(02)00053-7

Clear, T. (2004). Critical inquiry within computer science education In S. Fincher & M. Petre (Eds.), *Computer science education research* London, UK: Taylor & Francis Group plc.

Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology, 76*(6), 1051-1058. doi:10.1037/0022-0663.76.6.1051

Cohen, L., Manion, L., & Morrison, K. (2000). *Research methods in education*. London, UK: RoutledgeFalmer.

Coldwell, M., & Simkins, T. (2011). Level models of continuing professional development evaluation: A grounded review and critique. *Professional Development in Education, 37*(1), 143-157. doi:10.1080/19415257.2010.495497

Comi, S. L., Argentin, G., Gui, M., Origo, F., & Pagani, L. (2017). Is it the way they use it? Teachers, ict and student achievement. *Economics of Education Review, 56*, 24-39. doi:10.1016/j.econedurev.2016.11.007

Compton, M., & Almpanis, T. (2018). One size doesn't fit all: Rethinking approaches to continuing professional development in technology enhanced learning. *Compass: Journal of Learning and Teaching, 11*(1), 1-12. doi:10.21100/compass.v11i1.708

Condamines, T. (2011). Gathering practice sharing and problem solving on a single platform for teacher training: A collaborative platform model for teacher professional development. *11th International Conference on Advanced Learning Technologies; Athens, GA; IEEE*, 290-294. doi:10.1109/ICALT.2011.90

Conneely, C. (2018). The bridge21 model of technology-mediated, team-based, 21st century learning in post-primary classrooms in ireland: Evolution, experiences and evaluation *Ph.D Thesis, Trinity College Dublin. Ireland*, 1-370. doi:http://hdl.handle.net/2262/83430

Conneely, C., Girvan, C., & Tangney, B. (2012). *An exploration into the adaption of the bridge21 model for 21st century learning in irish classrooms: Case study report for the ncca. Bridge21, tcd, dublin*. Retrieved from http://www.bridge21.ie/wp-content/uploads/2013/01/Bridge21_Case_Study_Report_2011-12_for_NCCA_Final1.pdf

Coombs, J. R. (1994). Doing justice to rationality. *Educational Theory, 44*(1), 119-133. doi:10.1111/j.1741-5446.1994.00119.x

Cooper, H. M. (2010). *The handbook of research synthesis and meta-analysis. A step-by-step approach*. Thousand Oaks, CA: SAGE Publications, Inc.

Coorey, G. M., Neubeck, L., Mulley, J., & Redfern, J. (2018). Effectiveness, acceptability and usefulness of mobile applications for cardiovascular disease self-management: Systematic review with meta-synthesis of quantitative and qualitative data. *European Journal of Preventive Cardiology, 25*(5), 505-521. doi:10.1177/2047487317750913

Cordingley, P., Bell, M., Evans, D., & Firth, A. (2005). *The impact of collaborative cpd on classroom teaching and learning - what do teacher impact data tell us about collaborative cpd*: Report - Evidence for Policy and Practice Information and Co-ordinating Centre, Social Science Research Unit, University of London, http://wsassets.s3.amazonaws.com/ws/nso/pdf/09598003e49523abff794962e2752c81.pdf.

Cousins, C., & Brooke, J. (2018). Mars rovers drive stem teaching in schools. *Astronomy & Geophysics, 59*(1), 1.34-31.35. doi:10.1093/astrogeo/aty029

Creswell, J. W. (1998). *Qualitative inquiry and research design: Choosing among five traditions*. Thousand Oaks, CA: SAGE Publications, Inc.

Creswell, J. W. (2005). *Educational research - planning, conducting, and evaluating quantatative and qualatative research* (2nd ed.). Upper Saddle River, NJ: Pearson Education, Inc.

Creswell, J. W., & Clark, V. P. (2011). *Designing and conducting mixed methods research* (2nd ed.). Thousand Oaks, CA: Sage Publications, Inc.

CRITE. (2013). Trinity & google certified course in 21c stem teaching. In *Extract from an Internal CRITE Research Memo* (pp. 1). Dublin, Ireland: TCD.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika, 16*(3), 297-334. doi:10.1007/bf02310555

Cronbach, L. J. (1980). *Toward reform of program evaluation*. San Francisco, CA: Jossey-Bass Publishers.

Cronbach, L. J. (1983a). Course improvement through evaluation. In G. F. Madaus, M. Scriven, & D. L. Stufflebeam (Eds.), *Evaluation models: Viewpoints on educational and human services evaluation* (pp. 101-117). Boston, MA: Kluwer-Nijhoff Publishing.

Cronbach, L. J. (1983b). Ninety-five theses for reforming program evaluation. In G. F. Madaus, M. Scriven, & D. L. Stufflebeam (Eds.), *Evaluation models: Viewpoints on educational and human services evaluation* (pp. 405-413). Boston, MA: Kluwer-Nijhoff Publishing.

Cronbach, L. J., Rajaratnam, N., & Gleser, G. C. (1963). Theory of generalizability: A liberalization of reliability theory. *British Journal of Mathematical and Statistical Psychology, 16*(2), 137-163. doi:10.1111/j.2044-8317.1963.tb00206.x

Crook, C. (2018). *Computers and the collaborative experience of learning* London, UK: Routledge.

Crotty, M. (1998). *The foundations of social research: Meaning and perspective in the research process*. London, UK: SAGE Publications Ltd.

Cruickshank, J. (2007). The usefulness of fallibilism in post-positivist philosophy: A popperian critique of critical realism. *Philosophy of the Social Sciences, 37*(3), 263-288. doi:10.1177/0048393107303759

Curzon, P. (2007). Serious fun in computer science. *ACM Special Interest Group on Computer Science Education Bulletin, 39*(3), 1. doi:10.1145/1269900.1268785

Cutts, Q., Brown, M., Kemp, L., & Matheson, C. (2007). Enthusing and informing potential computer science students and their teachers. *ACM Special Interest Group on Computer Science Education Bulletin, 39*(3), 196-200. doi:10.1145/1269900.1268842

Cutts, Q., Patitsas, E., Cole, E., Donaldson, P., Alshaigy, B., Gutica, M., . . . Riedesel, C. (2018). Early developmental activities and computing proficiency. *2018 ITiCSE Conference on Working Group Reports; Bologna, Italy; ACM*, 140-157. doi:10.1145/3174781.3174789

Cutts, Q., Robertson, J., Donaldson, P., & O'Donnell, L. (2017). An evaluation of a professional learning network for computer science teachers. *Computer Science Education, 27*(1), 30-53. doi:10.1080/08993408.2017.1315958

Dagdilelis, V., Satratzemi, M., & Evangelidis, G. (2004). Introducing secondary education students to algorithms and programming. *Education and Information Technologies, 9*(2), 159-173. doi:10.1023/B:EAIT.0000027928.94039.7b

Dahler-Larsen, P. (2001). From programme theory to constructivism: On tragic, magic and competing programmes. *Evaluation, 7*(3), 331-349. doi:10.1177/13563890122209711

Darling-Hammond, L. (2017). Teacher education around the world: What can we learn from international practice? *European Journal of Teacher Education, 40*(3), 291-309. doi:10.1080/02619768.2017.1315399

Darling-Hammond, L., & Bransford, J. (2005). *Preparing teachers for a changing world: What teachers should learn and be able to do*. San Francisco, CA: John Wiley & Sons, Inc.

Darling-Hammond, L., & McLaughlin, M. W. (1995). Policies that support professional development in an era of reform. *Phi Delta Kappan, 76*(8), 597-604. doi:eric.ed.gov/?id=EJ501259

Davidson-Shivers, G. V., Rasmussen, K. L., & Lowenthal, P. R. (2018). *Web-based learning - design, implementation and evaluation*. Cham, CH: Springer International Publishing Switzerland.

Davidson, D. (2001). *Subjective, intersubjective, objective: Philosophical essays*. Oxford, UK: Clarendon Press.

Davies, H., Nutley, S., & Walter, I. (2008). Why 'knowledge transfer'is misconceived for applied social research. *Journal of Health Services Research & Policy, 13*(3), 188-190. doi:10.1258/jhsrp.2008.008055

Davis, N., Desforges, C., Jessel, J., Somekh, B., Taylor, C., & Vaughan, G. (1997). Can quality in learning be ehanced through the use of it? In B. Somekh & N. Davis (Eds.), *Using* *information technology effectively in teaching and learning: Studies in pre-service and in-service teacher education* (pp. 14-27). Abingdon, Oxon, UK: Routledge.

Davis, S., Garcia, O., & Stephenson, C. (2018). Working together to provide equitable cs k-12 educator professional development at scale. *2018 Technical Symposium on Computer Science Education; Baltimore, MD, USA; ACM*, 1066-1066. doi:10.1145/3159450.3162193

Davis, S., Ravitz, J., & Blazevski, J. (2018). Evaluating computer science professional development models and educator outcomes to ensure equity. *2018 Research on Equity and Sustained Participation in Engineering, Computing, and Technology Conference; Baltimore, MD, USA; IEEE*, 1-4. doi:10.1109/RESPECT.2018.8491716

Dawes, J. (2008). Do data characteristics change according to the number of scale points used? An experiment using 5-point, 7-point and 10-point scales. *International Journal of Market Research, 50*(1), 61-104. doi:10.1177/147078530805000106

Day, C., & Leitch, R. (2001). Teachers' and teacher educators' lives: The role of emotion. *Teaching and Teacher Education, 17*(4), 403-415. doi:https://doi.org/10.1016/S0742-051X(01)00003-8

de Lima, M. N. M., Laranja, D. C., Bromberg, E., Roesler, R., & Schröder, N. (2005). Pre-or post-training administration of the nmda receptor blocker mk-801 impairs object recognition memory in rats. *Behavioural Brain Research, 156*(1), 139-143. doi:10.1016/j.bbr.2004.05.016

De Miranda, M. A. (2018). Pedagogical content knowledge for technology education. In M. J. De Vries (Ed.), *Handbook of technology education* (pp. 685-698). Cham, Switzerland: Springer International Publishing AG

de Paor, C., & Murphy, T. R. N. (2018). Teachers' views on research as a model of cpd: Implications for policy. *European Journal of Teacher Education, 41*(2), 169-186. doi:10.1080/02619768.2017.1416086

De Villiers, M. R. (2005). Interpretive research models for informatics: Action research, grounded theory, and the family of design-and development research. *Alternation, 12*(2), 10-52. doi:http://www.ingentaconnect.com/content/sabinet/alt/2005/00000012/00000002/art00008 de Vries, S., Jansen, E. P., & van de Grift, W. J. (2013). Profiling teachers' continuing professional development and the relation with their beliefs about learning and teaching. *Teaching and Teacher Education, 33*, 78-89. doi:10.1016/j.tate.2013.02.006

De Vrieze, J. (2017). Bruno latour, a veteran of the ‘science wars,’has a new mission. *Science*, 1-8. doi:10.1126/science.aaq1805

Decker, A., McGill, M. M., Ravitz, J., Snow, E., & Zarch, R. (2018). Connecting evaluation and computing education research: Why is it so important? *2018 Technical Symposium on Computer Science Education; Baltimore, MD, USA; ACM*, 818-819. doi:10.1145/3159450.3159642

Dede, C. (2010). Comparing frameworks for 21st century skills. In J. A. Bellanca & R. Brandt (Eds.), *21st century skills: Rethinking how students learn* (pp. 51-77). Bloomington, IN: Solution Tree Press.

Delaney, K., O’Keeffe, M., & Fragou, O. (2018). A design framework for interdisciplinary communities of practice towards stem learning in 2nd level education. In M. E. Auer, D. Guralnick, & I. Simonics (Eds.), *Teaching and learning in a digital world. Proceedings of the 20th international conference on interactive collaborative learning – volume 1* (pp. 739-750). Cham, CH: Springer International Publishing Switzerland.

DeMaris, A., & Selman, S. H. (2013). *Converting data into evidence. A statistics primer for the medical pracitioner*. New York, NY: Springer Science + Business Media.

Dennis, A. K. (2013). *Raspberry pi home automation with arduino*. Birmingham, UK: Packt Publishing Ltd.

Denzin, N. K. (1989). *Interpretive interactionism*. Thousand Oaks, CA: SAGE Publications, Inc.

Denzin, N. K. (1997). *Interpretive ethnography: Ethnographic practices for the 21st century*. Thousand Oaks, CA: SAGE Publications Inc.

DES. (2018a). Action plan for education 2016-2019. *An Roinn Oideachais agus Scileanna | Department of Education & Skills, Web Page Accessed 10th July 2018.* Retrieved from https://www.education.ie/en/Publications/Corporate-Reports/Strategy-Statement/Department-of-Education-and-Skills-Strategy-Statement-2016-2019.pdf

DES. (2018b). Minister bruton announces leaving certificate computer science subject - 40 schools will be first to start studying leaving certificate computer science from this september *An Roinn Oideachais agus Scileanna | Department of Education & Skills, Web Page Accessed 26th May 2018.* Retrieved from https://www.education.ie/en/Press-Events/Press-Releases/2018-press-releases/PR2018-01-04.html

DES. (2018c). Post primary education. *An Roinn Oideachais agus Scileanna | Department of Education & Skills, Web Page Accessed 2nd July 2018.* Retrieved from https://www.education.ie/en/The-Education-System/Post-Primary/

Desimone, L. M., Porter, A. C., Garet, M. S., Yoon, K. S., & Birman, B. F. (2002). Effects of professional development on teachers' instruction: Results from a three-year longitudinal study. *Educational Evaluation and Policy Analysis, 24*(2), 81-112. doi:10.3102/01623737024002081

Devenyi, G., Emonet, R., Harris, R., Hertweck, K., Irving, D., & Milligan, I. (2018). Ten simple rules for collaborative lesson development. *PLoS Computatinal Biology, 14*(3), 1-8. doi:10.1371/journal.pcbi.1005963

Dick, W. (1978). The educational psychologist as instructional designer. *Contemporary Educational Psychology, 3*(3), 265-271. doi:10.1016/0361-476X(78)90032-2

Dick, W., & Johnson, R. (2007). Evaluation in instructional design: The impact of kirkpatrick's four-level model. In R. A. Reiser & J. V. Dempsey (Eds.), *Trends and issues in instructional design and technology* (2nd ed., pp. 94-103). Upper Saddle River, NJ: Pearson Education, Inc.

Dikilitas, K. (2015). *Innovative professional development methods and strategies for stem education*. Hershey, USA: IGI Global

Donaldson, P., & Cutts, Q. (2018). Flexible low-cost activities to develop novice code comprehension skills in schools. *13th Workshop in Primary and Secondary Computing Education; Potsdam, Germany; ACM*, 1-4. doi:10.1145/3265757.3265776

Dorling, M. (2016). Effective computing cpd. *Seced*(22), 13-13. doi:10.12968/sece.2016.22.13

Dort, J., Trickey, A., Paige, J., Schwarz, E., Cecil, T., Coleman, M., & Dunkin, B. (2018). All in: Expansion of the acquisition of data for outcomes and procedure transfer (adopt) program to an entire sages annual meeting hands-on hernia course. *Surgical Endoscopy, 32*(11), 1-7. doi:10.1007/s00464-018-6196-y

Drennan, J., & Hyde, A. (2008). Controlling response shift bias: The use of the retrospective pre-test design in the evaluation of a master's programme. *Assessment & Evaluation in Higher Education, 33*(6), 699-709. doi:10.1080/02602930701773026

Du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research, 2*(1), 57-73. doi:10.2190/3LFX-9RRF-67T8-UVK9

Dudley, P. (2014). *Lesson study: Professional learning for our time*. Abingdon, Oxon, UK: Routledge.

Duval, E., Sharples, M., & Sutherland, R. (2017). *Research themes in technology enhanced learning* Cham, CH: Springer International Publishing Switzerland.

Dybå, T., Arisholm, E., Sjøberg, D. I., Hannay, J. E., & Shull, F. (2007). Are two heads better than one? On the effectiveness of pair programming. *IEEE Software, 24*(6), 12-15. doi:10.1109/MS.2007.158

Ebert-May, D., Derting, T. L., Henkel, T. P., Maher, J. M., Momsen, J. L., Arnold, B., & Passmore, H. A. (2015). Breaking the cycle: Future faculty begin teaching with learner-centered strategies after professional development. *CBE-Life Sciences Education, 14*(2), 1-30. doi:10.1187/cbe.14-12-0222

EC. (2017). Coding - the 21st century skill. European comission. *Web Page Accessed 10th June 2017.* Retrieved from https://ec.europa.eu/digital-single-market/en/coding-21st-century-skill

EC. (2018). Eu-funded fet projects on computer science. European comission. *Web Page Accessed 11th July 2018.* Retrieved from https://ec.europa.eu/digital-single-market/en/programme-and-projects/eu-funded-fet-projects-computer-science

Eco, U. (1984). *Semiotics and the philosphy of language* London, UK: The Macmillan Press Ltd.

EES. (2018). European evaluation society. *Web Page Accessed 21st June 2018.* Retrieved from https://www.europeanevaluation.org/

Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management Review, 14*(4), 532-550. doi:10.5465/amr.1989.4308385

Ekmekci, A., Parr, R., & Fisher, A. (2018). Results from rice university weteach_cs: A computer science teaching collaborative serving teachers with different needs through variety of pathways. *2018 Society for Information Technology & Teacher Education International Conference; Washington, D.C.,USA; AACE*, 2025-2030. doi:http://www.learntechlib.org/primary/p/182806/

Ely, M., Anzul, M., Friedman, T., Garner, D., & McCormack-Steinmetz, A. (1991). *Doing qualitative research: Circles within circles*. London, UK: The Falmer Press

Emerson, R. M. (1995). *Writing ethnographic fieldnotes*. Chicago, ILL: University Of Chicago Press.

Esteve-Faubel, J.-M., Martin, T. J., & Junda, M. E. (2018). Cross-curricular teaching going forward: A view from “strange fruit”. *International Journal of Education & the Arts, 19*(4), 1-27. doi:10.18113/P8ijea1904

Faerman, S. R., & Ban, C. (1993). Trainee satisfaction and training impact: Issues in training evaluation. *Public Productivity & Management Review, 16*(3), 299-314. doi:10.2307/3380872

Feldman, A., Altrichter, H., Posch, P., & Somekh, B. (2018). *Teachers investigate their work: An introduction to action research across the professions* (3rd ed.). Abingdon, Oxon, UK: Routledge.

Fereday, J., & Muir-Cochrane, E. (2006). Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International Journal of Qualitative Methods, 5*(1), 80-92. doi:10.1177/160940690600500107

Fetterman, D. M. (1989). *Ethnography: Step by step*. Thousand Oaks, CA: SAGE Publications, Inc.

Fetterman, D. M. (2001). *Foundations of empowerment evaluation: Step by step*. Thousand Oaks, CA: SAGE Publications, Inc.

Fetterman, D. M. (2005). *Empowerment evaluation principles in practice*. New York, NY: The Guilford Press.

Fetterman, D. M., Kaftarian, S. J., & Wandersman, A. (1996). *Empowerment evaluation: Knowledge and tools for self-assessment and accountability*. Thousand Oaks, CA: SAGE Publications, Inc.

Fetterman, D. M., Kaftarian, S. J., & Wandersman, A. (2015). *Empowerment evaluation: Knowledge and tools for self-assessment, evaluation capacity building, and accountability*. Thousand Oaks, CA: SAGE Publications, Inc.

Fischer, C., Fishman, B., Dede, C., Eisenkraft, A., Frumin, K., Foster, B., . . . McCoy, A. (2018). Investigating relationships between school context, teacher professional development, teaching practices, and student achievement in response to a nationwide science reform. *Teaching and Teacher Education, 72*, 107-121. doi:10.1016/j.tate.2018.02.011

Fish, S. (1980). *Is there a text in this class? The authorities of interpretive communities*. Cambridge, MA: Harvard University Press.

Fisher, L., Byrne, J. R., & Tangney, B. (2015a). Exploring teacher reactions towards a 21st century teaching and learning approach to continuing professional development programme in computer science. *7th International Conference on Computer Supported Education; Lisbon, Portugal; CSEDU, 2*, 353-362 doi:10.5220/0005432603530362

Fisher, L., Byrne, J. R., & Tangney, B. (2015b). Using an ethnographic approach to educational evaluation to understand computer science teacher continuing professional development needs *37th Oxford Conference on Ethnography and Education; Oxford, UK; EEO*, 1-25. doi:http://www.ethnographyandeducation.org/conferences

Fisher, L., Byrne, J. R., & Tangney, B. (2016). Teacher experiences of learning computing using a 21st century model of computer science continuing professional development. *8th International Conference on Computer Supported Education; Rome, Italy; CSEDU, 2*, 273-280. doi:10.5220/0005906702730280

Fisher, L., O'Sullivan, K., Tangney, B., & Byrne, J. R. (2017). A social constructivist approach to computer science teacher professional development – the bridge21 cpd approach. *2017 World Conference on Computers in Education; Dublin, Ireland; WCCE*, 128. doi:https://www.ics.ie/images/wcce_programme%20final_w.pdf

Flatland, R., Goldstein, I., Egan, M. L., Vandenberg, S., Fryling, M., & Small, S. (2018). Connecting colleges/universities and local high schools: A new model for high school cs teacher development. *49th Technical Symposium on Computer Science Education; Baltimore, MD, USA; ACM*, 958-963. doi:10.1145/3159450.3159570

Forbes, N., & Messina, P. (2002). Computer science today in the european union. *Computing in Science & Engineering, 4*(1), 10-14. doi:10.1109/5992.976431

Gagné, R. M. (1970). *The conditions of learning* (2nd ed.). Oxford, UK: Holt, Rinehart and Winston.

Galili, I. (2018). Scientific knowledge as a culture: A paradigm for meaningful teaching and learning of science. In M. Matthews (Ed.), *History, philosophy and science teaching* (pp. 203-233). Cham, CH: Springer International Publishing AG.

Garet, M. S., Porter, A. C., Desimone, L., Birman, B. F., & Yoon, K. S. (2001). What makes professional development effective? Results from a national sample of teachers. *American Educational Research Journal, 38*(4), 915-945. doi:10.3102/00028312038004915

Geertz, C. (1973). *The interpretation of cultures: Selected essays* (2000 ed.). New York, NY: Basic Books.

Geertz, C. (1988). *Works and lives: The anthropologist as author*. Stanford, CA: Stanford University Press.

Geuss, R. (1981). *The idea of a critical theory: Habermas and the frankfurt school*. Cambridge, UK: Cambridge University Press.

Giddings, L. S., & Grant, B. M. (2006). Mixed methods research for the novice researcher. *Contemporary Nurse, 23*(1), 3-11. doi:10.5172/conu.2006.23.1.3

Gil-Flores, J., Rodríguez-Santero, J., & Torres-Gordillo, J.-J. (2017). Factors that explain the use of ict in secondary-education classrooms: The role of teacher characteristics and school infrastructure. *Computers in Human Behavior, 68*, 441-449. doi:10.1016/j.chb.2016.11.057

Girvan, C., Conneely, C., & Tangney, B. (2016). Extending experiential learning in teacher professional development. *Teaching and Teacher Education, 58*, 129-139. doi:10.1016/j.tate.2016.04.009

Glaser, B. G., & Strauss, A. L. (2012). *The discovery of grounded theory: Strategies for qualitative research*. New Brunswick, NJ: Aldine Transaction.

Glazer, E. M., & Hannafin, M. J. (2006). The collaborative apprenticeship model: Situated professional development within school settings. *Teaching and Teacher Education, 22*(2), 179-193. doi:10.1016/j.tate.2005.09.004

Gluga, R., Kay, J., Lister, R., & Teague, D. (2012). On the reliability of classifying programming tasks using a neo-piagetian theory of cognitive development. *9th Annual International Conference on International Computing Education Research; Auckland, New Zealand; ACM*, 31-38. doi:10.1145/2361276.2361284

Goetz, J. P., & LeCompte, M. D. (1981). Ethnographic research and the problem of data reduction. *Anthropology & Education Quarterly, 12*(1), 51-70. doi:10.1525/aeq.1981.12.1.05x1283i

Goode, J., Flapan, J., & Margolis, J. (2018). Computer science for all. In W. G. Tierney, Z. B. Corwin, & A. Oscher (Eds.), *Diversifying digital learning: Online literacy and educational opportunity* (pp. 45-66). Baltimore, Maryland: John Hopkins University Press

Grant, M. J., & Booth, A. (2009). A typology of reviews: An analysis of 14 review types and associated methodologies. Health information & libraries journal. *Health Information & Libraries Journal, 26*, 91-108. doi:10.1111/j.1471-1842.2009.00848.x

Greene, J. C. (2012). The educative evaluator: An interpretation of lee. J. Cronbach's vision of evaluation. In M. C. Alkin (Ed.), *Evaluation roots: A wider perspective of theorists' views and influences* (2nd ed., pp. 97-106). Thousand Oaks, CA: SAGE Publications, Inc.

Gretter, S., & Yadav, A. (2016). Computational thinking and media & information literacy: An integrated approach to teaching twenty-first century skills. *TechTrends, 60*(5), 510-516. doi:10.1007/s11528-016-0098-4

Grover, S., Basu, S., & Schank, P. (2018). What we can learn about student learning from open-ended programming projects in middle school computer science. *2018 Technical Symposium on Computer Science Education; Baltimore, MD, USA; ACM*, 999-1004. doi:10.1145/3159450.3159522

Grover, S., & Pea, R. (2013). Computational thinking in k–12: A review of the state of the field. *Educational Researcher, 42*(1), 38-43. doi:10.3102/0013189X12463051

Grover, S., & Pea, R. (2018). Computational thinking: A competency whos time has come. In S. Sentance, E. Barendsen, & C. Schulte (Eds.), *Computer science education: Perspectives on teaching and learning in school* (pp. 19-39). London, UK: Bloomsbury Academic.

Guba, E. G., & Lincoln, Y. S. (1981). *Effective evaluation: Improving the usefulness of evaluation results through responsive and naturalistic approaches*. San Francisco, CA, US: Jossey-Bass.

Guba, E. G., & Lincoln, Y. S. (2000). *Handbook of qualitative research.* Thousand Oaks, CA: SAGE Publications, Inc.

Guskey, T. (2000). *Evaluating professional development*. Thousand Oaks, CA: Corwin Press, Inc.

Guskey, T. (2002). Does it make a difference? Evaluating professional development. *Educational Leadership, 59*(6), 45-51. doi:eric.ed.gov/?id=EJ640979

Guskey, T. (2016). Gauge impact with 5 levels of data. *Journal of Staff Development, 37*(1), 32-37. doi:http://learningforward.org/docs/default-source/jsd-february-2016/gauge-impact-with-5-levels-of-data-feb16.pdf

Guyatt, G., Walter, S., & Norman, G. (1987). Measuring change over time: Assessing the usefulness of evaluative instruments. *Journal of Chronic Diseases, 40*(2), 171-178. doi:10.1016/0021-9681(87)90069-5

Guzdial, M. (2014). Preparing teachers is different than preparing software developers. *9th Workshop in Primary and Secondary Computing Education; Berlin, Germany; ACM*, 1. doi:10.1145/2670757.2676866

Guzdial, M. (2016). *Learner-centered design of computing education: Research on computing for everyone*. San Rafael, CA: Morgan & Claypool Publishers

Haduong, P., & Brennan, K. (2018). Getting unstuck: New resources for teaching debugging strategies in scratch. *2018 Technical Symposium on Computer Science Education; Baltimore, MD, USA; ACM*, 1092-1092. doi:10.1145/3159450.3162248

Halfacree, G., & Upton, E. (2012). *Raspberry pi user guide*. Hoboken, NJ: John Wiley & Sons, Inc.

Hamilton, M. (2018). Pedagogical transitions among science teachers: How does context intersect with teacher beliefs? *Teachers and Teaching, 24*(2), 151-165. doi:10.1080/13540602.2017.1367658

Hamlen, K., Sridhar, N., Bievenue, L., Jackson, D. K., & Lalwani, A. (2018). Effects of teacher training in a computer science principles curriculum on teacher and student skills, confidence, and beliefs. *2018 Technical Symposium on Computer Science Education; Baltimore, MD, USA; ACM*, 741-746. doi:10.1145/3159450.3159496

Hammersley, M. (2001). *What's wrong with ethnography?* London: Routledge.

Hammersley, M., & Atkinson, P. (1995). *Ethnography: Principles in practice*. London: Routledge.

Hammersley, M., & Atkinson, P. (2006). *Ethnography: Principles in practice* (2nd Edition ed.). Abingdon, Oxon, UK: Routledge.

Harden, R. M. (2002). Learning outcomes and instructional objectives: Is there a difference? *Medical teacher, 24*(2), 151-155. doi:10.1080/0142159022020687

Hardy, I. (2012). *The politics of teacher professional development: Policy, research and practice*. London, UK: Taylor & Francis Group plc.

Hargreaves, A., & Dawe, R. (1990). Paths of professional development: Contrived collegiality, collaborative culture, and the case of peer coaching. *Teaching and Teacher Education, 6*(3), 227-241. doi:10.1016/0742-051X(90)90015-W

Hargreaves, A., & O'Connor, M. T. (2018). *Collaborative professionalism: When teaching together means learning for all*. Thousand Oaks, CA: Corwin

Harland, J., & Kinder, K. (2014). Teachers' continuing professional development: Framing a model of outcomes. *Professional Development in Education, 40*(4), 669-682. doi:10.1080/19415257.2014.952094

Harris, J., Phillips, M., Koehler, M., & Rosenberg, J. (2017). Tpck/tpack research and development: Past, present, and future directions. *Australasian Journal of Educational Technology, 33*(3), i-viii. doi:10.14742/ajet.3907

Harris, W. V. (2000). *Interpretive acts: In search of meaning*. Oxford, UK: Clarendon Press.

Hassard, J. (1993). *Sociology and organization theory: Positivism, paradigms and postmodernity*. Cambridge, UK: Cambridge University Press.

Hatlevik, I. K. R., & Hatlevik, O. E. (2018). Students' evaluation of digital information: The role teachers play and factors that influence variability in teacher behaviour. *Computers in Human Behavior, 83*, 56-63. doi:10.1016/j.chb.2018.01.022

Hazzan, O., Heyd-Metzuyanim, E., Even-Zahav, A., Tal, T., & Dori, Y. J. (2018). Research–practice partnerships in stem education: An organizational perspective. In O. Hazzan, E. Heyd-Metzuyanim, A. Even-Zahav, T. Tal, & Y. J. Dori (Eds.), *Application of management theories for stem education* (pp. 43-74). Cham, CH: Springer.

Hazzan, O., Lapidot, T., & Ragonis, N. (2014). *Guide to teaching computer science: An activity-based approach* London, UK: Springer-Verlag.

Heaysman, O., & Tubin, D. (2019). Content teaching: Innovative and traditional practices. *Educational Studies, 44*(3), 342-356. doi:10.1080/03055698.2018.1446334

Hempel, C. G. (1983). Kuhn and salmon on rationality and theory choice. *The Journal of Philosophy, 80*(10), 570-572. doi:10.2307/2026151

Henriksen, D., & Mishra, P. (2018). Creativity as invention, discovery, innovation and intuition: An interview with dr. Richard buchanan. *TechTrends, 62*(3), 215-220. doi:10.1007/s11528-018-0279-4

Henriksen, D., Richardson, C., & Mehta, R. (2017). Design thinking: A creative approach to educational problems of practice. *Thinking Skills and Creativity, 26*, 140-153. doi:10.1016/j.tsc.2017.10.001

Herawati, A. (2018). The application of project-based learning in teachers' professional development program. *6th International Conference on Information and Education Technology; Osaka, Japan; ACM*, 119-123. doi:10.1145/3178158.3178182

Heshusius, L. (1990). Discusssion on criteria In E. G. Guba (Ed.), *The paradigm dialog* (pp. 198-202). Thousand Oaks, CA: SAGE Publications, Inc.

Hesse-Biber, S. N., & Leavy, P. (2003). *Approaches to qualitative research: A reader on theory and practice*. Oxford, UK: Oxford University Press.

Hobbs, L., Clark, J. C., & Plant, B. (2018). Successful students–stem program: Teacher learning through a multifaceted vision for stem education. In J. R & L. K. (Eds.), *Stem education in the junior secondary* (pp. 133-168). Singapore, SG: Springer Nature Singapore Pte Ltd.

Hogarty, K. Y., Lang, T. R., & Kromrey, J. D. (2003). Another look at technology use in classrooms: The development and validation of an instrument to measure teachers' perceptions. *Educational and Psychological Measurement, 63*(1), 139-162. doi:10.1177/0013164402239322

Holton, E. F. (1996). The flawed four-level evaluation model. *Human Resource Dvelopment Quarterly, 7*(1), 5-21. doi:10.1002/hrdq.3920070103

Horvath, M., Goodell, J. E., & Kosteas, V. D. (2018). Decisions to enter and continue in the teaching profession: Evidence from a sample of us secondary stem teacher candidates. *Teaching and Teacher Education, 71*, 57-65. doi:10.1016/j.tate.2017.12.007

Houghton, C., Murphy, K., Meehan, B., Thomas, J., Brooker, D., & Casey, D. (2017). From screening to synthesis: Using nvivo to enhance transparency in qualitative evidence synthesis. *Journal of Clinical Nursing, 26*(5-6), 873-881. doi:10.1111/jocn.13443

Howland, J. L., Jonassen, D. H., & Marra, R. M. (2014). *Meaningful learning with technology*. Essex, UK: Pearson Education Limited.

Hubbard, A. (2018). Pedagogical content knowledge in computing education: A review of the research literature. *Computer Science Education, 28*(2), 117-135. doi:10.1080/08993408.2018.1509580

Hubwieser, P., Armoni, M., Giannakos, M. N., & Mittermeir, R. T. (2014). Perspectives and visions of computer science education in primary and secondary (k-12) schools. *ACM Transactions Computing Education, 14*(2), 1-9. doi:10.1145/2602482

Humphrey, A. (2005). Swot analysis for management consulting. *SRI Alumni Newsletter, 1*, 7-8. doi:https://www.sri.com/sites/default/files/brochures/dec-05.pdf

ICS. (2017). The irish computer society – the professional body for it. *Web Page Accessed 15th June 2018.* Retrieved from https://www.ics.ie/training

Informatics Europe. (2018). European computer science summit. Informatics europe. *Web Page Accessed 11th July 2018.* Retrieved from http://www.informatics-europe.org/ecss/about/about-ecss.html

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education, 82*, 263-279. doi:10.1016/j.compedu.2014.11.022

Ivankova, N. V., Creswell, J. W., & Stick, S. L. (2006). Using mixed-methods sequential explanatory design: From theory to practice. *Field methods, 18*(1), 3-20. doi:10.1177/1525822X05282260

Jarvis, P. (2012). *Paradoxes of learning: On becoming an individual in society*. Abingdon, Oxon, UK: Routledge.

JCT. (2017). Junior cycle for teachers. *Web Page Accessed 15th June 2017.* Retrieved from https://www.jct.ie/home/home

Jick, T. D. (1979). Mixing qualitative and quantitative methods: Triangulation in action. *Administrative Science Quarterly, 24*(2), 602-611. doi:https://www.jstor.org/stable/2392366

Johnson, R. B., & Christensen, L. (2014). *Educational research: Quantitative, qualitative, and mixed approaches*. Thousand Oaks, CA SAGE Publications, Inc.

Johnston, K., Conneely, C., Murchan, D., & Tangney, B. (2015). Enacting key skills-based curricula in secondary education: Lessons from a technology-mediated, group-based learning initiative. *Technology, Pedagogy and Education, 24*(4), 423-442. doi:10.1080/1475939X.2014.890641

Jonassen, D. H. (1991). Objectivism versus constructivism: Do we need a new philosophical paradigm? *Educational Technology Research and Development, 39*(3), 5-14. doi:10.1007/BF02296434

Jonassen, D. H. (1999). Designing constructivist learning environments. In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory* (pp. 215-241). Mahwah, NJ: Lawrence Erlbaum Associates, Inc., Publishers.

Jonassen, D. H. (2000). *Computers as mindtools for schools: Engaging critical thinking* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Jonassen, D. H., Howland, J., Moore, J., & Marra, R. M. (2002). *Learning to solve problems with technology: A constructivist perspective*. Upper Saddle River, NJ: Prentice Hall.

Jones, M., & Dexter, S. (2018). Teacher perspectives on technology integration professional development: Formal, informal, and independent learning activities. *Journal of Educational Multimedia and Hypermedia, 27*(1), 83-102. doi:https://www.learntechlib.org/primary/p/178511/

Joshi, A., Kale, S., Chandel, S., & Pal, D. (2015). Likert scale: Explored and explained. *British Journal of Applied Science & Technology, 7*(4), 396-403. doi:10.9734/BJAST/2015/14975

Kaasbøll, J. J. (1998). Exploring didactic models for programming. *NIK 1998 Norsk Informatikkonferanse - Norwegian Computer Science Conference, November 23-25, Kristiansand, Norway*, 195-203. doi:http://nik.no/nik-1998/

Kale, U. (2018). Technology valued? Observation and review activities to enhance future teachers' utility value toward technology integration. *Computers & Education, 117*, 160-174. doi:10.1016/j.compedu.2017.10.007

Kalelioğlu, F. (2015). A new way of teaching programming skills to k-12 students: Code. Org. *Computers in Human Behavior, 52*, 200-210. doi:10.1016/j.chb.2015.05.047

Kaufman, R., Keller, J., & Watkins, R. (1996). What works and what doesn't: Evaluation beyond kirkpatrick. *Performance+ Instruction, 35*(2), 8-12. doi:10.1002/pfi.4170350204

Kaufman, R., & Keller, J. M. (1994). Levels of evaluation: Beyond kirkpatrick. *Human Resource Development Quarterly, 5*(4), 371-380. doi:10.1002/hrdq.3920050408

Kay, J., Barg, M., Fekete, A., Greening, T., Hollands, O., Kingston, J. H., & Crawford, K. (2000). Problem-based learning for foundation computer science courses. *Computer Science Education, 10*(2), 109-128. doi:10.1076/0899-3408(200008)10:2;1-C;FT109

Kearney, S. (2018). Using the bridge21 pedagogical model of 21 st century teaching and learning to teach new literacies in second-level english education, and the impact on students' attitudes. *Ph.D Thesis, Trinity College Dublin. Ireland*, 1-309. doi:http://hdl.handle.net/2262/83160

Kelchtermans, G. (2004). Cpd for professional renewal: Moving beyond knowledge for practice. In C. Day & J. Sachs (Eds.), *International handbook on the continuing professional development of teachers* (pp. 217-237). Maidenhead, UK: Open University Press.

Kellaghan, T., Stufflebeam, D. L., & Wingate, L. A. (2012). *International handbook of educational evaluation*. Dordrecht, NL: Hluwer Academic Publishers.

Kelley, T. R., & Knowles, J. G. (2016). A conceptual framework for integrated stem education. *International Journal of STEM Education, 3*(1), 1-11. doi:10.1186/s40594-016-0046-z

Kennedy, A. (2005). Models of continuing professional development: A framework for analysis. *Journal of In-Service Education, 31*(2), 235-250. doi:10.1080/13674580500200277

Kennedy, A. (2011). Collaborative continuing professional development (cpd) for teachers in scotland: Aspirations, opportunities and barriers. *European Journal of Teacher Education, 34*(1), 25-41. doi:10.1080/02619768.2010.534980

Kennedy, A. (2014). Understanding continuing professional development: The need for theory to impact on policy and practice. *Professional Development in Education, 40*(5), 688-697. doi:10.1080/19415257.2014.955122

Kennedy, A. (2016). Professional learning in and for communities: Seeking alternatives discourses. *Professional Development in Education, 42*(5), 667-670. doi:10.1080/19415257.2016.1220541

Kent, S. (1996). *Cultural diversity among twentieth century foragers: An african perspective*. Cambridge, UK: Cambridge University Press.

Kheir, N., & Wilbur, K. (2018). Continuing professional development and self-learning for pharmacists. In A. I. Fathelrahman, M. I. M. Ibrahim, A. A. Alrasheedy, & A. I. Wertheimer (Eds.), *Pharmacy education in the twenty first century and beyond* (pp. 191-200). Amsterdam, Netherlands: Elsevier Inc.

Kilpatrick, S., & Fraser, S. (2018). Using the stem framework collegially for mentoring, peer learning and planning. *Professional Development in Education*, 1-13. doi:10.1080/19415257.2018.1463925

Kim, B., Park, H., & Baek, Y. (2009). Not just fun, but serious strategies: Using meta-cognitive strategies in game-based learning. *Computers & Education, 52*(4), 800-810. doi:10.1016/j.compedu.2008.12.004

Kim, S., Song, K., Lockee, B., & Burton, J. (2018). *Gamification in learning and education*. Cham, CH: Springer International Publishing Switzerland.

King, A. (1994). Guiding knowledge construction in the classroom: Effects of teaching children how to question and how to explain. *American Educational Research Journal, 31*(2), 338-368. doi:10.3102/00028312031002338

King, K. P. (2002). Educational technology professional development as transformative learning opportunities. *Computers & Education, 39*(3), 283-297. doi:10.1016/S0360-1315(02)00073-8

Kirkpatrick, D. L. (1956). How to start an objective evaluation of your training program. *Journal of the American Society of Training Directors, 10*, 18-22. doi:https://www.kirkpatrickpartners.com/LinkClick.aspx?fileticket=DVoFB3I1gjo%3D&tabid=56&portalid=0&mid=415

Kirkpatrick, D. L. (1994). *Evaluating training programs. The four levels*. San Francisco, CA: Berrett-Koehler Publishers.

Kirkpatrick, D. L. (2007). *The four levels of evaluation. Tips, tools, and intelligance for trainers* (Vol. 0701). Alexandria, VA: ASTD - American Society for Training and Development.

Kirkpatrick, D. L. (2016). Forward by don kirkpatrick. In J. Kirkpatrick & W. K. Kirkpatrick (Eds.), *Kirkpatrick's four levels of training evaluation* (pp. ix). Alexandria, VA: ATD Press.

Kirkpatrick Partners. (2018). The new world kirkpatrick model *Web Page Accessed 30th June 2018.* Retrieved from https://www.kirkpatrickpartners.com/Our-Philosophy/The-New-World-Kirkpatrick-Model

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist, 41*(2), 75-86. doi:10.1207/s15326985ep4102_1

Kirschner, P. A., Sweller, J., Kirschner, F., & Zambrano R, J. (2018). From cognitive load theory to collaborative cognitive load theory. *International Journal of Computer-Supported Collaborative Learning, 13*(2), 213–233. doi:10.1007/s11412-018-9277-y

Kitts, K. (2009). The paradox of middle and high school students' attitudes towards science versus their attitudes about science as a career. *Journal of Geoscience Education, 57*(2), 159-164. doi:10.5408/1.3544253

Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly, 23*(1), 67-93. doi:10.2307/249410

Knowles, M. S. (1988). Preface. In D. Boud (Ed.), *Developing student autonomy in learning* (Second Edition ed., pp. 1-6). London, UK: Taylor & Francis

Ko, A. J., Myers, B. A., & Aung, H. H. (2004). Six learning barriers in end-user programming systems. *2004 Symposium on Visual Languages and Human Centric Computing; Rome, Italy; IEEE*, 199-206. doi:10.1109/VLHCC.2004.47

Koehler, M., Greenhalgh, S., Rosenberg, J., & Keenan, S. (2017). What the tech is going on with teachers’ digital teaching portfolios? Using the tpack framework to analyze teachers’ technological understanding. *Journal of Technology and Teacher Education, 25*(1), 31-59. doi:www.learntechlib.org/primary/p/173346/

Koehler, M., & Mishra, P. (2009). What is technological pedagogical content knowledge (tpack)? *Contemporary Issues in Technology and Teacher Education Journal, 9*(1), 60-70. doi:https://www.learntechlib.org/p/29544/

Kong, R., & Wong, G. K. (2017). Teachers' perception of professional development in coding education. *6th International Conference onTeaching, Assessment, and Learning for Engineering; Hong Kong, China; IEEE*, 377-380. doi:10.1109/TALE.2017.8252365

Krippendorff, K., & Bock, M. A. (2009). *The content analysis reader*. Thousand Oaks, CA: SAGE Publications, Inc.

Kristiansen, N. (2007). Level 1 training evaluation form. In D. Kirkpatrick (Ed.), *The four levels of evaluation. Measurement & evaluation. Tips, tools, and intelligence for trainers* (Vol. 0701, pp. 3). Alexandria, VA: ASTD.

Krolak-Schwerdt, S., Glock, S., & Böhmer, M. (2014). *Teacher's professional development: Assessment, training, and learning*. Dordrecht: NL: Springer Science & Business Media.

Kubica, J. (2012). *Computational fairy tales*. New York, NY: CreateSpace Independent Publishing Platform.

Kubica, J. (2013). *Best practices of spell design*. New York, NY: CreateSpace Independent Publishing Platform.

Kubica, J. (2016). *The cs detective. An algorithmic tale of crime, conspiracy and computation*. San Francisco, CA: No Startch Press.

Kuhn, T. S. (1970). *The structure of scientific revolutions*. Chicago, ILL: University of Chicago Press.

Lambert, J., Cioc, C., Cioc, S., & Sandt, D. (2018). Making connections: Evaluation of a professional development program for teachers focused on stem integration. *Journal of STEM Teacher Education, 53*(1), 3-25. doi:10.30707/JSTE53.1Lambert

Lampley, S. A., Gardner, G. E., & Barlow, A. T. (2018). Exploring pedagogical content knowledge of biology graduate teaching assistants through their participation in lesson study. *Teaching in Higher Education, 23*(4), 468-487. doi:10.1080/13562517.2017.1414786

Landers, R. N., & Armstrong, M. B. (2017). Enhancing instructional outcomes with gamification: An empirical test of the technology-enhanced training effectiveness model. *Computers in Human Behavior, 71*, 499-507. doi:10.1016/j.chb.2015.07.031

Laurillard, D. (2013). Forward to the second edition. In H. Beetham & R. Sharpe (Eds.), *Rethinking pedagogy for a digital age: Designing for 21st century learning - designing for 21st century learning* (Second Edition ed., pp. xvi - xix). New York, NY: Routledge.

Lave, J. (1988). *Cognition in practice: Mind, mathematics and culture in everyday life*. Cambridge, UK: Cambridge University Press.

Lavonen, J. M., Meisalo, V. P., Lattu, M., & Sutinen, E. (2003). Concretising the programming task: A case study in a secondary school. *Computers & Education, 40*(2), 115-135. doi:http://dx.doi.org/10.1016/S0360-1315(02)00101-X

Lawless, K. A., & Pellegrino, J. W. (2007). Professional development in integrating technology into teaching and learning: Knowns, unknowns, and ways to pursue better questions and answers. *Review of Educational Research, 77*(4), 575-614.

Lawlor, J. (2016). Bridge21 - a model for team based, technology, mediated learning in and out of school context. *Ph.D Thesis, Trinity College Dublin. Ireland*, 1-307. doi:http://hdl.handle.net/2262/83464

Lawlor, J., Conneely, C., Oldham, E., Marshall, K., & Tangney, B. (2018). Bridge21: Teamwork, technology and learning. A pragmatic model for effective twenty-first-century team-based learning. *Technology, Pedagogy and Education, 27*(2), 211-232. doi:10.1080/1475939X.2017.1405066

Lawlor, J., Conneely, C., & Tangney, B. (2013). An exploratory case study into the adaptation of the bridge21 model for 21st century learning in irish classrooms. In K. Marshall (Ed.), *Shaping our future: How the lessons of the past can shape educational transformation* (pp. 1-16). Dublin, IE: Liffey Press.

Lawlor, J., Marshall, K., & Tangney, B. (2016). Bridge21–exploring the potential to foster intrinsic student motivation through a team-based, technology-mediated learning model. *Technology, Pedagogy and Education, 25*(2), 187-206. doi:10.1080/1475939X.2015.1023828

LeCompte, M. D., & Goetz, J. P. (1982). Problems of reliability and validity in ethnographic research. *Review of Educational Research, 52*(1), 31-60. doi:10.3102/00346543052001031

LeCompte, M. D., & Schensul, J. J. (1999). *Analyzing and interpreting ethnographic data* (Vol. 5). Walnut Creek: AltaMira Press.

LeCompte, M. D., & Schensul, J. J. (2013). Recursivity, induction and deduction: Moving between levels of abstraction. In *Analysis and interpretation of ethnographic data: A mixed methods approach* (Vol. 5, pp. 65-79). Lanham, Maryland: AltaMira Press.

Lee, C. J., & Kim, C. (2017). A technological pedagogical content knowledge based instructional design model: A third version implementation study in a technology integration course. *Educational Technology Research and Development, 65*(6), 1627-1654. doi:10.1007/s11423-017-9544-z

Lee, R. E. (2018). Breaking down barriers and building bridges: Transformative practices in community- and school-based urban teacher preparation. *Journal of Teacher Education, 69*(2), 118-126. doi:10.1177/0022487117751127

Lefkowitz, J. (1972). Evaluation of a supervisory training program for police sergeants. *Personnel Psychology, 25*(1), 95-106. doi:10.1111/j.1744-6570.1972.tb01093.x

Lewis, C. M. (2011). Is pair programming more effective than other forms of collaboration for young students? *Computer Science Education, 21*(2), 105-134. doi:10.1080/08993408.2011.579805

Leys, C., Ley, C., Klein, O., Bernard, P., & Licata, L. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology, 49*(4), 764-766. doi:10.1016/j.jesp.2013.03.013

Lieberman, H. (2001). *Your wish is my command: Programming by example*. San Francisco, CA: Morgan Kaufmann Publishers.

Lincoln, Y. S. (2002). Emerging criteria for quality in qualitative and interpretive research. In N. K. Denzin & Y. S. Lincoln (Eds.), *The qualitative inquiry reader*. Thousand Oaks, CA: SAGE Publications, Inc.

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. Newbury Park, CA: SAGE Publications, Inc.

Lingard, R. (2010). Teaching and assessing teamwork skills in engineering and computer science. *Journal of Systemics, Cybernetics and Informatics, 18*(1), 34-37. doi:10.1.1.458.4703

Lingard, R., & Barkataki, S. (2011). Teaching teamwork in engineering and computer science. *2011 Frontiers in Education Conference; Rapid City, SD, USA; IEEE*, 1-5. doi:10.1109/FIE.2011.6143000

Lister, R. (2011). Concrete and other neo-piagetian forms of reasoning in the novice programmer. *13th Australasian Computing Education Conference; Perth, Australia; ACM, 114*, 9-18. doi:https://dl.acm.org/citation.cfm?id=2459938

Lister, R. (2016). Toward a developmental epistemology of computer programming. *11th Workshop in Primary and Secondary Computing Education; Münster, Germany; ACM*, 5-16. doi:10.1145/2978249.2978251

Lister, R., Berglund, A., Box, I., Cope, C., Pears, A., Avram, C., . . . de Raadt, M. (2007). Differing ways that computing academics understand teaching. *9th Australasian Conference on Computing Education; Ballarat, Victoria, Australia; ACM, 66*, 97-106. doi:https://dl.acm.org/citation.cfm?id=1273684

Littell, J. H., Corcoran, J., & Pillai, V. (2008). *Systematic reviews and meta-analysis*. Oxford, UK: Oxford University Press.

Little, D. (1995). Learning as dialogue: The dependence of learner autonomy on teacher autonomy. *System, 23*(2), 175-181. doi:http://dx.doi.org/10.1016/0346-251X(95)00006-6

Lofland, J. (1971). *Analyzing social settings*. Belmont, CA: Wadsworth Publishing Company.

Lofland, J., & Lofland, L. H. (1984). *Analysing social settings: A guide to qualitative observation and analysis*. Belmont, CA: Wadsworth.

Loucks-Horsley, S., Stiles, K. E., Mundry, S., Love, N., & Hewson, P. W. (2009). *Designing professional development for teachers of science and mathematics* (3rd ed.). Thousand Oaks, CA: Corwin, A SAGE Company.

Luckin, R. (2010). *Re-designing learning contexts: Technology-rich, learner-centred ecologies*. Abingdon, Oxon: Routledge.

Luckin, R., & Du Boulay, B. (1999). Ecolab: The development and evaluation of a vygotskian design framework. *International Journal of Artificial Intelligence in Education, 10*(2), 198-220. doi:10.1.1.323.7281

Lundorff, M., Holmgren, H., Zachariae, R., Farver-Vestergaard, I., & O'Connor, M. (2017). Prevalence of prolonged grief disorder in adult bereavement: A systematic review and meta-analysis. *Journal of Affective Disorders, 212*, 138-149. doi:10.1016/j.jad.2017.01.030

Lydon, S., & King, C. (2009). Can a single, short continuing professional development workshop cause change in the classroom? *Professional Development in Education, 35*(1), 63-82. doi:10.1080/13674580802264746

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for k-12? *Computers in Human Behavior, 41*, 51-61. doi:10.1016/j.chb.2014.09.012

Lynch, B. K. (1996). *Language program evaluation: Theory and practice*. Cambridge, UK: Cambridge University Press.

Ma, S., Shang, J., & Xiao, H. (2017). A curriculum design to cultivate creative thinking through educational games. In K. L. Cheung S., Ma W., Lee LK., Yang H (Ed.), *Blended learning. New challenges and innovative practices. Icbl 2017. Lecture notes in computer science* (Vol. 10309, pp. 453-463). Cham, CH: Springer.

Maiorana, F., Berry, M., Nelson, M., Lucarelli, C., Phillipps, M., Mishra, S., & Benassi, A. (2017). International perspectives on cs teacher formation and professional development. *2017 Conference on Innovation and Technology in Computer Science Education; Bologna, Italy; ACM*, 236-237. doi:10.1145/3059009.3059067

Major, L., Kyriacou, T., & Brereton, O. P. (2012). Systematic literature review: Teaching novices programming using robots. *Institution of Engineering and Technology Software, 6*(6), 502-513. doi:10.1049/iet-sen.2011.0125

Makki, T. W., O'Neal, L. J., Cotten, S. R., & Rikard, R. V. (2018). When first-order barriers are high: A comparison of second- and third-order barriers to classroom computing integration. *Computers & Education, 120*, 90-97. doi:10.1016/j.compedu.2018.01.005

Malan, S. (2000). The'new paradigm'of outcomes-based education in perspective. *Journal of Consumer Sciences, 28*(1), 22-28. doi:10.1.1.564.1106

Malec, M. (2018). *Essential statistics for social research* (2 ed.). New York, NY: Routledge.

Marshall, G. (1998). *A dictionary of sociology*. Oxford, UK: Oxford Paperbacks.

Marshall, S. P. (2009). Re-imagining specialized stem academies: Igniting and nurturing decidedly different minds, by design. *Roeper Review, 32*(1), 48-60. doi:10.1080/02783190903386884

Martin, L. E., Kragler, S., Quatroche, D. J., Bauserman, K. L., & Hargreaves, A. (2014). *Handbook of professional development in education: Successful models and practices, prek-12*. New York, NY: Guilford Publications.

Martinez, M. C., Gomez, M. J., Moresi, M., & Benotti, L. (2016). Lessons learned on computer science teachers professional development. *2016 Conference on Innovation and Technology in Computer Science Education; Arequipa, Peru; ACM*, 77-82. doi:10.1145/2899415.2899460

Martinovic, D., Ezeife, C. I., Whent, R., Reed, J., Burgess, G. H., Pomerleau, C. M., . . . Chaturvedi, R. (2014). “Critic-proofing” of the cognitive aspects of simple games. *Computers & Education, 72*, 132-144. doi:http://dx.doi.org/10.1016/j.compedu.2013.10.017

Matthíasdóttir, Á., & Arnalds, H. (2015). Rethinking teaching and assessing in a programming course a case study. *16th International Conference on Computer Systems and Technologies; Dublin, Ireland; ACM*, 313-318. doi:10.1145/2812428.2812470

Maxwell, J. A., Bashook, P. G., & Sandlow, L. J. (1986). *Educational evaluation - ethnography in theory, practice and politics*. Beverly Hills, CA: SAGE Publications, Inc.

Mayer, R. E. (1975). Different problem-solving competencies established in learning computer programming with and without meaningful models. *Journal of Educational Psychology, 67*(6), 725. doi:https://eric.ed.gov/?id=EJ133634

Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? *American Psychologist, 59*(1), 14-19. doi:10.1037/0003-066X.59.1.14

Mayer, R. E. (2008). *Teaching and learning computer programming: Multiple research perspectives*. New York, NY: Routledge.

Mayorova, V., Grishko, D., & Leonov, V. (2018). New educational tools to encourage high-school students' activity in stem. *Advances in Space Research, 61*(1), 457-465. doi:10.1016/j.asr.2017.07.037

McInerney, D. M. (2013). *Educational psychology: Constructing learning* (6th ed.). Melbourne, AU: Pearson Higher Education.

McLaughlin, J. A., & Jordan, G. B. (2010). Using logical models In J. S. Wholey, H. P. Hatry, & K. E. Newcomer (Eds.), *Handbook of practical program evaluation* (3rd ed., Vol. 19, pp. 55-81). San Fransisco, CA: John Wiley & Sons

McLaughlin, J. A., & Jordan, G. B. (2015). Using logical models. In K. E. Newcomer, H. P. Hatry, & J. S. Wholey (Eds.), *Handbook of practical program evaluation* (4th ed., pp. 62-88). Hoboken: John Wiley & Sons, Inc.

McMillan, D. J., McConnell, B., & O'Sullivan, H. (2016). Continuing professional development – why bother? Perceptions and motivations of teachers in ireland. *Professional Development in Education, 42*(1), 150-167. doi:10.1080/19415257.2014.952044

Mead, G. H. (1934). *Mind, self and society*. Chicago, ILL: University of Chicago Press.

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2011). Habits of programming in scratch. *16th Annual Joint Conference on Innovation and Technology in Computer Science Education; Darmstadt, Germany; ACM*, 168-172. doi:10.1145/1999747.1999796

Melcer, E. F., & Isbister, K. (2018). Bots & (main) frames: Exploring the impact of tangible blocks and collaborative play in an educational programming game. *2018 Conference on Human Factors in Computing Systems; Montreal QC, Canada; ACM*, 1-14. doi:https://dl.acm.org/citation.cfm?id=3173574.3173840

Menekse, M. (2015). Computer science teacher professional development in the united states: A review of studies published between 2004 and 2014. *Computer Science Education, 25*(4), 325-350. doi:10.1080/08993408.2015.1111645

Merchie, E., Tuytens, M., Devos, G., & Vanderlinde, R. (2018). Evaluating teachers' professional development initiatives: Towards an extended evaluative framework. *Research Papers in Education, 33*(2), 143-168. doi:10.1080/02671522.2016.1271003

Mezirow, J. (1996). Contemporary paradigms of learning. *Adult Education Quarterly, 46*(3), 158-172. doi:10.1177/074171369604600303

Michalski, G. V., & Cousins, J. B. (2000). Differences in stakeholder perceptions about training evaluation: A concept mapping/pattern matching investigation. *Evaluation and Program Planning, 23*(2), 211-230.

Michie, D., & Johnston, R. (1985). *The creative computer - machine intelligence and human knowledge* Middlesex, UK: Pelican Books

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: A methods sourcebook* (2nd Ed.). Thousand Oaks, CA: SAGE Publications, Inc.

Miller, K., Sonnert, G., & Sadler, P. (2018). The influence of students' participation in stem competitions on their interest in stem careers. *International Journal of Science Education, 8*(2), 95-114. doi:10.1080/21548455.2017.1397298

Millwood, R., Strong, G., Bresnihan, N., & Cowan, P. (2016). Ctwins: Improving computational thinking confidence in educators through paired activities. *11th Workshop in Primary and Secondary Computing Education; Münster, Germany; ACM*, 106-107 doi:10.1145/2978249.2978269

Milne, I., & Rowe, G. (2002). Difficulties in learning and teaching programming—views of students and tutors. *Education and Information Technologies, 7*(1), 55-66. doi:10.1023/A:1015362608943

Mingers, J. (1992). Criticizing the phenomenological critique—autopoiesis and critical realism. *Systems Practice, 5*(2), 173-180. doi:10.1007/BF01059939

Mishra, P., & Henriksen, D. (2018). *Creativity, technology & education: Exploring their convergence*. Cham, CH: Springer.

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record, 108*(6), 1017-1054. doi:10.1111/j.1467-9620.2006.00684.x

Moffie, D. J., Calhoon, R., & O'Brien, J. K. (1964). Evaluation of a management development program. *Personnel Psychology, 17*(4), 431-440. doi:10.1111/j.1744-6570.1964.tb00078.x

Moller, L., Foshay, W. R., & Huett, J. (2008). Implications for instructional design on the potential of the web. *TechTrends, 52*(3), 70-75. doi:10.1007/s11528-008-0158-5

Montgomery, C., & Fernández-Cárdenas, J. M. (2018). Teaching stem education through dialogue and transformative learning: Global significance and local interactions in mexico and the uk. *Journal of Education for Teaching, 44*(1), 2-13. doi:10.1080/02607476.2018.1422606

Moon, B., Butcher, J., & Bird, E. (2000). *Leading professional development in education*. London, UK: RoutledgeFalmer.

Mosson, R., Augustsson, H., Bäck, A., Åhström, M., von Thiele Schwarz, U., Richter, A., . . . Hasson, H. (2019). Building implementation capacity (bic): A longitudinal mixed methods evaluation of a team intervention. *BMC Health Services Research, 19*(287), 1-12. doi:10.1186/s12913-019-4086-1

Mouza, C., Yadav, A., & Ottenbreit-Leftwich, A. (2018). Developing computationally literate teachers: Current perspectives and future directions for teacher preparation in computing education. *Journal of Technology and Teacher Education, 26*(3), 333-352. doi:https://www.learntechlib.org/p/184602/

Muijs, D., Day, C., Harris, A., & Lindsay, G. (2004). Evaluating cpd: An overview. In C. Day & J. Sachs (Eds.), *International handbook on the continuing professional development of teachers* (pp. 291-311). Maidenhead, UK: Open University Press.

Muijs, D., & Lindsay, G. (2008). Where are we at? An empirical study of levels and methods of evaluating continuing professional development. *British Educational Research Journal, 34*(2), 195-211. doi:10.1080/01411920701532194

Murtonen, M., Gruber, H., & Lehtinen, E. (2017). The return of behaviourist epistemology: A review of learning outcomes studies. *Educational Research Review, 22*, 114-128. doi:10.1016/j.edurev.2017.08.001

Myers, M. D. (1999). Investigating information systems with ethnographic research. *Communications of the Association for Information Systems, 2*(23), 1-10. doi:dl.acm.org/citation.cfm?id=374486.374487

Nadelson, L. S., Callahan, J., Pyke, P., Hay, A., Dance, M., & Pfiester, J. (2013). Teacher stem perception and preparation: Inquiry-based stem professional development for elementary teachers. *The Journal of Educational Research, 106*(2), 157-168. doi:10.1080/00220671.2012.667014

Nadelson, L. S., Seifert, A., Moll, A. J., & Coats, B. (2012). I-stem summer institute: An integrated approach to teacher professional development in stem. *Journal of STEM Education: Innovations and Research, 13*(2), 69-83. doi:https://jstem.org/index.php/JSTEM/article/view/1644

Naugle, K. A., Naugle, L. B., & Naugle, R. J. (2000). Kirkpatrick's evaluation model as a means of evaluating teacher performance. *Education, 121*(1), 135-144.

NCCA. (2014a). Short course - coding. *An Chomhairle Náisiúnta Curaclaim agus Measúnachta / National Council for Curriculum and Assessment, Web Page Accessed 8th Feburary 2015.* Retrieved from http://www.curriculumonline.ie/Junior-cycle/Short-Courses/Coding

NCCA. (2014b). Short course - digital media literacy. *An Chomhairle Náisiúnta Curaclaim agus Measúnachta / National Council for Curriculum and Assessment, Web Page Accessed 8th Feburary 2015.* Retrieved from https://www.curriculumonline.ie/Junior-cycle/Short-Courses/Digital-Media-Literacy

NCCA. (2015). Juniorcycle.Ie - statements of learning. *An Chomhairle Náisiúnta Curaclaim agus Measúnachta / National Council for Curriculum and Assessment, Web Page Accessed 14th September 2015.* Retrieved from https://www.education.ie/en/Publications/Policy-Reports/Framework-for-Junior-Cycle-2015.pdf

NCCA. (2017). Draft curriculum specification for leaving certificate computer science. *An Chomhairle Náisiúnta Curaclaim agus Measúnachta / National Council for Curriculum and Assessment, Web Page Accessed 8th Feburary 2017.* Retrieved from http://www.ncca.ie/en/Curriculum_and_Assessment/Post-Primary_Education/Senior_Cycle/Consultation/Draft-ComputerScience-Spec-Post-Consultation.pdf

NCCA. (2018a). Coding in primary schools. National council for curriculum and assessment. *An Chomhairle Náisiúnta Curaclaim agus Measúnachta / National Council for Curriculum and Assessment, Web Page Accessed 26th May 2018.* Retrieved from https://www.ncca.ie/en/primary/primary-developments/coding-in-primary-schools

NCCA. (2018b). Computer science curriculum specification - leaving certificate ordinary and higher level. *An Chomhairle Náisiúnta Curaclaim agus Measúnachta / National Council for Curriculum and Assessment, Web Page Accessed 2nd July 2018.* Retrieved from https://www.ncca.ie/media/3369/computer-science-final-specification.pdf

NCCA. (2018c). Computer science. National council for curriculum and assessment. *An Chomhairle Náisiúnta Curaclaim agus Measúnachta / National Council for Curriculum and Assessment, Web Page Accessed 2nd July 2018.* Retrieved from https://www.ncca.ie/en/senior-cycle/curriculum-developments/subjects-and-frameworks-in-development/computer-science

NCCA. (2018d). Lccs 2018 information - leaving certificate computer science - ncca.Ie/en/senior-cycle. *National Council for Curriculum and Assessment, Web Page Accessed 02 July 2018.* Retrieved from https://www.ncca.ie/media/3261/lccs-information-brochure.pdf

NCCA. (2018e). Short course - coding - statements of learning *An Chomhairle Náisiúnta Curaclaim agus Measúnachta / National Council for Curriculum and Assessment, Web Page Accessed 2nd July 2018.* Retrieved from https://curriculumonline.ie/Junior-cycle/Short-Courses/Coding/Statements-of-Learning

NCCA. (2019). Leaving certificate computer science coursework assessment guidelines 201904. *An Chomhairle Náisiúnta Curaclaim agus Measúnachta / National Council for Curriculum and Assessment, Web Page Accessed 25/05 May 2019.* Retrieved from https://curriculumonline.ie/getmedia/a4feb183-69fe-4144-9add-56a13a5aa3f8/CS-Coursework-Assessment-Guidelines-201904_en.pdf

Nelson, R., Spence-Thomas, K., & Taylor, C. (2015). What makes great pedagogy and great professional development. Final report: Teaching schools r&d network national themes projects 2012-14: Spring 2015.

Neutens, T., & Wyffels, F. (2018). Bringing computer science education to secondary school: A teacher first approach. *2018 Technical Symposium on Computer Science Education; Baltimore, MD, USA; ACM*, 840-845. doi:10.1145/3159450.3159568

Newcomer, K. E., Hatry, H. P., & Wholey, J. S. (2010). Planning and designing useful evaluations In J. S. Wholey, H. P. Hatry, & K. E. Newcomer (Eds.), *Handbook of practical program evaluation* (3rd ed., Vol. 19, pp. 5-30). San Fransisco, CA: John Wiley & Sons

Newton, X. A. (2018). *Improving teacher knowledge in k-12 schooling: Perspectives on stem learning*. Basinstoke, UK: Palgrave MacMillan.

Noblit, G. W., & Hare, R. D. (1988). *Meta-ethnography: Synthesizing qualitative studies*. Thousand Oaks, CA: SAGE Publications, Inc.

Noe, R. A., & Schmitt, N. (1986). The influence of trainee attitudes on training effectiveness: Test of a model. *Personnel Psychology, 39*(3), 497-523. doi:10.1111/j.1744-6570.1986.tb00950.x

Nørmark, K., Thomsen, L. L., & Torp, K. (2008). Mini project programming exams. In J. Bennedsen, M. E. Caspersen, & M. Kölling (Eds.), *Reflections on the teaching of programming: Methods and implementations* (pp. 228-244). Berlin, Heidelberg: Springer-Verlag.

Ntemngwa, C., & Oliver, S. (2018). The implementation of integrated science technology, engineering and mathematics (stem) instruction using robotics in the middle school science classroom. *International Journal of Education in Mathematics, Science and Technology, 6*(1), 12-40. doi:10.18404/ijemst.380617

NUI Galway. (2018). Nui galway receives funding from google for computer science professional development programmes. Nui galway. *Web Page Accessed 30th June 2018.* Retrieved from https://www.nuigalway.ie/about-us/news-and-events/news-archive/2018/june/nui-galway-receives-funding-from-google-for-computer-science-professional-development-programmes.html

NUI Maynooth. (2017). A computational thinking initiative for primary and second level education. Nui maynooth. *Web Page Accessed 15th June 2018.* Retrieved from https://www.maynoothuniversity.ie/computer-science/pact

O'Donoghue, T. (2007). *Planning your qualitative research project: An introduction to interpretivist research in education*. Abingdon, Oxon, UK: Routledge.

O'Donovan, D. (2014). Junior cycle reform is a chance to give meaningful lessons in history - getting hung up on the removal of history as a core subject misses the bigger opportunities in the new curriculum. *The Irish Times*. Retrieved from https://www.irishtimes.com/news/education/junior-cycle-reform-is-a-chance-to-give-meaningful-lessons-in-history-1.1774128

O'Donovan, D. (2015). Enquiry based learning at bridge21 *Web Page Accessed 5th August 2015.* Retrieved from https://sites.google.com/site/enquirybasedlearningatbridge21/home

O'Donovan, D., & Kearney, S. (2015). Contextualising humanities and technology skills in heritage sites; february 28; galway, ireland. *CESI - beyond the horizon technology in education, http://www.cesi.ie/conferences/2015-beyond-the-horizon-technology-in-education/*.

O'Neil, F., & Pegrum, M. (2018). Keeping up the momentum: A longitudinal evaluation of professional development in digital technologies for academic librarians at an australian university. *The Journal of Academic Librarianship, 44*(4), 439-445. doi:https://doi.org/10.1016/j.acalib.2018.05.009

Olofsson, A. D., Lindberg, J. O., & Fransson, G. (2017). What do upper secondary school teachers want to know from research on the use of ict and how does this inform a research design? *Education and Information Technologies,, 22*(6), 2897-2914. doi:10.1007/s10639-017-9590-5

Olsen, B. (2015). *Teaching what they learn, learning what they live: How teachers' personal histories shape their professional development*. London, UK: Taylor & Francis.

Onwuegbuzie, A. J., & Weinbaum, R. K. (2017). A framework for using qualitative comparative analysis for the review of the literature. *The Qualitative Report, 22*(2), 359-372. doi:https://nsuworks.nova.edu/tqr/vol22/iss2/1

Oppenheim, A. N. (2000). *Questionnaire design, interviewing and attitude measurement* (2nd ed.). New York, NY: Continuum.

Ott, L., Ureel, L. C., & Wallace, C. (2018). Building and nurturing communities of practice among k-12 cs teachers. *2018 Technical Symposium on Computer Science Education; Baltimore, MD, USA; ACM*, 1067-1067. doi:10.1145/3159450.3162190

P21. (2017). Partnership for 21st century skills - framework for 21st century learning. *Web Page Accessed 5th Feburary 2017.* Retrieved from http://www.p21.org/our-work/p21-framework

Papadakis, S. (2018). Is pair programming more effective than solo programming for secondary education novice programmers? A case study. *International Journal of Web-Based Learning and Teaching Technologies, 13*(1), 1-16. doi:10.4018/IJWLTT.2018010101

Papert, S. (1993). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books

Paras, M. K., Pal, R., & Ekwall, D. (2017). Systematic literature review to develop a conceptual framework for a reuse-based clothing value chain. *The International Review of Retail, Distribution and Consumer Research, 28*(3), 231-258. doi:10.1080/09593969.2017.1380066

Paraskeva, F., Bouta, H., & Papagianni, A. (2008). Individual characteristics and computer self-efficacy in secondary education teachers to integrate technology in educational practice. *Computers & Education, 50*(3), 1084-1091. doi:10.1016/j.compedu.2006.10.006

Paré, G., Trudel, M.-C., Jaana, M., & Kitsiou, S. (2015). Synthesizing information systems knowledge: A typology of literature reviews. *Information & Management, 52*(2), 183-199. doi:https://doi.org/10.1016/j.im.2014.08.008

Pareja Roblin, N., Tondeur, J., Voogt, J., Bruggeman, B., Mathieu, G., & van Braak, J. (2018). Practical considerations informing teachers' technology integration decisions: The case of tablet pcs. *Technology, Pedagogy and Education, 27*(2), 1-17. doi:10.1080/1475939X.2017.1414714

Pass, S. (2004). *Parallel paths to constructivism: Jean piaget and lev vygotsky*. Greenwich, UK: Information Age Pub.

Passey, D. (2006). Technology enhancing learning: Analysing uses of information and communication technologies by primary and secondary school pupils with learning frameworks. *The Curriculum Journal, 17*(2), 139-166. doi:10.1080/09585170600792761

Passey, D. (2017). Computer science (cs) in the compulsory education curriculum: Implications for future research. *Education and Information Technologies, 22*(2), 421-443. doi:10.1007/s10639-016-9475-z

Patton, M. Q. (2011). *Developmental evaluation: Applying complexity concepts to enhance innovation and use*. New York, NY: Guilford Press.

PDST. (2017). Professional development service for teachers (pdst). *Web Page Accessed 12th June 2018.* Retrieved from http://www.pdst.ie/postprimary

Pea, R. D. (1987). Socializing the knowledge transfer problem. *International Journal of Educational Research, 11*(6), 639-663. doi:10.1016/0883-0355(87)90007-3

Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology, 2*(2), 137-168. doi:0.1016/0732-118X(84)90018-7

Perkins, D. N., & Martin, F. (1986). Empirical studies of programmers. In E. Soloway & S. Iyengar (Eds.), (pp. 213-230). Norwood, NJ: Ablex Publishing Corproation.

Perkins, D. N., & Salomon, G. (1988). Teaching for transfer. *Educational Leadership, 46*(1), 22-32. doi:https://pdfs.semanticscholar.org/d1fe/324a117c069b09cbc4ae8a82c5ac18ba3ac9.pdf

Perry, E., & Boylan, M. (2018). Developing the developers: Supporting and researching the learning of professional development facilitators. *Professional Development in Education, 44*(2), 254-271. doi:10.1080/19415257.2017.1287767

Persico, D., Manca, S., & Pozzi, F. (2014). Adapting the technology acceptance model to evaluate the innovative potential of e-learning systems. *Computers in Human Behavior, 30*, 614-622. doi:10.1016/j.chb.2013.07.045

Phillips, M., Koehler, M., Rosenberg, J., & Zunica, B. (2017). Unpacking tpack: Reconsidering knowledge and context in teacher practice. *2017 Society for Information Technology & Teacher Education International Conference; Austin, TX, USA; AACE*, 2422-2429. doi:http://www.learntechlib.org/primary/p/177538/

Piaget, J. (1950). *The psychology of intelligence* (M. Piercy & D. E. Berlyne, Trans.). London, UK: Routledge.

Pollock, L., Mouza, C., Czik, A., Little, A., Coffey, D., & Buttram, J. (2017). From professional development to the classroom: Findings from cs k-12 teachers. *2017 Technical Symposium on Computer Science Education; Seattle, WN, USA; ACM*, 477-482. doi:10.1145/3017680.3017739

Popper, K. R. (1972). *Objective knowledge: An evolutionary approach*. Oxford, UK: Clarendon Press.

Posavac, E. J. (2016). *Program evaluation: Methods and case studies*. Abingdon, Oxon, UK: Routledge.

Powerll, J. (1988). Reducing teacher control. In D. Boud (Ed.), *Developing student autonomy in learning* (Second Edition ed., pp. 109-119). London, UK: Taylor & Francis

Praslova, L. (2010). Adaptation of kirkpatrick's four level model of training criteria to assessment of learning outcomes and program evaluation in higher education. *Educational Assessment, Evaluation and Accountability, 22*(3), 215-225. doi:10.1007/s11092-010-9098-7

Prøitz, T. S. (2010). Learning outcomes: What are they? Who defines them? When and where are they defined? *Educational Assessment, Evaluation and Accountability, 22*(2), 119-137. doi:10.1007/s11092-010-9097-8

Prus, R. C. (1996). *Symbolic interaction and ethnographic research: Intersubjectivity and the study of human lived experience*. Albany, NY: State University of New York.

Puentedura, R. (2006). Transformation, technology, and education. *Web Page Accessed 3rd April 2018.* Retrieved from http://hippasus.com/resources/tte/

Qian, Y., Hambrusch, S., Yadav, A., & Gretter, S. (2018). Who needs what: Recommendations for designing effective online professional development for computer science teachers. *Journal of Research on Technology in Education, 50*(2), 164-181. doi:10.1080/15391523.2018.1433565

Quille, K., Faherty, R., Bergin, S., & Becker, B. A. (2018). Second level computer science: The irish k-12 journey begins. *18th International Conference on Computing Education Research; Koli, Finland; ACM*, 22. doi:10.1145/3279720.3279742

Ratka-Krüger, P., Wölber, J., Blank, J., Holst, K., Hörmeyer, I., & Vögele, E. (2018). Masteronline periodontology and implant therapy—revisited after seven years: A case study of the structures and outcomes in a blended learning cpd. *European Journal of Dental Education, 22*(1), e7-e13. doi:10.1111/eje.12249

Ravitz, J., Stephenson, C., Parker, K., & Blazevski, J. (2017). Early lessons from evaluation of computer science teacher professional development in google's cs4hs program. *ACM Transactions Computing Education, 17*(4), 1-16. doi:10.1145/3077617

Reding, T. E., & Dorn, B. (2017). Understanding the "teacher experience" in primary and secondary cs professional development. *2017 Conference on International Computing Education Research; Tacoma, WA; ACM*, 155-163. doi:10.1145/3105726.3106185

Reio, T. G., Rocco, T. S., Smith, D. H., & Chang, E. (2017). A critique of kirkpatrick's evaluation model. *New Horizons in Adult Education and Human Resource Development, 29*(2), 35-53. doi:10.1002/nha3.20178

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., . . . Silverman, B. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60-67. doi:10.1145/1592761.1592779

Resnick, M., & Robinson, K. (2017). *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. Cambridge, MA: MIT Press.

Richardson, C., & Mishra, P. (2018). Learning environments that support student creativity: Developing the scale. *Thinking Skills and Creativity, 27*, 45-54. doi:10.1016/j.tsc.2017.11.004

Ricoeur, P. (1974). *The conflict of interpretations* (R. Sweeney, Trans.). London, UK: Continuum.

Ridgway, J., & Passey, D. (1991). A constructivist approach to educational computing. *Australian Educational Computing, 6*(2), 4-9. doi:http://acce.edu.au/journal/6/2/constructivist-approach-educational-computing

Riga, F., Winterbottom, M., Harris, E., & Newby, L. (2017). Inquiry-based science education. In K. S. Taber & B. Akpan (Eds.), *Science education: An international course companion* (pp. 247-261). Rotterdam, NL: SensePublishers.

Roberts, E. (2018). Teaching computer science in a time of opportunities and challenges. *2018 Technical Symposium on Computer Science Education; Baltimore, MD, USA; ACM*, 136. doi:10.1145/3159450.3166086

Roberts, M., Prottsman, K., & Gray, J. (2018). Priming the pump: Reflections on training k-5 teachers in computer science. *2018 Technical Symposium on Computer Science Education; Baltimore, MD, USA; ACM*, 723-728. doi:10.1145/3159450.3159560

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education, 13*(2), 137-172. doi:10.1076/csed.13.2.137.14200

Robinson, K. (2017). *Out of our minds - the power of being creative*. Chichester, UK: John Wiley & Sons, Ltd.

Roesken-Winter, B., Schüler, S., Stahnke, R., & Blömeke, S. (2015). Effective cpd on a large scale: Examining the development of multipliers. *ZDM, 47*(1), 13-25. doi:10.1007/s11858-014-0644-5

Roessger, K. M. (2015). But does it work? Reflective activities, learning outcomes and instrumental learning in continuing professional development. *Journal of Education and Work, 28*(1), 83-105. doi:10.1080/13639080.2013.805186

Rogers, E. M. (1995). *Diffusion of innovations* (4th ed.). New York, NY: Free Press.

Rogers, P. J., & Weiss, C. H. (2007). Theory-based evaluation: Reflections ten years on: Theory-based evaluation: Past, present, and future. *New Directions for Evaluation, 2007*(114), 63-81. doi:10.1002/ev.1086

Rogoff, B. (2003). *The cultural nature of human development*. New York, NY: Oxford University Press, Inc.

Rose, D., & Sullivan, O. (1996). Introducing data analysis for social scientists In. Buckingham, UK: Open University Press.

Rossi, P. H., Lipsey, M. W., & Freeman, H. E. (2004). *Evaluation: A systematic approach* (7th ed.). Thousand Oaks, CA: Sage Publications Inc.

Rowe, F. (2014). What literature review is not: Diversity, boundaries and recommendations. *European Journal of Information Systems, 23*(3), 241-255. doi:10.1057/ejis.2014.7

Rubin, C. (2018). The global search for education: Teachers' top tips for designing meaningful pd programs. *The Education Digest, 83*(5), 10-16. doi:https://blog.edmodo.com/2017/10/20/the-global-search-for-education-teachers-top-tips-for-designing-meaningful-pd-programs/

Ruiz, J., & Snoeck, M. (2018). Adapting kirkpatrick's evaluation model to technology enhanced learning. *21st International Conference on Model Driven Engineering Languages and Systems; Copenhagen, Denmark; ACM/IEEE*, 135-142. doi:10.1145/3270112.3270114

Ryan, G. W., & Bernard, H. R. (2000). Data management and analysis methods. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of qualitative research* (pp. 769-802). Thousand Oaks, CA: SAGE Publications, Inc.

Ryoo, J., Goode, J., & Margolis, J. (2015). It takes a village: Supporting inquiry-and equity-oriented computer science pedagogy through a professional learning community. *Computer Science Education, 25*(4), 351-370. doi:10.1080/08993408.2015.1130952

Sachs, J. (2011). *Rethinking educational practice through reflexive inquiry*. Cham, CH: Springer International Publishing Switzerland.

Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2011). Teaching programming in secondary school: A pedagogical content knowledge perspective. *Informatics in Education, 10*(1), 73-88. doi:https://eric.ed.gov/?id=EJ1064282

Sagan, C. (1980). *Cosmos*. New York, NY: The Random House Publishing Group.

Salomon, G., & Perkins, D. N. (1987). Transfer of cognitive skills from programming: When and how? *Journal of Educational Computing Research, 3*(2), 149-169. doi:10.2190/6F4Q-7861-QWA5-8PL1

Samurcay, R. (2013). The concept of variable in programming. Its meaning and use in problem solving by novice programmers. In E. Soloway (Ed.), *Studying the novice programmer* (pp. 161-179). London, UK: Psychology Press.

Sauberschwarz, L., & Weiss, L. (2018). How corporations can win the race against disruptive startups. In C. Linnhoff-Popien, R. Schneide, & M. Zaddach (Eds.), *Digital marketplaces unleashed* (pp. 155-167). Berlin, Germany: Springer-Verlag GmbH.

Saxton, E., Burns, R., Holveck, S., Kelley, S., Prince, D., Rigelman, N., & Skinner, E. A. (2014). A common measurement system for k-12 stem education: Adopting an educational evaluation methodology that elevates theoretical foundations and systems thinking. *Studies in Educational Evaluation, 40*, 18-35. doi:https://doi.org/10.1016/j.stueduc.2013.11.005

Scales, P., Pickering, J., & Senior, L. (2011). *Continuing professional development in the lifelong learning sector*. New York, NY: McGraw-Hill Education.

Scapp, R. (2003). *Teaching values: Critical perspectives on education, politics, and culture*. London, UK: RoutledgeFalmer.

Schalock, R. L., & Bonham, G. S. (2003). Measuring outcomes and managing for results. *Evaluation and Program Planning, 26*(3), 229-235. doi:10.1016/S0149-7189(03)00027-2

Schuck, S., Aubusson, P., Burden, K., & Brindley, S. (2018). *Uncertainty in teacher education futures: Scenarios, politics and stem*. Singapore, SG: Springer Nature Singapore Pte Ltd.

Schwandt, T. A. (2000). Three epistemolological stances for qualitative inquiry - interpretivistm, hermenutics, and social constructionism. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of qualitative research* (2nd ed., pp. 189-213). Thousand Oaks, CA: SAGE Publications, Inc.

Scriven, M. (1974). *Evaluation in education*. Richmond, CA: Mc Cutchan Publishing.

Scriven, M. (1991a). *Evaluation thesaurus* (4th ed.). Newbury Park, CA: SAGE Publications, Inc.

Scriven, M. (1991b). Prose and cons about goal-free evaluation. *Evaluation Practice, 12*(1), 55-62. doi:10.1177/109821409101200108

Sentance, S. (2018a). Introduction to part 1 why teach computer science in school. In S. Sentance, E. Barendsen, & C. Schulte (Eds.), *Computer science education. Perspectives on teaching and learning in school* (pp. 3-5). London, UK: Bloomsbury Academic.

Sentance, S. (2018b). Introduction to part 2 aspects of teaching and learning computer science in school. In S. Sentance, E. Barendsen, & C. Schulte (Eds.), *Computer science education. Perspectives on teaching and learning in school* (pp. 75-77). London, UK: Bloomsbury Academic.

Sentance, S. (2018c). Recent developments in computer science educaion research. *Web Page Accessed 19th October 2018.* Retrieved from https://community.computingatschool.org.uk/files/10058/original.pdf

Sentance, S., Barendsen, E., & Schulte, C. (2018). *Computer science education: Perspectives on teaching and learning in school*. London, UK: Bloomsbury Academic.

Sentance, S., & Csizmadia, A. (2017a). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies, 22*, 469–495. doi:10.1007/s10639-016-9482-0

Sentance, S., & Csizmadia, A. (2017b). Professional recognition matters: Certification for in-service computer science teachers *2017 Technical Symposium on Computer Science Education; Seattle, Washington, USA; ACM*, 537-542. doi:10.1145/3017680.3017752

Sentance, S., Dorling, M., & McNicol, A. (2013). Computer science in secondary schools in the uk: Ways to empower teachers. In I. Diethelm & R. T. Mittermeir (Eds.), *Informatics in schools. Sustainable informatics education for pupils of all ages. Issep 2013. Lecture notes in computer science* (Vol. 7780, pp. 15-30). Berlin, Heidelberg: Springer.

Sentance, S., Dorling, M., McNicol, A., & Crick, T. (2012). Grand challenges for the uk: Upskilling teachers to teach computer science within the secondary curriculum. *7th Workshop in Primary and Secondary Computing Education; Hamburg, Germany; ACM*, 82-85. doi:10.1145/2481449.2481469

Sentance, S., & Humphreys, S. (2018). Understanding professional learning for computing teachers from the perspective of situated learning. *Computer Science Education, 2*(1), 1-26. doi:10.1080/08993408.2018.1525233

Sentance, S., Humphreys, S., & Dorling, M. (2014). The network of teaching excellence in computer science and master teachers. *9th Workshop in Primary and Secondary Computing Education; Berlin, Germany; ACM*, 80-88. doi:10.1145/2670757.2670789

Sentance, S., Sinclair, J., Simmons, C., & Csizmadia, A. (2016). Teacher research projects in computing. *11th Workshop in Primary and Secondary Computing Education; Münster, Germany; ACM*, 110-111. doi:10.1145/2978249.2978271

Sentance, S., Sinclair, J., Simmons, C., & Csizmadia, A. (2018). Classroom-based research projects for computing teachers: Facilitating professional learning. *ACM Transaction on Computing Education, 18*(3), 1-26. doi:10.1145/3171129

Sergis, S., Sampson, D. G., & Pelliccione, L. (2018). Investigating the impact of flipped classroom on students' learning experiences: A self-determination theory approach. *Computers in Human Behavior, 78*, 368-378. doi:10.1016/j.chb.2017.08.011

Shadish, W. R., Cook, T. D., & Leviton, L. C. (1991). *Foundations of program evaluation: Theories of practice*. Newbury Park, CA: SAGE Publications, Inc

Shah, N., Lewis, C. M., Caires, R., Khan, N., Qureshi, A., Ehsanipour, D., & Gupta, N. (2013). Building equitable computer science classrooms: Elements of a teaching approach. *44th Technical Symposium on Computer Science Education; Denver, CO, USA; ACM*, 263-268. doi:https://dl.acm.org/citation.cfm?id=2445276

Shaha, S., Lewis, V., O'Donnell, T., & Brown, D. (2004). Evaluating professional development in an approach to verifying program impact on teachers and students. *Journal of Research in Professional Learning, The National Staff Development Council, pp. 1-18, Web Page Accessed 18 September 2018,* https://plsclasses.com/wp-content/uploads/2013/03/EvalPDJRPL2004Finalarticle.pdf.

Shen, N., Yufe, S., Saadatfard, O., Sockalingam, S., & Wiljer, D. (2017). Rebooting kirkpatrick: Integrating information system theory into the evaluation of web-based continuing professional development interventions for interprofessional education. *Journal of Continuing Education in the Health Professions, 37*(2), 137-146. doi:10.1097/CEH.0000000000000154

Sitzmann, T., Brown, K. G., Casper, W. J., Ely, K., & Zimmerman, R. D. (2008). A review and meta-analysis of the nomological network of trainee reactions. *Journal of Applied Psychology, 93*(2), 280-295. doi:10.1037/0021-9010.93.2.280

Skinner, B. F. (1961). Why we need teaching machines. *Harvard Educational Review, 3*, 377-398.

Skinner, B. F. (1968). *The technology of teaching*. East Norwalk, CT: Appleton-Century-Crofts.

Slavin, R. E. (1987). Taking the mystery out of mastery: A response to guskey, anderson, and burns. *Review of Educational Research, 57*(2), 231-235. doi:10.3102/00346543057002231

Smerdon, B. A., Burkam, D. T., & Lee, V. E. (1999). Access to constructivist and didactic teaching: Who gets it? Where is it practiced? *Teachers College Record, 101*(1), 5-34. doi:https://eric.ed.gov/?id=EJ598238

Smidt, A., Balandin, S., Sigafoos, J., & Reed, V. A. (2009). The kirkpatrick model: A useful tool for evaluating training outcomes. *Journal of Intellectual & Developmental Disability, 34*(3), 266-274. doi:10.1080/13668250903093125

Spradley, J. P. (1979). *The ethnographic interview*. Belmont, CA: Wadsworth Publishing.

Stadler, F. (2004). *Induction and deduction in the sciences*. Dordrecht, NL: Kluwer Academic Publishers.

Stake, R. E. (1967). *The countenance of educational evaluation*. Retrieved from Center for Instructional Research and Curriculum Evaluation, University of Illinois, ILL, 1-18. doi: 10.1.1.543.5561&rep=rep1&type=pdf:

Stake, R. E. (1983a). *Evaluation models. Evaluation in education and human services*. Dordrecht, NL: Springer.

Stake, R. E. (1983b). Program evaluation, particulary responsive evaluation. In G. F. Madaus, M. Scriven, & D. L. Stufflebeam (Eds.), *Evaluation models: Viewpoints on educational and human services evaluation* (pp. 287-311). Boston, MA: Kluwer-Nijhoff Publishing.

Stake, R. E. (2000). Strategies of inquiry. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of qualitative research* (2nd ed., pp. 435-455). Thousand Oaks, CA: Sage Publications, Inc.

Stame, N. (2004). Theory-based evaluation and types of complexity. *Evaluation, 10*(1), 58-76. doi:10.1177/1356389004043135

Stanford, T., Wiburg, K., Chamberlin, B., Trujillo, K., & Parra, J. L. (2016). Transforming mathematics teaching through games and inquiry. In M. Neiss, S. Driskell, & K. Hollerbrands (Eds.), *K-12 stem education: Breakthroughs in research and practice* (pp. 279-305). Hershey, PA: IGI Global.

Stebbins, R. A. (2001). *Exploratory research in the social sciences*. Thousand Oaks, CA: SAGE Publications, Inc.

Stern, P. N., & Harris, C. C. (1985). Women's health and the self-care paradox. A model to guide self-care readiness. *Health Care for Women International, 6*(1-3), 151-163. doi:10.1080/07399338509515689

Stiller, E. (2009). Teaching programming using bricolage. *Journal of Computing Sciences in Colleges, 24*(6), 35-42. doi:https://dl.acm.org/citation.cfm?id=1530004

Stoecker, R. (1991). Evaluating and rethinking the case study. *The Sociological Review, 39*(1), 88-112. doi:10.1111/j.1467-954X.1991.tb02970.x

Strauss, A. L., & Corbin, J. M. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Thousand Oaks, CA: SAGE Publiations Inc.

Stronge, J. H. (2018). *Qualities of effective teachers, 3rd edition* (3rd ed.). Alexandria, VA: Association for Supervision & Curriculum Development.

Stufflebeam, D. L. (1966). A depth study of the evaluation requirement. *Theory Into Practice, 5*(3), 121-133. doi:10.1080/00405846609542011

Stufflebeam, D. L. (1983). *Evaluation models. Evaluation in education and human services* Dordrecht, NL: Springer.

Stufflebeam, D. L., & Webster, W. J. (1983). An analysis of alterative approaches to evaluation. In G. F. Madaus, M. Scriven, & D. L. Stufflebeam (Eds.), *Evaluation models: Viewpoints on educational and human services evaluation* (pp. 23-45). Boston, MA: Kluwer-Nijhoff Publishing.

Sugrue, C., & Mertkan, S. (2017). Professional responsibility, accountability and performativity among teachers: The leavening influence of cpd? *Teachers and Teaching, 23*(2), 171-190. doi:10.1080/13540602.2016.1203771

Sullivan, K., Marshall, K., & Tangney, B. (2015). Learning circles: A collaborative technology-mediated peer-teaching workshop. *Journal of Information Technology Education: Innovations in Practice, 14*, 63-83. doi:10.28945/2176

Symonds, P. M. (1924). On the loss of reliability in ratings due to coarseness of the scale. *Journal of Experimental Psychology, 7*(6), 456-461. doi:10.1037/h0074469

Tamkin, P., Yarnall, J., & Kerrin, M. (2002). *Kirkpatrick and beyond: A review of models of training evaluation*. Brighton, UK: Institute for Employment Studies.

Tangney, B., Boran, I., Knox, T., & Bray, A. (2018). Constructionist stem activities using the bridge21 model. *2018 Constructionism; Vilnius, Lithuania; Research Council of Lithuania*, 456-467.

Tangney, B., Oldham, E., Conneely, C., Barrett, S., & Lawlor, J. (2010). Pedagogy and processes for a computer programming outreach workshop—the bridge to college model. *IEEE Transactions on Education, 53*(1), 53-60. doi:10.1109/TE.2009.2023210

Taylor, J. L. (2018). *Independent learning*. London, UK: Bloomsbury Publishing Plc.

TCD. (2014). Trinity college launches 21st century teaching & learning course. *School of Education in association with School of Computer Science & Statistics (SCSS), Centre for Research in IT in Education (CRITE) and Trinity Access Programme (TAP), Web Page Accessed 1st October 2014.* Retrieved from https://www.tcd.ie/news_events/articles/trinity-college-launches-21st-century-teaching-learning-course/5017

TCD. (2017). *Postgraduate certificate in 21st century teaching and learning - handbook 2017/2018*. Retrieved from Dublin: https://www.tcd.ie/Education/programmes/certificate-21Century-Teaching-Learning/assets/PGCertHandbook17-18_FINAL.pdf

TCD. (2018). 21st century teaching and learning (p. Grad. Cert.). *Web Page Accessed 1st July 2018.* Retrieved from https://www.tcd.ie/courses/postgraduate/az/course.php?id=DPCED-TCTL-1P09

Teague, D., & Roe, P. (2008). Collaborative learning: Towards a solution for novice programmers. *10th Australasian Computing Education; Wollongong, NSW, Australia; ACM, 78*, 147-153. doi:https://dl.acm.org/citation.cfm?id=1379264

Tedre, M., Simon, & Malmi, L. (2018). Changing aims of computing education: A historical survey. *Computer Science Education, 28*(2), 158-186. doi:10.1080/08993408.2018.1486624

Tenenberg, J., Roth, W.-M., Chinn, D., Jornet, A., Socha, D., & Walter, S. (2018). More than the code: Learning rules of rejection in writing programs. *Communications of the ACM, 61*(5), 66-71. doi:10.1145/3132699

The Joint Committee on Standards for Educational Evaluation. (1994). *The program evaluation standards: How to assess evaluations of educational programs* (2nd ed.). Thousand Oaks, CA: SAGE Publications, Inc.

Thibaut, L., Knipprath, H., Dehaene, W., & Depaepe, F. (2018). The influence of teachers' attitudes and school context on instructional practices in integrated stem education. *Teaching and Teacher Education, 71*, 190-205. doi:10.1016/j.tate.2017.12.014

Thietart, R.-A., & Wauchope, S. (2001). *Doing management research: A comprehensive guide*. London, UK: SAGE Publications, Inc.

Thompson, G. (2017). Coding comes of age: Coding is gradually making its way from club to curriculum, thanks largely to the nationwide science, technology, engineering and mathematics phenomenon embraced by so many american schools. *THE Journal (Technological Horizons In Education), 44*(1), 1-3. doi:thejournal.com/articles/2017/02/02/coding-comes-of-age.aspx

Thoroddsen, A., & Ehnfors, M. (2007). Putting policy into practice: Pre-and posttests of implementing standardized languages for nursing documentation. *Journal of Clinical Nursing, 16*(10), 1826-1838. doi:10.1111/j.1365-2702.2007.01836.x

Thurlings, M., & den Brok, P. (2017). Learning outcomes of teacher professional development activities: A meta-study. *Educational Review, 69*(5), 554-576. doi:10.1080/00131911.2017.1281226

Timmermans, S., & Tavory, I. (2012). Theory construction in qualitative research from grounded theory to abductive analysis. *Sociological Theory, 30*(3), 167-186. doi:10.1177/0735275112457914

Tom, M. (2015). Five c framework: A student-centered approach for teaching programming courses to students with diverse disciplinary background. *Journal of Learning Design, 8*(1), 21-27. doi:10.5204/jld.v8i1.193

Tondeur, J., van Braak, J., Ertmer, P. A., & Ottenbreit-Leftwich, A. (2017). Understanding the relationship between teachers' pedagogical beliefs and technology use in education: A systematic review of qualitative evidence. *Educational Technology Research and Development, 65*(3), 555-575. doi:10.1007/s11423-016-9481-2

Tondeur, J., Van Braak, J., Sang, G., Voogt, J., Fisser, P., & Ottenbreit-Leftwich, A. (2012). Preparing pre-service teachers to integrate technology in education: A synthesis of qualitative evidence. *Computers & Education, 59*(1), 134-144. doi:10.1016/j.compedu.2011.10.009

Trafimow, D. (2003). Hypothesis testing and theory evaluation at the boundaries: Surprising insights from bayes's theorem. *Psychological review, 110*(3), 526. doi:10.1037/0033-295X.110.3.526

Trauth, E. M., & Jessup, L. M. (2000). Understanding computer-mediated discussions: Positivist and interpretive analyses of group support system use. *MIS Quarterly, 24*(1), 43-79 doi:10.2307/3250979

Turnbow, D., & Roth, A. (2017). Engaging learners online: Using instructional design practices to create interactive tutorials. In T. Maddison & M. Kumaran (Eds.), *Distributed learning - pedagogy and technology in online information literacy instruction* (pp. 123-134). Cambridge, UK: Chandos Publishing.

Turner, J. C., Christensen, A., Kackar-Cam, H. Z., Fulmer, S. M., & Trucano, M. (2018). The development of professional learning communities and their teacher leaders: An activity systems analysis. *Journal of the Learning Sciences, 27*(1), 49-88. doi:10.1080/10508406.2017.1381962

Tytler, R., Symington, D., Williams, G., & White, P. (2018). Enlivening stem education through school-community partnerships. In R. Jorgensen & K. Larkin (Eds.), *Stem education in the junior secondary* (pp. 249-272). Singapore, SG: Springer Nature Singapore Pte Ltd.

UCD. (2017). School of education and ucd school of computer science welcome the first group of ucd microsoft award holders to the new pdes (computational thinking) programme. University college dublin. *Web Page Accessed 12th September 2017.* Retrieved from https://www.ucd.ie/education/newsandevents/allnewsevents/2017news/september2017/

UL. (2017). Computing for schools. University of limerick. *Web Page Accessed 12th September 2017.* Retrieved from https://www.lero.ie/

Ulloa, M. (1980). Teaching and learning computer programming: A survey of student problems, teaching methods, and automated instructional tools *ACM Special Interest*

Group on Computer Science Education Bulletin, 12(2), 48-64. doi:10.1145/989253.989263

UNESCO. (2011). Unesco ict competency framework for teachers. *Web Page Accessed 21st April 2018.* Retrieved from http://unesdoc.unesco.org/images/0021/002134/213475e.pdf

Vagias, W. M. (2006). Likert-type scale response anchors. Clemson international institute for tourism & research development, department of parks, recreation and tourism management. *Web Page Accessed 4th September 2018.* Retrieved from https://www.peru.edu/oira/wp-content/uploads/sites/65/2016/09/Likert-Scale-Examples.pdf

Van Den Akker, J., Gravemeijer, K., McKenney, S., & Nieveen, N. (2006). *Educational design research*. Abingdon, Oxon, UK: Routledge.

Van Driel, J. H., Meirink, J. A., Van Veen, K., & Zwart, R. C. (2012). Current trends and missing links in studies on teacher professional development in science education: A review of design features and quality of research. *Studies in Science Education, 48*(2), 129-160. doi:10.1080/03057267.2012.738020

Van Maanen, J. (1988). *Tales of the field: On writing ethnography*. Chicago, ILL: University of Chicago Press.

Van Merriënboer, J. J. G., & Kirschner, P. A. (2017). *Ten steps to complex learning: A systematic approach to four-component instructional design* (3rd ed.). New York, NY: Routledge.

Varadharajan, M., Buchanan, J., & Schuck, S. (2018). Changing course: The paradox of the career change student-teacher. *Professional Development in Education, 44*(5), 738-749. doi:10.1080/19415257.2017.1423369

Veen, W. (1993). How teachers use computers in instructional practice—four case studies in a dutch secondary school. *Computers & Education, 21*(1-2), 1-8. doi:10.1016/0360-1315(93)90041-G

Vennix, J., den Brok, P., & Taconis, R. (2017). Perceptions of stem-based outreach learning activities in secondary education. *Learning Environments Research, 20*(1), 21-46. doi:10.1007/s10984-016-9217-6

Vizcaíno, A., Contreras, J., Favela, J., & Prieto, M. (2000). An adaptive, collaborative environment to develop good habits in programming. In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *Intelligent tutoring systems* (pp. 262-271). Berlin, Heidelberg: Springer-Verlag.

Von Glasersfeld, E. (1989). Cognition, construction of knowledge, and teaching. *Synthese, 80*(1), 121-140. doi:10.1007/BF00869951

Voogt, J., Erstad, O., Dede, C., & Mishra, P. (2013). Challenges to learning and schooling in the digital networked world of the 21st century. *Journal of Computer Assisted Learning, 29*(5), 403-413. doi:10.1111/jcal.12029

Voogt, J., Laferrière, T., Breuleux, A., Itow, R. C., Hickey, D. T., & McKenney, S. (2015). Collaborative design as a form of professional development. *Instructional Science, 43*(2), 259-282. doi:10.1007/s11251-014-9340-7

Vygotsky, L. S. (1978). *Mind in society : The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Walker, H. M. (2018). *Teaching computing: A practitioner's perspective*. Boca Raton, FL: CRC Press.

Walsham, G. (1993). *Interpreting information systems in organizations*. Chichester, UK: John Wiley & Sons.

Walsham, G. (1995a). The emergence of interpretivism in is research. *Information Systems Research, 6*(4), 376-394. doi:10.1287/isre.6.4.376

Walsham, G. (1995b). Interpretive case studies in is research: Nature and method. *European Journal of Information Systems, 4*(2), 74-81. doi:10.1057/ejis.1995.9

Wang, M.-L., & Chang, A.-C. (2019). Student learning outcome assessment for an information organization curriculum based on the kirkpatrick framework. *Qualitative and Quantitative Methods in Libraries, 6*(4), 527-544. doi:http://www.qqml-journal.net/index.php/qqml/article/view/439

Warner, J. R., Fletcher, C. L., Monroe, W., & Garbrecht, L. S. (2018). Growing the high school cs teacher workforce: Predictors of success in achieving cs certification. *49th Technical Symposium on Computer Science Education; Baltimore, MA, USA; ACM*, 1096-1096. doi:10.1145/3159450.3162278

Watkins, R., Leigh, D., Foshay, R., & Kaufman, R. (1998). Kirkpatrick plus: Evaluation and continuous improvement with a community focus. *Educational Technology Research and Development, 46*(4), 90-96. doi:10.1007/BF02299676

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer science in k-12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies, 22*(2), 445-468. doi:10.1007/s10639-016-9493-x

Webb, N. M., Ender, P., & Lewis, S. (1986). Problem-solving strategies and group processes in small groups learning computer programming. *American Educational Research Journal, 23*(2), 243-261. doi:10.3102/00028312023002243

Weimer, M. (2013). *Learner-centered teaching: Five key changes to practice*. Hoboken, NJ: John Wiley & Sons, Inc.

Weiss, C. H. (1997). Theory-based evaluation: Past, present, and future. *New Directions for Evaluation, 1997*(76), 41-55. doi:10.1002/ev.1086

Weiss, C. H. (1998). Have we learned anything new about the use of evaluation? *American Journal of Evaluation, 19*(1), 21-33. doi:10.1177/109821409801900103

Weiss, C. H. (2000). Which links in which theories shall we evaluate? *New Directions for Evaluation, 2000*(87), 35-45. doi:10.1002/ev.1180

Wells, A. S., Hirshberg, D., Lipton, M., & Oakes, J. (2002). Bounding the case within its context: A constructivist approach to studying detracking reform. In M. B. Miles & A. M. Huberman (Eds.), *The qualitative researcher's companion: Classic and contemporary readings*. Thousand Oaks, CA: SAGE Publications, Inc.

Weston, C., McAlpine, L., & Bordonaro, T. (1995). A model for understanding formative evaluation in instructional design. *Educational Technology Research and Development, 43*(3), 29-48. doi:10.1007/BF02300454

Whitley, E., & Ball, J. (2002). Statistics review 6: Nonparametric methods. *Critical Care, 6*(6), 509-513. doi:https://www.ncbi.nlm.nih.gov/pmc/articles/PMC153434/

Whittemore, R., & Knafl, K. (2005). The integrative review: Updated methodology. *Journal of Advanced Nursing, 52*(5), 546-553. doi:10.1111/j.1365-2648.2005.03621.x

Wholey, J. S. (1987). Evaluability assessment: Developing program theory. *New Directions for Evaluation, 1987*(33), 77-92. doi:10.1002/ev.1447

Wholey, J. S. (2015). Exploratory evaluation. In K. E. Newcomer, H. P. Hatry, & J. S. Wholey (Eds.), *Handbook of practical program evaluation* (4th ed., pp. 105-106). Hoboken, NJ: John Wiley & Sons, Inc.

Wickham, C., Girvan, C., & Tangney, B. (2016). Constructionism and microworlds as part of a 21st century learning activity to impact student engagement and confidence in physics. In A. Sipitakiat & N. Tutiyapheugprasert (Eds.), *Constructionism* (pp. 34-43). Bankok, Thailand: Suksapatana Foundation.

Wilcoxon, F. (1945). Individual comparions by ranking methods. *Biometrics, 1*, 80-83. doi:10.2307/3001968

Williamson, B. (2016). Political computational thinking: Policy networks, digital governance and 'learning to code'. *Critical Policy Studies, 10*(1), 39-58. doi:10.1080/19460171.2015.1052003

Wolcott, H. F. (1994). *Transforming qualitative data, description, analysis and interpretation*. Thousand Oaks, CA: SAGE Publications, Inc.

Wood, K., Jaidin, H., Jawawi, R., Perera, J., Salleh, S., Shahrill, M., & Sithamparam, S. (2017). How and what teachers learn from collaborative professional development. *International Journal for Lesson and Learning Studies, 6*(2), 151-168. doi:10.1108/IJLLS-09-2016-0028

Woodworth, R. S., & Thorndike, E. (1901). The influence of improvement in one mental function upon the efficiency of other functions. *Psychological review, 8*(3), 247. doi:10.1037/h0074898

Worthen, B. R. (1968). Toward a taxonomy of evaluation designs *Educational Technology, 8*(15), 3-9. doi:http://www.jstor.org/stable/44422359

Worthen, B. R., & Sanders, J. R. (1973). *Educational evaluation: Theory and practice*. Belmont, CA: Wadsworth Publishing Company.

Yadav, A., & Beckerman, J. L. (2018). Implementing case studies in a plant pathology course: Impact on student learning and engagement. In K. Barbarick (Ed.), *A collection of case studies* (pp. 100-105). Madison, WI: American Society of Agronomy.

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2017). Expanding computer science education in schools: Understanding teacher experiences and challenges *Computer Science Education, 26*(4), 235-254. doi:10.1080/08993408.2016.1257418

Yeh, R. T., Good, D. I., & Musser, D. R. (1973). New directions in teaching the fundamentals of computer science- discrete structures and computational analysis. *ACM Special Interest Group on Computer Science Education Bulletin, 5*(1), 60-67. doi:10.1145/800010.808080

Yildiz Durak, H. (2018). Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences. *Journal of Computer Assisted Learning, 1*(1-13). doi:10.1111/jcal.12281

Yin, R. K. (2003). *Case study research. Design and methods*. Thousand Oaks, CA: Sage Publications, Inc.

Zambak, V. S., Alston, D. M., Marshall, J. C., & Tyminski, A. M. (2017). Convincing science teachers for inquiry-based instruction: Guskey's staff development model revisited. *Science Educator, 25*(2), 108-116. doi:https://search.proquest.com/openview/da92642ae02bb65f8ee637e66e728f41/1?pq-origsite=gscholar&cbl=26548

Zawacki-Richter, O., & Latchem, C. (2018). Exploring four decades of research in computers & education. *Computers & Education, 122*, 136-152. doi:10.1016/j.compedu.2018.04.001

Zeidler, D. L. (2016). Stem education: A deficit framework for the twenty first century? A sociocultural socioscientific response. *Cultural Studies of Science Education, 11*(1), 11-26.

Zendler, A. (2018). Instructional methods in computing education judged by computer science teachers and educational experts. *IT-Information Technology, 60*(2), 79-90. doi:10.1515/itit-2017-0015

9 Appendix

9.1 CS CPD Participant Consent Sheet

Teacher Computing Workshop – Ethics Consent Sheet

Please read, and then sign the following if you wish to participate in the workshop research process. You are under no obligation to participate in workshop research process, and can opt in / out of at any stage of the process.

Data Protection: I agree to Trinity College, University of Dublin storing of any personal data relating to me which results from this project. I agree to the processing of such data for purposes connected with the research project as outlined below.

- My participation is voluntary, and I may withdraw at any time and for any reason without penalty from the workshop research process and related data gathering.
- All my data (audio / video / photographs / text) will be treated with full confidentiality and stored securely so that, in the event that any data is published or used for promotional purposes, my data will not be identified as mine, nor identify my school or students.
- Where observational tools are used for data gathering purposes, the research team must ask for my *explicit verbal* consent to be observed, and explain why.
- When completing surveys, I can omit questions I do not wish to answer. *The questionnaires are administered online (use video displays). If you or anyone in your family has a history of epilepsy, please be aware you are proceeding at your own risk.*
- In the extremely unlikely event that illicit activity is reported during the study, the research team will be obliged to report it to appropriate authorities.

Consent: I have been provided with an information and consent sheet detailing how my data will be processed and how I can contact the research team. The research team will also provide a research debrief at the end of the workshop. By signing this consent form I confirm that I am over 18, have read and understood the contents of the information and consent forms, ***and give permission to be contacted by the team***

PRINTED TEACHER NAME: ___________________________***Signature:***____________________

Signature of Researcher (TCD): ______________________***Date:***________________________

9.2 CS CPD Participant Information Sheet

Teacher Computing Workshop – Ethics Information Sheet

You are invited to participate in workshop research which aims to explore your experience of the Bridge21 learning model, in practice. This project is developed in partnership with Google and moderated by Professor Brendan Tangney (Academic Director, Centre for Research into IT in Education (CRITE) and Bridge21), and Dr Jake Byrne (Workshop Programme Director). Lorraine Fisher (email address) is a PhD student conducting research in this area, mentored by the above advisors.

Bridge21 Programme: www.bridge21.ie

For those new to Bridge21, the programme is committed to developing a model of teaching and learning suited to the needs of students in the 21st century; aligned with the new Junior Cycle. The Bridge21 programme has been running in schools since 2010 and proven to be successful in improving students' attitudes towards learning.

Research Project Overview

A longitudinal research project is underway to evaluate the Bridge21 model in the context of teaching computing in second level schools. The researcher (Lorraine Fisher - (email address) *aims to work with educational professionals to understand the effectiveness of the model as a teaching method, and for developing student skills in the areas of teamwork, technology, creativity and problem solving.*

Workshop Research Procedure

By agreeing to participate in the workshop research process, you are invited to: (1) complete two questionnaires (one prior to the start of the workshop, and one at a future date, on return to teaching); (2) provide the research team with permission to video / observe / photograph technical demonstrations of your work, performed during the workshop; (3) provide consent to fill in workshop evaluation forms; (4) provide consent for the research team to record your voice contribution made within the context of a group presentation or reflection, conducted at the end of the workshop; (5) and finally, provide consent for the research team to *write to you* and ask for your permission to share with us the results of computing project work implemented in the classroom where school ethics has been granted.

9.3 CS CPD Participant Attendance, Ethics, and Data Set

ID	Year	Date	Name	Code	Att.	Ethics
1	2013/2014	16/11/2013	Scratch 1 - Animation	TA21-MOD-3	10	10
2	2013/2014	23/11/2013	Raspberry Pi	TA21-MOD-5	17	13
3	2013/2014	30/11/2013	Computational Thinking	TA21-MOD-2	14	5
4	2013/2014	05/04/2014	Computational Thinking	TA21-MOD-2	5	3
5	2013/2014	26/04/2014	Scratch 1 - Animation	TA21-MOD-3	9	7
6	2013/2014	10/05/2014	Scratch 2 - Game Design	TA21-MOD-4	8	5
7	2013/2014	17/05/2014	Raspberry Pi	TA21-MOD-5	11	5
8	2013/2014	24/05/2014	Python	TA21-MOD-6	8	8
9	2013/2014	10/06/2014	Scratch 1 - Animation	TA21-MOD-2	19	-
10	2013/2014	11/06/2014	Scratch 2 - Game Design	TA21-MOD-3	17	-
11	2013/2014	12/06/2014	Raspberry Pi	TA21-MOD-4	18	-
12	2013/2014	13/06/2014	Python	TA21-MOD-5	16	-
13	2013/2014	01/07/2014	Scratch 1 - Animation	TA21-MOD-2	16	-
14	2013/2014	02/07/2014	Scratch 2 - Game Design	TA21-MOD-3	15	-
15	2013/2014	03/07/2014	Python	TA21-MOD-5	12	-
16	2013/2014	04/07/2014	Raspberry Pi	TA21-MOD-4	14	-
17	2014/2015	29/10/2014	Scratch 1 - Animation	TA21-MOD-3	14	8
18	2014/2015	08/11/2014	Scratch 1 - Animation	TA21-MOD-3	18	10
19	2014/2015	06/12/2014	Raspberry Pi	TA21-MOD-5	9	7
20	2014/2015	17/01/2015	Computational Thinking	TA21-MOD-2	28	20
21	2014/2015	31/01/2015	Scratch 1 - Animation	TA21-MOD-3	23	15
22	2014/2015	07/02/2015	Scratch 2 - Game Design	TA21-MOD-4	15	7
23	2014/2015	18/02/2015	Scratch 1 - Animation	TA21-MOD-3	14	10
24	2014/2015	19/02/2015	Scratch 2 - Game Design	TA21-MOD-4	7	4
25	2014/2015	21/02/2015	Python	TA21-MOD-6	8	4
26	2014/2015	07/03/2015	Raspberry Pi	TA21-MOD-5	12	7
27	2014/2015	31/03/2015	Scratch 1 - Animation	TA21-MOD-3	9	7
28	2014/2015	01/04/2015	Scratch 2 - Game Design	TA21-MOD-4	5	-
29	2014/2015	03/06/2015	Scratch 1 - Animation	TA21-MOD-3	16	-
30	2014/2015	04/06/2015	Scratch 2 - Game Design	TA21-MOD-4	15	-
31	2014/2015	30/06/2015	Python	TA21-MOD-6	11	1

ID	Year	Date	Name	Code	Att.	Ethics
32	2014/2015	01/07/2015	Raspberry Pi	TA21-MOD-5	10	-
33	2015/2016	07/11/2015	Computational Thinking	TA21-MOD-2	13	9
34	2015/2016	14/11/2015	Scratch 2 - Game Design	TA21-MOD-4	9	9
35	2015/2016	21/11/2015	Raspberry Pi	TA21-MOD-5	17	14
36	2015/2016	05/12/2015	Python	TA21-MOD-6	9	6
37	2015/2016	23/01/2016	Scratch 1 - Animation	TA21-MOD-3	36	18
38	2015/2016	23/01/2016	Scratch 1 - Animation	TA21-MOD-3	-	-
39	2015/2016	06/02/2016	Scratch 2 - Game Design	TA21-MOD-4	19	-
40	2015/2016	13/02/2016	Computational Thinking	TA21-MOD-2	16	11
41	2015/2016	13/02/2016	Raspberry Pi	TA21-MOD-5	17	-
42	2015/2016	05/03/2016	Computational Thinking	TA21-MOD-2	17	-
43	2015/2016	05/03/2016	Scratch 1 - Animation	TA21-MOD-3	14	-
44	2015/2016	05/03/2016	Scratch 2 - Game Design	TA21-MOD-4	9	3
45	2015/2016	12/03/2016	Python	TA21-MOD-6	8	-
46	2015/2016	14/06/2016	Scratch 1 - Animation	TA21-MOD-3	6	-
47	2015/2016	15/06/2016	Scratch 2 - Game Design	TA21-MOD-4	4	-
48	2015/2016	04/07/2016	Python	TA21-MOD-6	15	9
49	2015/2016	05/07/2016	Raspberry Pi	TA21-MOD-5	14	-
50	2016/2017	22/10/2016	Computational Thinking	TA21-MOD-2	23	-
51	2016/2017	22/10/2016	Raspberry Pi	TA21-MOD-5	13	-
52	2016/2017	22/10/2016	Scratch 1 - Animation	TA21-MOD-3	78	-
53	2016/2017	22/10/2016	Scratch 1 - Animation	TA21-MOD-3	-	-
54	2016/2017	22/10/2016	Scratch 1 - Animation	TA21-MOD-3	23	-
55	2016/2017	03/12/2016	Scratch 1 - Animation	TA21-MOD-3	29	1
56	2016/2017	03/12/2016	Scratch 2 - Game Design	TA21-MOD-4	44	-
57	2016/2017	11/03/2017	Python	TA21-MOD-6	10	-
58	2016/2017	11/03/2017	Raspberry Pi	TA21-MOD-5	11	-
59	2016/2017	11/03/2017	Computational Thinking	TA21-MOD-2	16	-
60	2016/2017	25/03/2017	Raspberry Pi	TA21-MOD-5	9	-
61	2016/2017	25/03/2017	Scratch 2 - Game Design	TA21-MOD-4	19	-
62	2016/2017	25/03/2017	Computational Thinking	TA21-MOD-2	19	-
63	2016/2017	17/08/2017	Scratch 1 - Animation	TA21-MOD-3	14	-

ID	Year	Date	Name	Code	Att.	Ethics
64	2016/2017	18/08/2017	Scratch 2 - Game Design	TA21-MOD-4	13	-
65	2017/2018	11/11/2017	Computational Thinking	TA21-MOD-2	31	18
66	2017/2018	11/11/2017	Scratch 1 - Animation	TA21-MOD-3	105	-
67	2017/2018	09/12/2017	Scratch 1 - Animation	TA21-MOD-3	27	-
68	2017/2018	09/12/2017	Scratch 2 - Game Design	TA21-MOD-4	15	-
69	2017/2018	03/02/2018	Raspberry Pi	TA21-MOD-5	16	1
70	2017/2018	03/02/2018	Scratch 2 - Game Design	TA21-MOD-5	25	1
71	2017/2018	10/03/2018	Computational Thinking	TA21-MOD-2	13	0
72	2017/2018	10/03/2018	Python	TA21-MOD-6	16	0
Totals					1,215	256

9.4 CS CPD Reaction Variables adapted from Kristiansen (2007)

Reaction Variables			
Original Question	Adapted Question	New Variable	Reported
Training Design			
• The objectives were clearly communicated and met to my satisfaction	• The overall process was clearly communicated and met to my satisfaction	(1) Pedagogical Process	Yes
• The topics were well organized and easy to understand	• The programming activities were well organised and easy to understand	(2) Programming Activities	Yes
• The pace of the training was appropriate for the topics covered	• The pace of the training was appropriate for the topics covered		No
• The level of difficulty of the content was appropriate for me	• The level of difficulty of the content was appropriate for me	(3) Curricula Content	Yes
Instructor			
• The instructor performed well overall	• The facilitator performed well overall	(4) Facilitator Approach	Yes

Reaction Variables continued			
Original Question	Adapted Question	New Variable	Reported
Instructor			
• The instructor is knowledgeable about the subject matter	• The facilitator is knowledgeable about the subject matter		No
• The instructor practiced effective time management	• The facilitator practiced effective time management		No
• The instructor answered my questions to my satisfaction	• The facilitator answered all my questions to my satisfaction		No
Other aspects of the program (if applicable)			
• The facilities were appropriate	• The facilities were appropriate		No
	• This workshop met my professional development expectations	(5) CPD Expectations	Yes
• The audio-visual aids were effective			No
• The exercises / case studies were helpful	• The examples / case studies were helpful	(6) Computing Examples	Yes
• The food was appropriate			No
• The handouts were helpful			No
Overall Rating			
• This training was worth attending	• The Bridge21 computing workshop was worth attending	(7) Workshop Satisfaction	Yes
Training Application			
• I will apply what I learned from this workshop in my teaching			No

Reaction Variables continued			
• I will recommend this training to others within my organisation			No
Original Question	Adapted Question	New Variable	Reported
Comments			
• What topic would you like to spend more or less time on? More/Less			No
• What would improve the programme?			No
• What key learning did you take away from today's Bridge21 computing workshop?		Key Learnings	Yes

9.5 CS CPD Intention Variables adapted from Kristiansen, (2007)

Intention Variables			
Original Question	Adapted Question	New Variable	Reported
Training Application			
	• I intend to use the Bridge21 model in my teaching on return to the classroom	(1) Intention to use Bridge21	Yes
Comments			
• Finally, how might this Bridge21 computing workshop help you in your classroom teaching?		Teaching Practice	Yes

9.6 CS CPD Learning Outcome Variables adapted from TCD (2017)

Learning Outcomes Variables			
Module	Learning outcomes	New Variable	Reported
TA21-MOD-02			
(1) I feel confident in my ability to solve logical problems using a variety of problem solving strategies		(1) Programming Ability	Yes
(2) I have a greater understanding of 'algorithms' and the process of 'algorithmic thinking'		(2) Programming Concepts	Yes

Learning Outcomes Variables continued			
Module	Learning outcomes	New Variable	Reported
(3) I feel confident in my ability to identify a number of outcomes to problems which have more than one possible solution		(3) Programming Processes	Yes
(4) I feel confident in my ability to plan a 21st century learning experience which incorporates logical or algorithmic forms of problem solving		(4) Planning Bridge21 Activities	Yes
(5) I feel confident in my ability to incorporate algorithmic thinking and problem solving into subject teaching activities		(5) Pedagogical Content Knowledge	Yes
(6) I feel confident in my ability to teach logical problem solving through algorithmic thinking using the bridge21 model of 21st century learning		(6) Teaching Programming using Bridge21	Yes
TA21-MOD-03			
(1) I feel confident in my ability to plan and implement scratch animation learning activities according to the Bridge21 model of 21C learning		(1) Planning Bridge21 Activities	Yes
(2) I feel confident in my ability to use scratch animation for teaching computer programming		(2) Pedagogical Content Knowledge	Yes
(3) I have a greater understanding of basic computer programming concepts		(3) Programming Concepts	Yes
(4) I feel confident in my ability to incorporate graphics, sound, and backgrounds into scratch animations		(4) Programming Ability	Yes
(5) I have a greater awareness of scratch programming concepts such as motion and control and their role in animating sprites		(5) Programming Processes	Yes
(6) I feel confident in my ability to teach computer programming through scratch animation using the Bridge21 model of 21C learning		(6) Teaching Programming using Bridge21	Yes

Learning Outcomes Variables continued			
Module	Learning outcomes	New Variable	Reported
TA21-MOD-04			
(1) I feel confident in my ability to plan and implement scratch animation learning activities according to the Bridge21 model of 21C learning		(1) Planning Bridge21 Activities	Yes
(2) I feel confident in my ability to use scratch game design for teaching computer programming		(2) Pedagogical Content Knowledge	Yes
(3) I have a greater understanding of game design computer programming concepts such as events and concurrency		(3) Programming Concepts	Yes
(4) I feel confident in my ability to incorporate jumping, score increments and changing screens or levels into Scratch games		(4) Programming Ability	Yes
(5) I have a greater awareness of scratch programming concepts such as variables and operators and their role in game design		(5) Programming Processes	Yes
(6) I feel confident in my ability to teach computer programming through scratch game design using the Bridge21 model of 21C learning		(6) Teaching Programming using Bridge21	Yes
TA21-MOD-05			
(1) I feel confident in my ability to plan and implement a Raspberry Pi learning activity according to the Bridge21 model of 21C learning		(1) Planning Bridge21 Activities	Yes
(2) I feel confident in my ability to create an electronic circuit and compile the code to interface with embedded systems such as the Raspberry Pi		(2) Content Knowledge	Yes
(3) I have a greater understanding of embedded systems concepts such as inputs, outputs and their use within real world applications		(3) Programming Concepts	Yes
(4) I feel confident in my ability to set up and then operate an embedded system such as Makey-Makey or the Raspberry Pi		(4) Programming Ability	Yes

Learning Outcomes Variables Continued			
Module / Learning outcomes		New Variable	Reported
(5) I have a greater understanding of hardware and software concepts and their role within embedded systems		(5) Programming Processes	Yes
(6) I feel confident in my ability to teach embedded system concepts through the use of Makey-Makey or Raspberry Pi using the Bridge21 model of 21C learning		(6) Teaching Programming using Bridge21	Yes
TA21-MOD-06			
(1) I feel confident in my ability to plan and implement text based computer programming learning activities according to the Bridge21 model of 21C learning		(1) Planning Bridge21 Activities	Yes
(2) I feel confident in my ability to use text based computer programming languages for teaching computer programming		(2) Content Knowledge	Yes
(3) I have a greater understanding of computer programming syntax and the difference between a computer code and a computer program		(3) Programming Concepts	Yes
(4) I have a greater understanding of computer programming operations such as how to run a program or how to debug a piece of code		(4) Programming Ability	Yes
(5) I feel confident in my ability to write a computer program which can perform arithmetic functions such as multiplication and division and print the results		(5) Programming Processes	Yes
(6) I feel confident in my ability to teach programming through use of the Python programming language using the Bridge21 model of 21st C learning		(6) Teaching Programming using Bridge21	Yes
All Computing Workshop Modules – Comments			
• Having completed the Bridge21 computing workshop what might you now do differently?		Teaching Practice	Yes

9.7 CS CPD Learning Pre Outcomes Questionnaire Example 1

Name*:________________________________	DOB*:___________________________________
PROBLEM SOLVING – TIME 1 - Learning Outcomes	
PROMPT: On a scale of 1 (Strongly agree) to 7 (Strongly disagree) please rate the learning outcomes	
	Strongly Agree Strongly Disagree
(1) I feel confident in my ability to solve logical problems using a variety of problem solving strategies.	1 2 3 4 5 6 7
(2) I have a greater understanding of 'algorithms' and the process of 'algorithmic thinking'.	1 2 3 4 5 6 7
(3) I feel confident in my ability to identify a number of outcomes to problems, which have more than one possible solution.	1 2 3 4 5 6 7
(4) I feel confident in my ability to plan a 21C learning experience, which incorporates logical or algorithmic forms of problem solving.	1 2 3 4 5 6 7
(5) I feel confident in my ability to incorporate algorithmic thinking and problem solving into subject teaching activities.	1 2 3 4 5 6 7
(6) I feel confident in my ability to teach logical problem solving through algorithmic thinking using the Bridge21 model of 21C learning.	1 2 3 4 5 6 7
Computing Knowledge and Skills	
PROMPT: Please list 3 new skills you would like to learn today.	**PROMPT:** Please list 3 existing skills you would like to improve.
(1)	(1)
(2)	(2)
(3)	(3)
*Your feedback will help us to ensure that we continue to meet your training needs. Please also note that the above details will be anonymised, processed, and then stored in accordance with TCD data protection guidelines.	

9.8 CS CPD Learning Post Outcomes Questionnaire Example 2

Name*:________________________________	DOB*:________________________________
PROBLEM SOLVING – TIME 2 - Learning Outcomes	
PROMPT: On a scale of 1 (Strongly agree) to 7 (Strongly disagree) please rate the learning outcomes	
	Strongly Agree Strongly Disagree
(1) I feel confident in my ability to solve logical problems using a variety of problem solving strategies.	1 2 3 4 5 6 7
(2) I have a greater understanding of 'algorithms' and the process of 'algorithmic thinking'.	1 2 3 4 5 6 7
(3) I feel confident in my ability to identify a number of outcomes to problems, which have more than one possible solution.	1 2 3 4 5 6 7
(4) I feel confident in my ability to plan a 21C learning experience, which incorporates logical or algorithmic forms of problem solving.	1 2 3 4 5 6 7
(5) I feel confident in my ability to incorporate algorithmic thinking and problem solving into subject teaching activities.	1 2 3 4 5 6 7
(6) I feel confident in my ability to teach logical problem solving through algorithmic thinking using the Bridge21 model of 21C learning.	1 2 3 4 5 6 7
Computing Knowledge and Skills	
PROMPT: Please list 3 skills you have learned today.	**PROMPT:** Please list 3 additional skills, which require further improvement.
(1)	(1)
(2)	(2)
(3)	(3)

9.9 CS CPD Reactions/Intentions Post Questionnaire Example 3

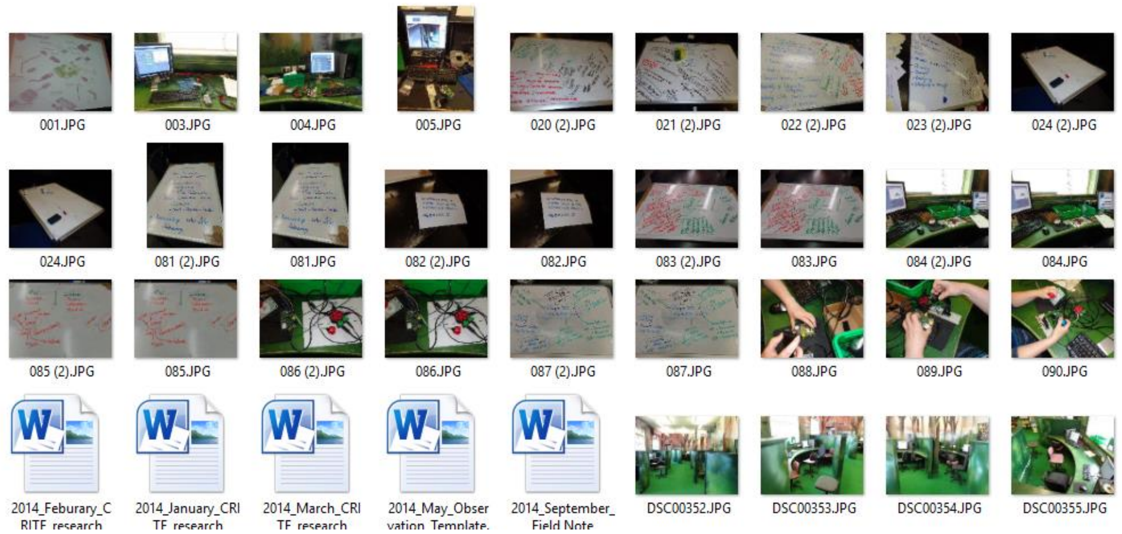
Name*:________________________________	DOB*:_____________________________						
Please use your experience of this workshop to answer the following questions. Your feedback will help us to address your training needs. Please note that your details will be anonymised, processed then stored in accordance with TCD data protection guidelines.							
REACTION QUESTIONS	Strongly Agree						Strongly Disagree
Workshop Design	*PROMPT: On a scale of 1 (Strongly agree) to 7 (Strongly disagree) please rate following*						
(1) The overall process was clearly communicated and met to my satisfaction.	1	2	3	4	5	6	7
(2) The programming activities were well organised and easy to understand.	1	2	3	4	5	6	7
(3) The pace of the training was appropriate for the topics covered.	1	2	3	4	5	6	7
(4) The level of difficulty of the content was appropriate for me.	1	2	3	4	5	6	7
Facilitator							
(5) The facilitator performed well overall.	1	2	3	4	5	6	7
(6) The facilitator is knowledgeable about the subject matter.	1	2	3	4	5	6	7
(7) The facilitator practiced effective time management.	1	2	3	4	5	6	7
(8) The facilitator answered all my questions to my satisfaction.	1	2	3	4	5	6	7
Other Aspects of the Workshop (if applicable)							
(9) This workshop met my professional development expectations.	1	2	3	4	5	6	7
(10) The examples / case studies were helpful.	1	2	3	4	5	6	7
Overall Rating							
(11) The Bridge21 computing workshop was worth attending.	1	2	3	4	5	6	7
(12) I intend to use the Bridge21 model in my teaching on return to the classroom	1	2	3	4	5	6	7
Improvements							
(13) What topics would you have liked to have spent more or less time on?							
More Time:______________________________	Less Time:___________________________						
(14) What key learning did you take away from today's Bridge21 computing workshop?							
(15) Having completed the Bridge21 computing workshop what might you now do differently?							
(16) Finally, how might this Bridge21 computing workshop help you in your classroom teaching?							

9.10 CS CPD Field Note Template adapted from Emerson (1995)

CS CPD Field Note Template Example (Fisher, Byrne, & Tangney, 2015b).

No.	Time Period	Workshop Activity	Description	Kirkpatrick Level
1	Morning	Ice Breaker (10:30am)	Group activity to facilitate introductions.	Learning – Performance of Knowledge and Skills
2		Brainstorming Activity (10:30 to 11:30 am)	Group activity to generate ideas.	
3	Morning / Afternoon	Main Activity (12:00 pm to 2.30 pm)	Group work on assigned project deliverable.	
4	Afternoon	Presentation (2.30pm to 3.30pm)	Group presentation of project.	
5		Reflection (3.30 pm)	Cohort reflection on day's learning experience	

Examples of media which were collected and documented in field notes.



9.11 Teaching Computing in Schools Variables and Scales

9.11.1 Scale 1 Bridge21 Elements adapted from Lawlor et al. (2013)

Bridge21 Element Scale		
Question	Variable	Reported
(1) Teamwork – the teacher organises students into teams, using sorting criteria	(1) Collaborative working	Yes
(2) Technology-mediated – teachers include tasks which involve students using technology	(2) Technology-Mediated Learning	Yes
(3) Learning space – teachers organise desks into groups, where possible, to facilitate teamwork	(3) Consideration of the Learning Space Design	Yes
(4) Facilitator and/or Mentor(s) – teachers change role to guide, facilitator or mentor	(4) Teacher as Facilitator / Mentor	Yes
(5) Reflection – teachers encourage student reflection at the end of learning experiences	(5) Reflection	Yes
(6) Skills development orientation – teachers design tasks which are skills centric.	(6) Skills focused tasks	Yes
(7) Social learning protocols – techniques to encourage confidence and autonomy	(7) Social learning protocols	Yes
(8) Project-based – teachers use projects, with students managing schedules and assigning roles, setting goals.	(8) Goal setting	Yes
Other	Other	Yes

9.11.2 Scale 2 Barriers adapted from Hogarty et al. (2003); Kirkpatrick (2007)

Bridge21 Barrier Scale		
Question	Variable	Reported
(1) It wasn't practical to my situation	(1) It wasn't practical to my situation	Yes
(2) My boss discouraged me from changing.	(2) Lack of support from senior management	Yes
(3) I haven't found the time	(3) I haven't found the time to implement Bridge21	Yes

Bridge21 Barrier Scale continued		
Question	Variable	Reported
(4) I tried it, and it didn't work	(4) I tried it, and it didn't work	Yes
	(5) Lack of confidence	Yes
(5) I have sufficient access to computers at my school	(6) Lack of technical infrastructure	Yes
(6) I have adequate time to learn computer skills	(7) Lack of time to practice programming	Yes
(7) I have had adequate training in using computers	(8) Lack of formal training	Yes
	(9) Lack of accreditation	Yes
(8) I receive a sufficient level of computer-related support at my school	(10) Lack of peer support	Yes
	(11) Lack of demand from students	Yes
	(12) Timetable or scheduling issues	Yes
(9) Other reasons	Other	Yes

9.11.3 Scale 3 Methods for CS adapted from Sentance and Csizmadia (2017a)

Methods used for Teaching CS		
Question	Variable	Reported
(1) Collaboration	(1) Teamwork	Yes
(2) Games	(2) Games	Yes
(3) Online learning	(3) Online Tutorials	Yes
(4) Peer mentoring	(4) Peer Teaching	Yes
	(5) Presentations	Yes
	(6) Projects	Yes
	(7) Printed Worksheets	Yes
(5) Team coding/pair programming	(8) Pair Programming	Yes

9.11.4 Scale 4 Further Bridge21 CPD Scale adapted from Conneely et al. (2012)

Further Bridge21 CPD Scale	
Variable	Reported
(1) Assigning my students roles and organising them into teams for group work	Yes
(2) Integrating group work into computing activities	Yes
(3) Planning computing projects	Yes
(4) Designing learning challenges that promote student autonomy	Yes
(5) Preparing student assessments	Yes
(6) Learning programming	Yes
(7) Using facilitation	Yes
(8) Assigning my students roles and organising them into teams for group work	Yes
(9) Other	Yes

9.11.5 Computing in the Curriculum Scale adapted from Israel et al. (2015)

CS and Subject Teaching in the Curricula		
Question	New Variable	Reported
	(1) Teaching NCCA Short Course	Yes
(1) Teaching computing as part of the curricula	(2) Teaching computing (main stream)	Yes
(2) Teaching computing in a computing club	(3) Computing club (lunch time)	Yes
(3) Teaching computing after school	(4) Computing club (after school)	Yes
(4) Running tutorials for other teachers	(5) Run computing Tutorials (Peers)	Yes
(5) Teaching computing out of school	(6) Mentoring Computing Club (off site)	Yes
	(7) Teaching Leaving Certificate (LC) Computer Science	Yes

9.11.6 Perceived Change in Teaching adapted from Kirkpatrick (2007)

Perceived Change in Teaching		
Question	Variable	Reported
• To what extent have you applied the things you learned?	• Describe any new things (if any) you are doing in your teaching of computing?	Yes
• Please describe any behaviour you have changed?	• What changes in classroom behaviour have you observed through implementing the Bridge21 model?	Yes

9.12 Teaching Computing in Schools Questionnaire

Teaching Computing in Schools
Information
You are invited to participate in Bridge 21 follow-up research exploring teaching computing in schools. The project is based in Trinity College Dublin. The principal investigator is Brendan Tangney. The research coordinator is Dr Katriona O'Sullivan. Lorraine Fisher is a Ph.D. student working under the supervision of Dr Jake Byrne to explore the use of the Bridge21 model as a method of Computer Science Continuing Professional Development. Questionnaire Participation You have the right to remain anonymous and to choose where your information may be used. Should you wish your data to be omitted from published research, you can still complete the questionnaire but your responses will not be processed. **Your Information** All information that is collected by the researchers will be anonymised and stored in accordance with the Data Protection Act at Trinity College, Dublin. In the unlikely event that information about illegal activities should emerge during the study, the researchers will follow the school's Child Protection policy and inform the relevant authorities. There may be lectures, Ph.D. theses, conference presentations and peer-reviewed journal articles written as a result of this project, however the students and school will not be identified. All TCD staff involved in the research undergoes Garda Vetting procedures to receive clearance to work with minors. **Voluntary nature** Participating in this project is voluntary. You may change your mind and stop at any time. You may also choose to not answer a question for any reason. **Benefits**

We hope that this project will result in the improvement of schools preparing students for college and the workforce and aid students in developing 21st century learning skills. **Risks and discomforts** Answering questions about one's experiences may be uncomfortable. You can choose not to answer a question at any time. You may withdraw from the study at any time without penalty. **Confidentiality** We plan to publish the results of this questionnaire. Our reporting will not include any information that would identify you, or your school. To keep your information safe, completed questionnaire responses will be stored on a password protected server. We really hope this work will develop a strong evidence base over the coming years about the kinds of supports at school level that would best enable the leadership, teachers and students to develop and sustain a strong educational culture within schools. We think our research will allow an opportunity for you to articulate what would help to support your school in its future development. We will not identify any individual or school in the process and all data analysis will be available to you to discuss with us and to use for your own purposes.	
1. Do you consent to participation?* • Yes • No	Mandatory Multiple Choice
Demographics	
2. Demographic Information • Name • Email address	Mandatory Comment Box
3. What subjects do you currently teach?	Optional Comment Box
Computing Workshops	
4. Please tick which Bridge21/TA21 computing workshops you completed according to each academic year. Select each option that applies to you. • Digital Media and the Bridge21 Model • Contextual Mathematics (Project Maths and the Bridge21 model) • Problem Solving in the 21st Century (Computational Thinking) • Introduction to programming (Scratch 1: Introduction & Animation)	Optional Matrix / Rating Scale Comment Box

• Intermediate programming (Scratch 2: Game design) • Text-based Programming (Python 1: Introduction) • Exploring Computer Systems (Raspberry Pi 1: Introduction) • Other (please specify):________ Select each option that applies to you. • 2013-2014 • 2014-2015 • 2015-2016 • 2016-2017 • 2017-2018	
5. Please estimate how many students have benefited from you participating in these workshops?	Optional Comment Box
Professional Development	
6. Describe any new things (if any) you are doing in your teaching of computing?	Optional Comment Box
7. Have you completed any other computer science CPD in the last 4 years? • Yes o If yes, what did you attend and why? • No	Optional Multiple Choice If Yes, Comment Box
Post-Primary Subject Teaching	
Computing Languages	
8. What computing languages do you currently teach and to which groups? (Select each option that applies to you.) Languages • Alice • App Inventor • Blockly • C • C++ • C# • HTML	Optional Matrix / Rating Scale Comment Box

• Hopscotch • Java • JavaScript • nodejs • PHP • Python • Processing • Scratch • Snap! • Swift • Other:________________ Year • Year 1 • Year 2 • Year 3 • Year 4 • Year 5 • Year 6	
Computing Systems	
9. What computing systems do you currently teach and to which year groups? (Select each option that applies to you.) System • Arduino • Makey-Makey • micro: bit • Raspberry Pi • Other (please specify):__________ Year • Year 1 • Year 2 • Year 3 • Year 4	Optional Multiple Choice Comment Box

• Year 5 • Year 6	
Computing Resources	
10. What computing resources do you currently use and with which year groups? (Select each option that applies to you.) Resource • CS First (google) • CS Unplugged • repl.it • www.Code.org • www.Codeacademy.com • W3schools • Other (please specify):________ Year • Year 1 • Year 2 • Year 3 • Year 4 • Year 5 • Year 6	Optional Multiple Choice Comment Box
Strategies for Teaching Computing	
11. Here is a list of strategies used for teaching computing. For each one, please identify the extent to which you use it for teaching computing? (Select one option per row.) • Teacher Presentation • Student Teamwork • Peer teaching • Pair programming • Projects • Online tutorials • Printed Worksheets	Optional Select one option Likert Scale: 1. Very effective 2. Effective 3. Neither effective or ineffective 4. Ineffective 5. Very ineffective

• Games • Other (please specify)	
Using the Bridge21 Model	
12. Here is a list of the elements of the Bridge21 model. For each one, please identify the extent to which you use it for teaching computing? (Select one option per row.) • Teamwork • Technology-mediated learning • Consideration of the learning space design • Teacher as facilitator or mentor • Reflection • Skills focused tasks • Social learning protocols e.g. open ended questioning • Goal setting • Other (please specify)	Optional Select one option Likert Scale: 1. Almost never 2. Rarely 3. Sometimes 4. Often 5. Almost always
Classroom Changes	
13. What changes in classroom behaviour have you observed through implementing the Bridge21 model?	Comment Box
Teaching Schedule	
14. When do you teach computing? (Select each option that applies to you.) • I teach computing as part of the mainstream curricula • I run a lunch time Computing Club for my students • I run an after school Computing Club for my students • I run in-school computing tutorials for my colleagues • I mentor in a Computing Club out of school hours e.g. CoderDojo • Other (please specify)	Optional Checkboxes Rank 1 2 3 4 5
NCCA Short Course in Coding	
15. Do you currently teach the NCCA short course in Coding? • Yes o If yes, what has been the result so far? If No why not?	Optional Multiple Choice If Yes,

• No	Comment Box
NCCA Leaving Certificate Computer Science	
16. Are you planning to teach the new Leaving Certificate in Computer Science? • Yes ○ If Yes, what stage are you at with your planning? If No, why not? • No	Optional Multiple Choice If Yes, Comment Box
Barriers to Teaching Computing in a School Context	
17. What barriers do you face when considering teaching computing in your school? (Select one option per row.) • Lack of formal training • Lack of technical infrastructure • Lack of demand from students • Timetable or scheduling issues • Lack of support from senior management • Lack of peer support • Lack of accreditation • Lack of confidence • Lack of time to practice programming • Other (please specify):__________	Likert 1. Strongly Disagree 2. Disagree 3. Neither Disagree nor Agree 4. Agree 5. Strongly Agree
Barriers to Using the Bridge21 Model in a Classroom Context	
18. What barriers do you face when considering using the Bridge21 model to teach computing? (Select one option per row.) • It isn't practical for my situation • My school principal discouraged me from changing • I haven't found the time to implement Bridge21 • I tried it and it didn't work • Timetable restrictions • Lack of confidence • Other (please specify):___________	Likert 1. Strongly Disagree 2. Disagree 3. Neither Disagree nor Agree 4. Agree 5. Strongly Agree

Further Professional Development Requirements	
19. If you are interested in using the Bridge21 model which areas do you feel you require further CPD? (Select each option that applies to you.) • Assigning my students roles and organising them into teams for group work • Integrating group work into computing activities • Planning computing projects • Designing learning challenges that promote student autonomy • Preparing student assessments • Learning programming • Using Facilitation • Other (please specify)	Optional CheckBox Rank
Introducing Computing in Schools	
20. Please indicate your level of agreement with the following. (Select one option per row.) • It is essential that computer science is introduced as a Leaving Certificate subject • It is essential that computer science is included as a Junior Certificate subject/short course • It is essential that computer science is included in the Primary school curriculum	Likert 1. Strongly Disagree 2. Disagree 3. Neither Disagree nor Agree 4. Agree 5. Strongly Agree
Delivering Computing Lessons	
21. Please indicate your level of agreement with the following. (Select one option per row.) • I am innovative in my use of technology for teaching • Computer programming is easy • Being able to teach programming is a relevant skill for me as a teacher • I would have difficulty explaining why teaching programming may or may not be beneficial (to children/students) • I would be keen to introduce computer science in my school • If I heard about a new technology, I would look for ways to	Likert 1. Strongly Disagree 2. Disagree 3. Neither Disagree nor Agree 4. Agree 5. Strongly Agree

experiment with it in my teaching • The educational value of teaching programming is apparent to me • I am willing to share with others my experiences of teaching computer programming • Teaching computer programming is easy • I would be willing to introduce computer science as a subject in my school	
Future Bridge21/TA21 computing workshops	
23. Finally, what suggestions do you have for enhancing future Bridge21/TA21 computing workshops?	Comment Box

9.13 Qualitative Data Analysis Coding Examples

9.13.1 Coding Example 1 Key Learning

1	Key Learning Teacher Response	Coding 1	Coding 2	Sub-Themes	Themes
2	Collaboration is coming	Collabouration	Collabouration	Collabouration	Collabouration
3	Communication – within group work	Communication, Group Work	Communication, Group Work	Communication, Group Work	Collabouration
4	Communication; teamwork; broadcasting	Communication, Team Work, Broadcasting	Team Work, Programming	Team Work	Collabouration
5	Comparison of Scratch / Python coding	Comparison, Programming, Coding	Programming, Coding	Programming	Scratch
6	Computational concepts	Computational Concepts,	Computational, Concepts	Computing	Scratch
7	Computational thinking can happen without computers	Computational Thinking, Without Computers	Computational Thinking, Offline	Computation, Offline	Problem Solving
8	computer programming	Computer Programming	Programming	Programming	Coding Basics
9	Computers are frustrating but practice to use scratch to teach a topic etc.	Computers, Frustraiting, Practice, Scratch, Teaching	Computers, Practice, Scratch, Teaching	Computers, Praatice, Teaching	Scratch
10	Computers in real life - inputs and outputs.	Computers, Real Life, Inputs, Outputs,	Computers, Context	Context	Raspbarry Pi
11	Planning is essential to the programmes success	Planning, Essential, Programming, Success	Planning, Programming, Success,	Planning, Success	Problem Solving
12	The importance of planning and initialisation part of writing programme (hide and show).	Planning, Initialisation, Writing	Planning, Programming, Success,	Planning, Success	Scratch
13	To put more planning into a problem solving exercise (not to launch in) i.e. come up with a plan and work as a team.	Planning, Problem Solving, Exercise,	Planning, Problem Solving, Teamwork	Planning, Teamwork	Problem Solving
14	Prob solve / pre planning.	Problem Solve, Pre Planning	Pre Planning,	Planning,	Planning
15	Combination of codes, the impact one code has on another	Programming, Codes, Combination, Impact	Codes, Combination,	Codes, Impact	Coding Basics

9.13.2 Coding Example 2 Field Notes Reaction Analysis

Field Notes		Coding _1	Coding_2	Sub_Themes	Themes
CSCPD_1196	*I liked the freedom to do what you like*	Freedom, Groups, Facilitation	Freedom, Facilitation	Facilitation	Facilitator Approach
CSCPD_1115	*I liked that we worked as a team and that we had the freedom to choose our own topics and do our own research. I felt good about what I was learning and I learned some skills that I can contribute*	Teamwork, Freedom, Choice, Skills, Contributions	Teamwork, Freedom, Choice, Facilitation	Teamwork, Freedom, Facilitation	Facilitator Approach
CSCPD_825	*I felt it was a positive experience and benefitted from the process. Team working was strong and I learned from others*	Positive Experience, Process, Teamwork, Learning, Other	Positive, Experience, Teamwork, Others	Positive, Experience, Satisfaction	Workshop Satisfaction
CSCPD_380	*I formed new friendships and support mechanisms. I felt challenged at times and did not process info correctly. Overall enjoyed the workshop and how it created a safe learning environment*	Friendships, Support, Challenge, Safe, Learning Context	Friendships, Challenge, Safe, Context	Friendships, Safe, Satisfaction	Workshop Satisfaction
CSCPD_987	*I achieved an understanding of how hardware and software comes together*	Understanding, Content Knowledge, Synthesis	Understanding, Examples, Synthesis	Synthesis, Examples	Computing Examples
CSCPD_1033	*I learned good examples of using Bridge21 Philosophy, good insights into programming scratch to python, and interfacing with electronics*	Examples, Bridge21, Philosophy, Computing, Programming	Examples, Philosophy, Computing	Examples, Computing	Computing Examples

9.13.3 Coding Example 3 Do Differently

A	B	C	D	E
Do Differently Teacher Responses	Coding 1	Coding 2	Sub-Theme	Theme
Have different approach at the beginning of teaching scratch (do group work not using computers at first).	Different , Approach, Beginning, Teaching Scratch, Group Work, (then) Computers	Different, Teaching, Group Work, Computers	Teaching, Group Work, Computers	Approach
Got pupils into groups; remove additional laptops	Pupils, Groups, Remove, Extra Laptops	Pupils, Groups, Laptop	Groups, Shared Laptop	Collabouration
Group rather than individual access to devices	Group, Individual Access, Devices	Group, Access, Devices	Groups, Shared, Devices	Collabouration
Group set up.	Group Set Up	Groups	Groups,	Collabouration
Group using windows	Groups, Using (MS) Windows	Groups, MS Windows,	Groups,	Collabouration
Group work - would use in classroom.	Group Work, Use, Classroom	Group Work, Classroom	Groups,	Collabouration
Group work.	Group Work, Use, Classroom	Group Work, Classroom	Groups,	Collabouration
Have an open task for students.	Open Task, Students	Open, Task	Open-ended, Task	Open-ended
Greater time studying programming	Greater Time, Programming	More Programming	Progrmaming	Programming
great with TY students - communication skills.	Transtion Year (TY), Students, Communicatio Skills	TY, Communication, Skills	Comminication, Skills	Skills
A foundational course on the absolute basics	Foundation Course, Basics	Foundation, Basics	Bascis	Content
A little more knowledge of computer programming which is useful for students.	Knowledge, Programming, Useful, Students	Knowledge, Students, Usefule	Knowledge, Students	Content
A little more scaffolding during intro.	Scaffolding, CPD, Introduction	Scaffolding, CPD	Scaffolding	Process

9.13.4 Coding Example 4 Field Notes Learning Outcomes

Field Notes		Coding _1	Coding _2	Sub_Themes	Themes
CSCPD_38	*it's important to build in a sense of achievement into the process*	Achievement, Process, Building	Achievement, Process	Process, Planning	Planning Bridge21 Computing Activities
CSCPD_84	*I'll use the Bridge21 model to help in my approach to problem solving in my present role. It's also given me new methods of working with teams and groups*	Bridge21, Approach, Problem Solving, Methods, Team, Groups	Approach, Methods, Teams, Groups	Approach, Methods, Planning	Planning Bridge21 Computing Activities
CSCPD_232	*my computational thinking abilities have grown considerably. Revisiting Scratch and the Raspberry Pi have given me confidence and the introduction to Python has encouraged me to learn*	Computational Thinking, Ability, Confidence, Encouragement	Ability, Confidence, Encouragement	Ability, Confidence	Programming Ability
CSCPD_152	*I worked well with partner to complete tasks. Felt more confident and able to do tasks*	Partner, Complete Tasks, Confident,	Partner, Tasks, Confident	Ability, Confidence	Programming Ability
CSCPD_151	*I will be able to give my opinion on python uses to colleagues. I will also investigate Python myself*	Opinion, Python, Uses, Colleagues, Myself, Computing	Opinion, Uses, Colleagues, Computing	Ability, Colleagues	Programming Ability
CSCPD_239	*writing code needs high concentration levels*	Writing, Code, Concentration,	Code, Concentration	Code, Concepts	Programming Concepts
CSCPD_252	*I enjoyed the discussion about computing concepts such as data collection and abstraction and the introduction to the Python programming language through Scratch*	Discussion, Computing, Programming Languages,	Discussion, Computing, Programming	Discussion, Concepts	Programming Concepts

9.13.5 Coding Example 5 Help in Teaching

A	B	C	D	E
Help in Teaching Responses	Coding 1	Coding 2	Sub-Theme	Theme
Address problem solving instil step by step method.	Problem Solving, Step-by-Step, Method	Problem Solving, Step-by-Step	Problem Solving	Problem Solving
allow more problem solving	Allow, More, Problem Solving	Allow, More	More	Problem Solving
Analogical reasoning. 2. Visual coding 3. Reinforce problem solving	Reasoning, Visual Coding, Problem Solving	Coding, Problem Solving	Problem Solving	Problem Solving
Because I have been forced to work in groups and problem solve myself I have enjoyed it. I will feel more comfortable about using this model in the classroom.	Group Work, Problem Solving, Enjoyment, Confidence, Classroom	Group Work, Problem Solving, Confidence, Classroom	Group Work, Problem Solving, Confidence	Problem Solving
Creating scenes / chapters in books. Problem solving. Working collaboratively	Creating Content, Problem Solving, Collaborative Working	Creating, Problem Solving, Collabouration	Collabouration, Problem Solving	Problem Solving
Definitely will help when planning maths lessons and problem solving.	Planning, Maths. Problem Solving	Planning, Problem Soving	Curricila Planning	Problem Solving
Develop logically thinking, problem solving and patience!	Logcal Thinking, Problem Solving, Patience	Logic, Problem Solving, Patience	Problem Solving, Patiance	Problem Solving
develop problem solving skills with learners	Developing, Problem Solving , Skills, Learners	Problem Solving, Skills, Learners	Problem Solving, Skills	Problem Solving
Different perspective. More likely to look at skills abstractly and from problem solving point of view.	Perspective, Abstract Skills, Problem Solving	Abstraction, Skills Problem Solving	Abstraction, Problem Solving	Problem Solving
Do more problem solving skills in class.	Problem Solving, Skills, Classroom	Problem Solving, Classroom	Problem Solving	Problem Solving
	Encouragement,	-		

9.13.6 Coding Example 6 Field Notes Student Engagement

Field Notes		Coding _1	Coding _2	Sub_Themes	Themes
CSCPD_294	*Will help me make classes I feel more interesting for students. I can also show colleagues how the Bridge21 model works*	Classes, Interesting, Demonstrate, Colleagues, Bridge21	Classes, Interesting, Demonstrate Bridge21	Classes, Bridge21	Strategies for Increasing Student Engagement
CSCPD_323	*Introduce more problem solving to get students to think outside the box and more creativity*	Introduce, Problem Solving, Students, Thinking, Creativity	Problem Solving, Students, Thinking, Creativity	Problem Solving, Students, Creativity	Strategies for Increasing Student Engagement
CSCPD_333	*Help students to devise their own problem solving skills*	Students, Construction, Problem Solving Skills	Students, Problem Solving, Skills	Students, Problem Solving	Strategies for Increasing Student Engagement
CSCPD_136	*I will allow my students more time to problem solve and have some independence. I can also introduce my colleagues to these topics and hope they will look into the area*	Students, Problem Solving, Independence, Colleagues, Topics,	Students, Problem Solving, Skills, Independence,	Students, Problem Solving	Strategies for Increasing Student Engagement
CSCPD_387	*Ice-breakers for motivation. I will group students to work together to solve problem*	Ice-breakers, Motivation, Students, Collaboration	Motivation, Students, Collaboration	Motivation, Collaboration	Strategies for Increasing Student Engagement
CSCPD_3	*I could present this workshop to colleagues with less or limited technical ability*	Present, Colleagues, Technical Ability.	Present, Colleagues, Ability	Share, Colleagues	Strategies for Increasing Student Engagement

9.13.7 Coding Example 7 Limitations and Negative Responses

B	C	D	E	F	G
Workshop Question	Teacher Response	Coding 1	Coding 2	Sub-Theme	Theme
Help in Teaching	A beginners raspberry pi course would be good. There is no way I can do an assignment to the level of the technology teachers. Teacher was good. Content too difficult. I choose wrong assignment.	Beginners, Raspberry Pi, Assignment, Level, Difficult	Beginners, Content, Assignment, Difficult	Beginners, Content, Difficult	Basics
Do Differently	A foundational course on the absolute basics	Foundational Course, Computing,	Beginners, Conent, Basics	Beginners, Basics,	Basics
Help in Teaching	Basic worksheet exercises are good class 'fillers' but not confident enough to teach Raspberry Pi to this level yet.	Basics, Excersises, Confidence, Raspberry Pi, Level	Basics, Excercises, Computing, Confidence	Basics, Compuing, Confidence	Basics
Help in Teaching	difficult to keep momentum of a Bridge21 session going over 5 class periods	Difficult, Momentum, Bridge21, Session, 5 Class Periods	Momentum, Bridge21, 5 Sessions,	Bridge21, Momentum	Momentum
Do Differently	Course is so highly structured - hard to be anyway different. No time to day to day practice of software - as task is set for the	Course, Structure, Time, Practice, Software, Project	Stucture, Time, Practice, Software	Time, Practice, Software	Practice
Help in Teaching	Could use maybe for resource teaching/ not my mainstream class.	Reource, Teaching, Mainstream, Class	Teaching, Peripheral Class,	Teaching	Teaching
Do Differently	Feel sorry for the weak student - plan groups carefully	Feelings, Weak Student, Planning, Groupwork	Feelings, Students, Planning, Groupwork	Planning, Groupwork	Planning
Do Differently	Felt more involved today, discussion at start - good for	Feelings, Involement,	Feelings, Involement,	Discussion, Awareness	Disucssion

9.13.8 Coding Example 8 Behavioural Change

A	B	C	D	E
New Practices Teacher Responses	Coding 1	Coding 2	Sub-Theme	Theme
- Computational thinking - Physical computing - Visual programming - Syntactical programming	Computational Thinking , Physical Computing, Visual Programming	Concept, Hardware, Visual, Text Programming	CPD Computing Content	CPD Conent
Allowing more opportunities for group work and discussion of	Groupwork, Discussion, Ideas,	Groupwork, Sharing	Groupwork	Collabouratior
Allowing more pupil freedom to design, problem solve and debug themselves. Encouraging more collaboration. More presentation of work	Students, Freedom, Design, Problem Solve, Collabouration, Presentation	Students, Freedom, Collabouration,	Students, Collabouration	Collabouratior
Collaborative work online, encouraging students to present projects using ICT, more computer based projects, building up my own resource repertoire.	Collabouration, Students, Presentation, Computing Projects, Teaching Resources, Repertoire	Collabouration, Students, Projects, Repertoire	Collabouration, Projects, Reperoire	Collabouratior
Computational thinking - Bebras puzzles Python & Raspberry Pi - working with 3rd Year Computer Science students from DIT Kevin Street Scratch - with MakeyMakey Starting to use App Inventor	Computational Thinking, Physical Computing, 3rd Level Mentoring,	Concept, Hardware, Visual, Text Programming,	CPD Computing Content, Mentors	Mentors
currently exploring Pi	Exploring Raspberry Pi	Exploring	Exploring	CPD Conent
Emphasis on Information Literacy.	Information Literacy	Information Literacy	Information Literacy	CPD Conent
Gradual release of responsibility	Gradual Release, Responsibility	Release, Freedom, Responsibility,	Students, Collabouration	Collabouratior
Group work, iMovie use, collaboration, critical thinking activities, brainstorming prior to learning activity.	Groupwork, Collabouration, Critical Thinking, Activities, Brainstorming, Learning	Groupwork, Collabouration, Brainstorming, Learning	Groupwork, Collabouration	Collabouratior

9.13.9 Coding Example 9 CPD Enhancements

Future CPD Enhancements	Coding 1	Coding 2	Sub-Themes	Themes
I BELIEVE THEY RUN WELL BUT MAYBE SOME SCREENCAST OF THE LECTURERS PROGRAMMING WOULD BE A GOOD REVISION TOOL TO HAVE AFTER YOU HAVE ATTENDED THE COURSE. tEACHERS NEED CPD IF THEY ARE TO IMPLEMENT THESE SHORT AND LC COURSE	ScreenCast, Lectures, Programming, Revision, Course Completion, CPD Access, Short Courses	ScreenCast, Lectures, Conent, CPD, Access	Content, Access, CPD	Access
Being in a team doesn't guarantee equal preparation for class planning, only the person sitting in front of the keyboard. Teachers in workshops do not simulate the problems of teaching teamwork/collaboration - they're too polite, so the reality is different and less	Teamwork, Facilitation, Equal Participation, Dynamics,	Teamwork, Facilitation, Participation	Teamwork, Facilitation	Collabouration
Create links between the schools (teachers taking the workshops are working in) and Trinity Bridge21/TA21 to help encourage student	Links, Schools, Teachers, Bridge21, Awareness, Networks,	Links, Schools, Teachers, Bridge21	Links, Schools, Bridge21	Community
Establishing a community of practice among interested teachers	Community, Practice, Teachers	Community, Teachers	Community	Community
Continued training and support. Access to different resources.	CPD Access, Resources	CPD Access	CPD Access	CPD Access
- Make school timetabling more flexible to accommodate learning by discovery. - Educate more teachers and school mangers to understand the benefits of design thinking and fabrication with technology. - Roll out school based hackathons and clubs - Resource IT labs in schools with suitable storage and equipment for physical computing, i.e., secure storage for micro:bits, arduinos and the like	Timetable, Accommodate Discovery, Management, Benefits, Clubs, IT Infrastructure, Storage	Timetable, Discovery, Management, IT Infrastruture	Timetable, Management, Infrastructure	Infrastructure

9.13.10 Coding Example 10 Limitations with Bridge21

Limitations	Coding _1	Coding_2	Sub_Themes	Themes
There is no way I can do an assignment to the level of the technology teachers. (The workshop) teacher was good. The content was too difficult	Assignments, Technology, Teachers, Workshop, Content, Difficult	Assignments, Technology, Content, Difficult	Assignments, Content, Difficult	Limitations
I have the basic computing ability for this assignment	Basic, Computing, Ability, Assignment	Computing, Ability, Assignment	Ability, Assignment	Limitations
A foundational course on the absolute basics	Foundational Course, Basics	Foundation, Basics	Basics	Limitations
More hands on training to become competent before starting to engage my students	Hands on Training, Competence, Engagement, Students	Practical, Competence, Engagement, Students	Practical, Engagement	Limitations
I don't think that it would currently help, as my knowledge is still quite new	New, Knowledge, Teaching, Practice	New, Knowledge, Practice	Knowledge, Practice	Limitations
(I lack the) confidence to implement today's material unless I got more CPD	Confidence, Implementation, More CPD	Confidence, More CPD	Confidence	Limitations
Follow up on this workshop needs considerable investment for schools	Follow up, Workshop, Investment, Schools	Investment, Schools	Investment	Limitations

9.14 Quantitative Data Analysis Statistics Tables

9.14.1 Reaction Tables Sum

(1) Variable - Facilitator Approach		
Question - Facilitation was an appropriate CPD approach.		
7 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly agree	551	67 %
2. Agree	169	22 %
3. Somewhat agree	33	4 %
4. Neither agree nor disagree	25	3 %
5. Somewhat disagree	9	1 %
6. Disagree	11	1 %
7. Strongly disagree	19	2 %
Valid Response	817	100 %

(2) Variable – Workshop Satisfaction		
Question - The Bridge21 computing workshop was worth attending.		
7 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly agree	534	66 %
2. Agree	153	19 %
3. Somewhat agree	55	6 %
4. Neither agree nor disagree	22	3 %
5. Somewhat disagree	16	2 %
6. Disagree	16	2 %
7. Strongly disagree	17	2 %
Valid Response	813	100 %

(3) Variable – Computing Examples		
Question - The examples / case studies were helpful.		
7 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly agree	487	61 %
2. Agree	189	23 %
3. Somewhat agree	60	7 %
4. Neither agree nor disagree	24	3 %
5. Somewhat disagree	16	2 %
6. Disagree	20	2 %
7. Strongly disagree	13	2 %
Valid Response	809	100 %

(4) Variable - Pedagogical Process		
Question - The overall process was clearly communicated and met to my satisfaction.		
7 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly agree	407	50 %
2. Agree	242	30 %
3. Somewhat agree	82	10 %
4. Neither agree nor disagree	29	4 %
5. Somewhat disagree	20	2 %
6. Disagree	23	2 %
7. Strongly disagree	14	2 %
Valid Response	817	100 %

(5) Variable - Programming Activities		
Question - The programming activities were well organised and easy to understand.		
7 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly agree	378	46 %
2. Agree	246	30 %
3. Somewhat agree	99	12 %
4. Neither agree nor disagree	41	5 %
5. Somewhat disagree	18	2 %
6. Disagree	21	4 %
7. Strongly disagree	12	1 %
Valid Response	815	100 %

(6) Variable - Curricula Content		
Question - The level of difficulty of the content was appropriate for me.		
7 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly agree	341	42 %
2. Agree	235	29 %
3. Somewhat agree	109	13 %
4. Neither agree nor disagree	52	6 %
5. Somewhat disagree	34	5 %
6. Disagree	28	3 %
7. Strongly disagree	14	2 %
Valid Response	813	100 %

(7) Variable - CPD Expectations		
Question - This Bridge21 CPD workshop met my professional development expectations.		
7 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly agree	244	57 %
2. Agree	100	23 %
3. Somewhat agree	38	9 %
4. Neither agree nor disagree	15	4 %
5. Somewhat disagree	9	2 %
6. Disagree	8	2 %
7. Strongly disagree	14	3 %
Valid Response	428	100 %

9.14.2 Reaction Variables Year on Year

(1) Variable - Facilitator Approach										
Question - Facilitation was an appropriate CPD approach.										
7 Point Likert Scale Measure of Agreement	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	155	73	132	75	101	65	151	78	60	72
2. Agree	36	17	22	12	24	15	23	11	18	22
3. Somewhat agree	11	6	8	4	7	4	7	4	4	5
4. Neither agree nor disagree	4	2	3	2	2	1	1	1	1	1
5. Somewhat disagree	2	1	3	2	3	2	2	1	-	-
6. Disagree	1	-	5	3	7	4	3	2	-	-
7. Strongly disagree	2	1	3	2	12	9	5	3	-	-
Valid Response	211	100	176	100	156	100	192	100	83	100

(2) Variable – Workshop Satisfaction										
Question - The Bridge21 computing workshop was worth attending.										
7 Point Likert Scale Measure of Agreement	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	150	71	116	67	100	64	120	63	48	58
2. Agree	34	16	26	15	22	14	44	23	27	33
3. Somewhat agree	19	9	13	8	7	4	11	5	5	6
4. Neither agree nor disagree	4	3	5	3	9	6	4	2	-	-
5. Somewhat disagree	1	-	5	3	5	3	3	2	2	2
6. Disagree	2	1	5	3	3	2	5	3	1	1
7. Strongly disagree	1	-	2	1	11	7	3	2	-	-
Valid Response	211	100	172	100	157	100	190	100	83	100

(3) Variable - Computing Examples										
Question - The examples / case studies were helpful.										
7 Point Likert Scale Measure of Agreement	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	139	67	106	61	84	54	111	58	47	57
2. Agree	40	19	35	20	37	23	51	27	26	31
3. Somewhat agree	19	11	12	7	10	6	12	6	7	9
4. Neither agree nor disagree	3	1	8	5	7	5	6	3	-	-
5. Somewhat disagree	4	2	2	1	4	3	5	3	1	1
6. Disagree	1	-	9	5	6	4	2	1	2	2
7. Strongly disagree	1	-	2	1	7	5	3	2	-	-
Valid Response	207	100	174	100	155	100	190	100	83	100

(4) Variable - Pedagogical Process										
Question - The overall process was clearly communicated and met to my satisfaction.										
7 Point Likert Scale Measure of Agreement	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	106	51	98	56	71	45	93	48	39	47
2. Agree	59	28	47	27	42	27	61	32	33	40
3. Somewhat agree	25	12	13	7	16	11	19	10	9	11
4. Neither agree nor disagree	6	3	6	3	10	6	6	3	1	1
5. Somewhat disagree	7	3	5	3	3	2	4	2	1	1
6. Disagree	2	1	6	3	10	6	5	3	-	-
7. Strongly disagree	4	2	1	1	5	3	4	2	-	-
Valid Response	209	100	176	100	157	100	192	100	83	100

(5) Variable - Programming Activities										
Question - The programming activities were well organised and easy to understand.										
7 Point Likert Scale Measure of Agreement	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	96	46	93	53	62	40	88	46	39	47
2. Agree	59	28	47	27	42	27	67	35	31	37
3. Somewhat agree	27	13	17	10	21	13	22	12	12	15
4. Neither agree nor disagree	12	6	11	6	13	8	4	2	1	1
5. Somewhat disagree	7	3	3	1	6	4	2	1	-	-
6. Disagree	4	2	5	3	8	5	4	2	-	-
7. Strongly disagree	4	2	-	-	4	3	4	2	-	-
Valid Response	209	100	176	100	156	100	191	100	83	100

(6) Variable – Curricula Content										
Question - The level of difficulty of the content was appropriate for me.										
7 Point Likert Scale Measure of Agreement	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	88	42	76	43	59	38	78	41	40	48
2. Agree	56	27	49	28	40	26	62	33	28	34
3. Somewhat agree	30	14	26	15	21	14	25	12	7	9
4. Neither agree nor disagree	19	9	9	5	15	10	7	4	2	2
5. Somewhat disagree	7	3	11	6	6	4	7	4	3	4
6. Disagree	6	3	5	3	10	5	6	3	1	1
7. Strongly disagree	3	2	-	-	4	3	5	3	2	2
Valid Response	209	100	176	100	155	100	190	100	83	100

(7) Variable – CPD Expectations										
Question - This Bridge21 CPD workshop met my professional development expectations.										
7 Point Likert Scale Measure of Agreement	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	-	-	-	-	85	55	115	60	44	53
2. Agree	-	-	-	-	29	19	45	24	26	31
3. Somewhat agree	-	-	-	-	16	10	15	7	7	9
4. Neither agree nor disagree	-	-	-	-	7	5	5	3	3	4
5. Somewhat disagree	-	-	-	-	5	3	3	2	1	1
6. Disagree	-	-	-	-	2	1	4	2	2	2
7. Strongly disagree	-	-	-	-	10	7	4	2	-	-
Valid Response	-	-	-	-	154	100	191	100	83	100

9.14.3 Learning Outcome Tables Sum

(1) Variable - Planning Bridge21 Activities				
Question - I feel confident in my ability to plan computing activities according to the Bridge21 model of 21C learning.				
7 Point Likert Scale Measure of Agreement	Pre Response	Valid Percent	Post Response	Valid Percent
1. Strongly agree	9	3 %	80	19 %
2. Agree	29	10 %	135	31 %
3. Somewhat agree	41	14 %	115	26 %
4. Neither agree nor disagree	43	15 %	54	13 %
5. Somewhat disagree	42	14 %	32	7 %
6. Disagree	53	19 %	11	3 %
7. Strongly disagree	74	25 %	4	1 %
Valid Response	291	100 %	431	100 %

(2) Variable – Programming Ability				
Question - I feel confident in my ability to program.				
7 Point Likert Scale Measure of Agreement	Pre Response	Valid Percent	Post Response	Valid Percent
1. Strongly agree	13	4 %	90	21 %
2. Agree	15	5 %	145	33 %
3. Somewhat agree	37	13 %	112	26 %
4. Neither agree nor disagree	39	13 %	45	10 %
5. Somewhat disagree	33	11 %	20	5 %
6. Disagree	62	21 %	11	3 %
7. Strongly disagree	94	33 %	8	2 %
Valid Response	293	100 %	431	100 %

(3) Variable – Programming Concepts				
Question - I have a greater understanding of basic computer programming concepts.				
7 Point Likert Scale Measure of Agreement	Pre Response	Valid Percent	Post Response	Valid Percent
1. Strongly agree	11	4 %	116	27 %
2. Agree	22	8 %	152	35 %
3. Somewhat agree	30	10 %	73	17 %
4. Neither agree nor disagree	38	13 %	34	8 %
5. Somewhat disagree	41	14 %	24	6 %
6. Disagree	57	20 %	13	3 %
7. Strongly disagree	93	31 %	18	4 %
Valid Response	292	100 %	430	100%

(4) Variable – Programming Processes				
Question - I have a greater understanding of basic programming processes				
7 Point Likert Scale Measure of Agreement	Pre Response	Valid Percent	Post Response	Valid Percent
1. Strongly agree	16	7 %	118	28 %
2. Agree	30	10 %	151	35 %
3. Somewhat agree	35	12 %	88	20 %
4. Neither agree nor disagree	36	12 %	40	9 %
5. Somewhat disagree	39	13 %	14	3 %
6. Disagree	51	17 %	11	3 %
7. Strongly disagree	85	29 %	9	2 %
Valid Response	292	100 %	431	100 %

(5) Variable – Content Knowledge				
Question - I feel confident in my ability to teach computer programming content.				
7 Point Likert Scale Measure of Agreement	Pre Response	Valid Percent	Post Response	Valid Percent
1. Strongly agree	11	4 %	62	14 %
2. Agree	32	10%	151	35 %
3. Somewhat agree	39	13 %	112	26 %
4. Neither agree nor disagree	37	13 %	51	12 %
5. Somewhat disagree	49	17 %	33	8 %
6. Disagree	46	16 %	15	3 %
7. Strongly disagree	79	27 %	7	2 %
Valid Response	293	100 %	431	100 %

(6) Variable - Teaching Programming using Bridge21				
Question -I feel confident in my ability to teach computer programming using the Bridge21 model of 21C learning.				
7 Point Likert Scale Measure of Agreement	Pre Response	Valid Percent	Post Response	Valid Percent
1. Strongly agree	7	2 %	73	17 %
2. Agree	16	5 %	133	32 %
3. Somewhat agree	34	12 %	111	26 %
4. Neither agree nor disagree	42	14 %	54	13 %
5. Somewhat disagree	41	14 %	40	9 %
6. Disagree	68	23 %	10	2 %
7. Strongly disagree	84	30 %	6	1 %
Valid Response	292	100 %	427	100 %

9.14.4 Learning Outcome Variables Year on Year

(1) Planning Bridge21 Activities										
Question - I feel confident in my ability to plan computing activities according to the Bridge21 model of 21C learning.										
7 Point Likert Scale Measure of Agreement	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Pre	%	Pre	%	Pre	%	Pre	%	Pre	%
1. Strongly agree	-	-	-	-	-	-	6	2	3	3
2. Agree	-	-	-	-	-	-	14	7	15	19
3. Somewhat agree	-	-	-	-	-	-	25	12	16	20
4. Neither agree nor disagree	-	-	-	-	-	-	36	17	7	9
5. Somewhat disagree	-	-	-	-	-	-	34	16	8	10
6. Disagree	-	-	-	-	-	-	35	17	18	22
7. Strongly disagree	-	-	-	-	-	-	60	29	14	17
Valid Response	-	-	-	-	-	-	210	100	81	100

7 Point Likert Scale Measure of Agreement	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	-	-	14	17	22	22	29	17	15	19
2. Agree	-	-	28	35	27	26	48	28	32	42
3. Somewhat agree	-	-	23	28	20	20	47	27	25	32
4. Neither agree nor disagree	-	-	7	9	16	16	29	17	2	3
5. Somewhat disagree	-	-	5	6	11	10	13	8	3	4
6. Disagree	-	-	4	5	3	3	4	2	-	-
7. Strongly disagree	-	-	-	-	3	3	1	1	-	-
Valid Response	-	-	81	100	102	100	171	100	77	100

(2) Programming Ability										
Question - I feel confident in my ability to program.										
7 Point Likert Scale Measure of Agreement.	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Pre	%	Pre	%	Pre	%	Pre	%	Pre	%
1. Strongly agree	-	-	-	-	-	-	5	2	8	10
2. Agree	-	-	-	-	-	-	9	4	6	7
3. Somewhat agree	-	-	-	-	-	-	19	9	18	22
4. Neither agree nor disagree	-	-	-	-	-	-	28	13	11	14
5. Somewhat disagree	-	-	-	-	-	-	27	13	6	7
6. Disagree	-	-	-	-	-	-	44	21	18	23
7. Strongly disagree	-	-	-	-	-	-	80	38	14	17
Valid Response	-	-	-	-	-	-	212	100	81	100

7 Point Likert Scale Measure of Agreement.	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	-	-	16	20	25	25	29	17	20	26
2. Agree	-	-	28	35	28	27	50	29	39	51
3. Somewhat agree	-	-	23	28	27	26	50	29	12	16
4. Neither agree nor disagree	-	-	6	7	10	10	27	16	2	2
5. Somewhat disagree	-	-	5	6	5	5	8	5	2	3
6. Disagree	-	-	2	3	3	3	5	3	1	1
7. Strongly disagree	-	-	1	1	4	4	2	1	1	1
Valid Response	-	-	81	100	102	100	171	100	77	100

(3) Variable - Programming Concepts										
Question - I have a greater understanding of basic computer programming concepts.										
7 Point Likert Scale Measure of Agreement.	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Pre	%	Pre	%	Pre	%	Pre	%	Pre	%
1. Strongly agree	-	-	-	-	-	-	6	3	5	6
2. Agree	-	-	-	-	-	-	9	4	13	16
3. Somewhat agree	-	-	-	-	-	-	19	9	11	14
4. Neither agree nor disagree	-	-	-	-	-	-	31	15	7	9
5. Somewhat disagree	-	-	-	-	-	-	30	14	11	14
6. Disagree	-	-	-	-	-	-	39	18	18	22
7. Strongly disagree	-	-	-	-	-	-	77	37	16	19
Valid Response	-	-	-	-	-	-	211	100	81	100

7 Point Likert Scale Measure of Agreement.	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	-	-	24	30	33	32	35	21	24	31
2. Agree	-	-	26	32	32	31	59	35	35	45
3. Somewhat agree	-	-	19	23	12	12	32	19	10	13
4. Neither agree nor disagree	-	-	3	4	8	8	21	12	2	3
5. Somewhat disagree	-	-	4	5	4	4	12	7	4	5
6. Disagree	-	-	5	6	4	4	4	2	-	-
7. Strongly disagree	-	-	-	-	9	9	7	4	2	3
Valid Response	-	-	81	100	102	100	170	100	77	100

(4) Programming Processes

Question - I have a greater understanding of basic programming processes.

7 Point Likert Scale Measure of Agreement.	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Pre	%	Pre	%	Pre	%	Pre	%	Pre	%
1. Strongly agree	-	-	-	-	-	-	8	4	8	10
2. Agree	-	-	-	-	-	-	14	7	16	20
3. Somewhat agree	-	-	-	-	-	-	22	10	13	16
4. Neither agree nor disagree	-	-	-	-	-	-	29	14	7	9
5. Somewhat disagree	-	-	-	-	-	-	29	14	10	12
6. Disagree	-	-	-	-	-	-	39	18	12	14
7. Strongly disagree	-	-	-	-	-	-	70	33	15	19
Valid Response	-	-	-	-	-	-	211	100	81	100

7 Point Likert Scale Measure of Agreement.	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	-	-	23	28	32	31	37	22	26	34
2. Agree	-	-	27	33	29	28	59	35	36	47
3. Somewhat agree	-	-	19	23	20	20	39	23	10	13
4. Neither agree nor disagree	-	-	7	9	10	10	22	13	1	1
5. Somewhat disagree	-	-	2	3	3	3	7	3	2	3
6. Disagree	-	-	2	3	5	5	3	2	1	1
7. Strongly disagree	-	-	1	1	3	3	4	2	1	1
Valid Response	-	-	81	100	102	100	171	100	77	100

(5) Variable – Content Knowledge										
Question - I feel confident in my ability use my computing content knowledge for teaching.										
7 Point Likert Scale Measure of Agreement.	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Pre	%	Pre	%	Pre	%	Pre	%	Pre	%
1. Strongly agree	-	-	-	-	-	-	6	3	5	6
2. Agree	-	-	-	-	-	-	15	7	17	21
3. Somewhat agree	-	-	-	-	-	-	27	13	12	15
4. Neither agree nor disagree	-	-	-	-	-	-	28	13	9	11
5. Somewhat disagree	-	-	-	-	-	-	41	19	8	10
6. Disagree	-	-	-	-	-	-	32	15	14	17
7. Strongly disagree	-	-	-	-	-	-	63	30	16	20
Valid Response	-	-	-	-	-	-	212	100	81	100

7 Point Likert Scale Measure of Agreement.	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	-	-	8	10	21	21	21	12	12	16
2. Agree	-	-	28	35	24	24	61	36	38	49
3. Somewhat agree	-	-	26	32	23	23	44	26	19	25
4. Neither agree nor disagree	-	-	9	11	17	17	25	14	-	-
5. Somewhat disagree	-	-	6	7	11	9	11	6	5	6
6. Disagree	-	-	3	4	4	4	6	4	2	3
7. Strongly disagree	-	-	1	1	2	2	3	2	1	1
Valid Response	-	-	81	100	102	100	171	100	77	100

(6) Variable – Teaching Programming Bridge21										
Question - I feel confident in my ability to teach computer programming using the Bridge21 model of 21C learning.										
7 Point Likert Scale Measure of Agreement.	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Pre	%	Pre	%	Pre	%	Pre	%	Pre	%
1. Strongly agree	-	-	-	-	-	-	6	3	1	1
2. Agree	-	-	-	-	-	-	5	2	11	14
3. Somewhat agree	-	-	-	-	-	-	17	8	17	21
4. Neither agree nor disagree	-	-	-	-	-	-	35	17	7	9
5. Somewhat disagree	-	-	-	-	-	-	33	16	8	10
6. Disagree	-	-	-	-	-	-	47	22	21	25
7. Strongly disagree	-	-	-	-	-	-	68	32	16	20
Valid Response	-	-	-	-	-	-	211	100	81	100

7 Point Likert Scale Measure of Agreement.	2013/2014		2014/2015		2015/2016		2016/2017		2017/2018	
	Post	%	Post	%	Post	%	Post	%	Post	%
1. Strongly agree	-	-	10	13	24	24	26	15	13	17
2. Agree	-	-	26	33	23	22	50	29	34	44
3. Somewhat agree	-	-	24	31	25	25	44	26	18	23
4. Neither agree nor disagree	-	-	10	13	12	12	30	18	2	3
5. Somewhat disagree	-	-	6	8	12	12	17	9	5	7
6. Disagree	-	-	1	1	2	2	3	2	4	5
7. Strongly disagree	-	-	1	1	3	3	1	1	1	1
Valid Response	-	-	78	100	101	100	171	100	77	100

9.14.5 Wilcoxon Tables Learning Outcomes All CS CPD Modules

TA21-MOD-2 Learning Outcomes				
Variables	**Responses**	**Pre Median**	**Post Median**	**P Value**
(1) Planning Bridge21 Activities	47	4	2	p< .005
(2) Programming Ability	47	4	2	p< .005
(3) Programming Concepts	47	5	2	p< .005
(4) Programming Processes	47	4	2	p< .005
(5) Content Knowledge	47	4	2	p< .005
(6) Teaching Programming using Bridge21	47	4	2	p< .005
TA21-MOD-3 Learning Outcomes				
Variables	**Responses**	**Pre Median**	**Post Median**	**P Value**
(1) Planning Bridge21 Activities	89	6	3	p< .005
(2) Programming Ability	89	6	2	p< .005
(3) Programming Concepts	89	7	2	p< .005
(4) Programming Processes	89	6	3	p< .005
(5) Content Knowledge	89	6	3	p< .005
(6) Teaching Programming using Bridge21	89	6	3	p< .005

TA21-MOD-4 Learning Outcomes				
Variables	Responses	Pre Median	Post Median	P Value
(1) Planning Bridge21 Activities	35	4	2	$p < .005$
(2) Programming Ability	35	6	2	$p < .005$
(3) Programming Concepts	35	6	2	$p < .005$
(4) Programming Processes	35	6	2	$p < .005$
(5) Content Knowledge	35	6	2	$p < .005$
(6) Teaching Programming using Bridge21	35	6	2	$p < .005$
TA21-MOD-5 Learning Outcomes				
Variables	**Responses**	**Pre Median**	**Post Median**	**P Value**
(1) Planning Bridge21 Activities	16	7	3	$p < .005$
(2) Programming Ability	16	7	2	$p < .005$
(3) Programming Concepts	16	7	1.5	$p < .005$
(4) Programming Processes	16	7	3	$p < .005$
(5) Content Knowledge	16	7	3	$p < .005$
(6) Teaching Programming using Bridge21	16	7	2	$p < .005$

9.14.6 Teaching Computing in Schools Tables

9.14.6.1 Primary /Post-Primary Profile

Primary and Post-Primary Teacher Profile Data		
Number	Subject	Level
1	All subjects at Primary Level	Primary
2	Primary	Primary
3	Primary	Primary
4	Primary School	Primary
5	All – Primary	Primary
6	Primary school	Primary
7	Primary	Primary
8	All Primary School Subjects	Primary
9	Primary school, English, Irish, Maths, History, Geography, Science, Art, Music, Drama, Physical Education, Social, Personal and Health Education, Religion	Primary

10	Primary	Primary
11	Primary School (all areas)	Primary
12	Primary School	Primary
13	Primary School - All subjects	Primary
14	I'm a primary school teacher so I teach a range...English, Irish, Maths, Science, History, Geography, Music, Art, Drama, Physical Education, Religion and Social, Personal and Health Education	Primary
15	Technical Graphics, Materials Technology Wood, Design and Communication Graphics, Construction, Computers	Secondary
16	Computer Science, Geography	Secondary
17	Mathematics, Programming in Coder Dojo	Secondary
18	Information Communication Technologies and Computing	Secondary
19	Art, Coding & Programming	Secondary
20	Computer Studies - 1st to 6th Year	Secondary
21	Coding	Secondary
22	Information Technology and the Irish Language	Secondary
23	Biology, Chemistry, Junior Science, Information Communication Technologies	Secondary
24	Business Studies, Coding	Secondary
25	Science, Chemistry, Programming	Secondary
26	Physics, Applied Maths, Maths, Science, Computer Science	Secondary
27	Maths. I have become a Deputy Principal since finishing the course so I only have two classes.	Secondary
28	Junior Certificate Schools Programme Librarian - Curriculum & Teaching support for students and teachers	Secondary
29	Principal	Secondary
30	None, I am employed full time as a Special Needs Assistant. I support teachers I work with in Information Communication Technology classes	Secondary
31	English, Religion, History, Civic, Social and Political Education	Secondary
32	English, Religious Education, Special Education Needs	Secondary

33	English (on secondment)	Secondary
34	Design Communication Graphics	Secondary
35	English and Religion	Secondary
36	Social, Personal and Health Education, Leaving Certificate Applied Social Education, Irish, History, Career Guidance	Secondary
37	English Leaving Certificate Vocational Programme	Secondary
38	Art, Craft & Design	Secondary
39	English and Geography	Secondary
40	History, English, Digital literacy, Special Educational Needs	Secondary
41	German, Early Childhood Education	Secondary
42	Maths, Science, Information Technology (sometimes Music), Biology	Secondary
43	Science, Leaving Certificate Biology	Secondary
44	Biology, Science	Secondary
45	Technical Graphics, Materials Technology Wood, Design and Communication Graphics, and Information Technology	Secondary
46	Engineering, Design and Communication Graphics	Secondary
47	Maths	Secondary
48	Art, Digital Media	Secondary
49	Biology, Junior Certificate Science, Transition Year, Material Science	Secondary
50	Business, Accounting, Information Communication Technologies	Secondary
51	Physical Education, Social, Personal and Health Education, Computers	Secondary
52	Mathematics, Gaeilge, Information Communication Technologies, Drama	Secondary
53	Science, Biology	Secondary
54	English, Maths	Secondary

9.14.6.2 Teaching Computing Context

Teaching Computing at School Context		
School Context	%	Post
Do you currently teach the NCCA short course in Coding (Y/N)		
1. I teach the NCCA short course in coding	5	3
When do you teach computing? (Select each option that applies to you). Optional Checkbox, Rank.		
2. I teach computing as part of the mainstream curricula	33	19
3. I run a lunch time Computing Club for my students	21	12
4. I run an after school Computing Club for my students	9	5
5. I run in-school computing tutorials for my colleagues	14	8
6. I mentor in a Computing Club out of school hours e.g. CoderDojo	9	5
Are you planning to teach the new Leaving Certificate in Computer Science? (Y/N)		
7. I plan to teach the Leaving Certificate in Computer Science	9	5
Total	100 %	57

9.14.6.3 Using Bridge21 Element Elements

(1) Variable - Social Learning Protocols		
5 Point Likert Scale Measure of Frequency	Post Response	Valid Percent
1. Almost never	-	-
2. Rarely	1	3 %
3. Sometimes	4	11 %
4. Often	17	47 %
5. Almost always	14	39%
Valid Response	36	100 %

(2) Variable -Technology-mediated learning		
5 Point Likert Scale Measure of Frequency	Post Response	Valid Percent
1. Almost never	1	3 %
2. Rarely	3	8 %
3. Sometimes	7	20 %
4. Often	12	33 %
5. Almost always	13	36 %
Valid Response	36	100 %

(3) Variable -Consideration of the learning space design		
5 Point Likert Scale Measure of Frequency	Post Response	Valid Percent
1. Almost never	2	6 %
2. Rarely	4	10%
3. Sometimes	8	22 %
4. Often	11	31 %
5. Almost always	11	31 %
Valid Response	36	100 %

(4) Variable -Teacher as facilitator / mentor		
5 Point Likert Scale Measure of Frequency	Post Response	Valid Percent
1. Almost never	-	-
2. Rarely	-	-
3. Sometimes	1	3 %
4. Often	17	47 %
5. Almost always	18	50 %
Valid Response	36	100 %

(5) Variable - Reflection		
5 Point Likert Scale Measure of Frequency	Post Response	Valid Percent
1. Almost never	-	-
2. Rarely	-	-
3. Sometimes	8	23 %
4. Often	13	37 %
5. Almost always	14	40 %
Valid Response	35	100 %

(6) Variable -Skills focused tasks		
5 Point Likert Scale Measure of Frequency	Post Response	Valid Percent
1. Almost never	1	3 %
2. Rarely	-	-
3. Sometimes	6	16 %
4. Often	15	42 %
5. Almost always	14	39 %
Valid Response	36	100 %

(7) Variable - Collaborative learning through teamwork		
5 Point Likert Scale Measure of Frequency	Post Response	Valid Percent
1. Almost never	-	-
2. Rarely	1	3 %
3. Sometimes	6	16%
4. Often	10	28 %
5. Almost always	19	53 %
Valid Response	36	100 %

(8) Variable - Goal setting		
5 Point Likert Scale Measure of Frequency	Post Response	Valid Percent
1. Almost never	1	3 %
2. Rarely	1	3 %
3. Sometimes	6	16 %
4. Often	11	31 %
5. Almost always	17	47 %
Valid Response	36	100 %

9.14.6.4 Barriers to Using the Bridge21 Model

(1) Variable - Lack of technical infrastructure		
5 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly disagree	3	8 %
2. Disagree	3	8 %
3. Neither disagree nor agree	4	11 %
4. Agree	12	32 %
5. Strongly agree	15	41 %
Valid Response	37	100 %

(2) Variable - Time to practice programming		
5 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly disagree	2	6 %
2. Disagree	2	6 %
3. Neither disagree nor agree	4	10 %
4. Agree	18	50 %
5. Strongly agree	10	28 %
Valid Response	36	100 %

(3) Variable – Lack of formal training		
5 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly disagree	2	6 %
2. Disagree	4	11 %
3. Neither disagree nor agree	5	14 %
4. Agree	16	44 %
5. Strongly agree	9	25 %
Valid Response	36	100 %

(4) Variable – Lack of Accreditation		
5 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly disagree	2	5 %
2. Disagree	7	19 %
3. Neither disagree nor agree	14	38 %
4. Agree	8	22 %
5. Strongly agree	6	16 %
Valid Response	37	100 %

(5) Variable - Timetable restrictions		
5 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly disagree	6	16 %
2. Disagree	4	11 %
3. Neither disagree nor agree	7	19 %
4. Agree	16	43 %
5. Strongly agree	4	11 %
Valid Response	37	100 %

(6) Variable - Lack of peer support		
5 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly disagree	2	5 %
2. Disagree	8	22 %
3. Neither disagree nor agree	13	35 %
4. Agree	11	30 %
5. Strongly agree	3	8 %
Valid Response	37	100 %

(7) Variable - I haven't found the time to implement Bridge21		
5 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly disagree	9	25 %
2. Disagree	10	28 %
3. Neither disagree nor agree	7	19 %
4. Agree	6	17 %
5. Strongly agree	4	11 %
Valid Response	36	100 %

(8) Variable - It wasn't practical for my situation		
5 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly disagree	7	19 %
2. Disagree	14	38 %
3. Neither disagree nor agree	11	29 %
4. Agree	5	14 %
5. Strongly agree	-	-
Valid Response	37	100 %

(9) Variable - Lack of confidence		
5 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly disagree	12	32 %
2. Disagree	9	24 %
3. Neither disagree nor agree	7	20 %
4. Agree	9	24 %
5. Strongly agree	-	-
Valid Response	37	100 %

(10) Variable - Lack of demand from students		
5 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly disagree	11	30 %
2. Disagree	13	35 %
3. Neither disagree nor agree	5	13 %
4. Agree	8	22 %
5. Strongly agree	-	-
Valid Response	37	100 %

(11) Variable - Lack of support from senior management		
5 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly disagree	16	43 %
2. Disagree	11	30 %
3. Neither disagree nor agree	7	19 %
4. Agree	2	5 %
5. Strongly agree	1	3 %
Valid Response	37	100 %

(12) Variable - I tried it (the Bridge21 model) and it didn't work		
5 Point Likert Scale Measure of Agreement	Post Response	Valid Percent
1. Strongly disagree	17	46 %
2. Disagree	12	32 %
3. Neither disagree nor agree	8	22 %
4. Agree	-	-
5. Strongly agree	-	-
Valid Response	37	100 %

9.14.6.5 Other Methods for Teaching Computing

(1) Variable – Peer Teaching		
5 Point Likert Scale Measure of Effectiveness	Post Response	Valid Percent
1. Very effective	21	75 %
2. Effective	6	21 %
3. Neither ineffective or effective	1	4 %
4. Ineffective	-	-
5. Very ineffective	-	-
Valid Response	28	100 %

(2) Variable – Teacher Presentations		
5 Point Likert Scale Measure of Effectiveness	Post Response	Valid Percent
1. Very effective	4	14 %
2. Effective	19	68 %
3. Neither ineffective or effective	2	7 %
4. Ineffective	3	11 %
5. Very ineffective	-	-
Valid Response	28	100%

(3) Variable – Student Teamwork		
5 Point Likert Scale Measure of Effectiveness	Post Response	Valid Percent
1. Very effective	17	63 %
2. Effective	9	33 %
3. Neither ineffective or effective	1	4 %
4. Ineffective	-	-
5. Very ineffective	-	-
Valid Response	27	100 %

(4) Variable – Pair Programming		
5 Point Likert Scale Measure of Effectiveness	Post Response	Valid Percent
1. Very effective	9	39 %
2. Effective	8	35 %
3. Neither ineffective or effective	5	22 %
4. Ineffective	1	4%
5. Very ineffective	-	-
Valid Response	23	100 %

(5) Variable - Projects		
5 Point Likert Scale Measure of Effectiveness	Post Response	Valid Percent
1. Very effective	13	50 %
2. Effective	9	35 %
3. Neither ineffective or effective	2	7 %
4. Ineffective	1	4 %
5. Very ineffective	1	4 %
Valid Response	26	100 %

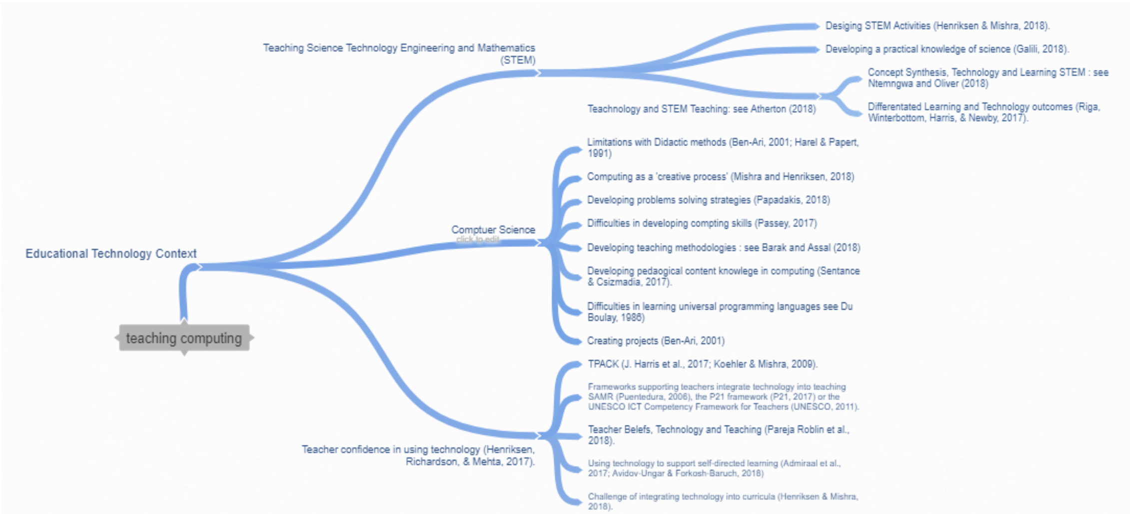
(6) Variable – Printed Worksheets		
5 Point Likert Scale Measure of Effectiveness	Post Response	Valid Percent
1. Very effective	3	12 %
2. Effective	11	42 %
3. Neither ineffective or effective	9	35 %
4. Ineffective	1	4 %
5. Very ineffective	2	7 %
Valid Response	26	100 %

(7) Variable – Games		
5 Point Likert Scale Measure of Effectiveness	Post Response	Valid Percent
1. Very effective	12	48 %
2. Effective	12	48 %
3. Neither ineffective or effective	1	4 %
4. Ineffective	-	-
5. Very ineffective	-	-
Valid Response	25	100 %

(8) Variable – Online Tutorials		
5 Point Likert Scale Measure of Effectiveness	Post Response	Valid Percent
1. Very effective	7	32 %
2. Effective	5	23 %
3. Neither ineffective or effective	10	45 %
4. Ineffective	-	-
5. Very ineffective	-	-
Valid Response	22	100 %

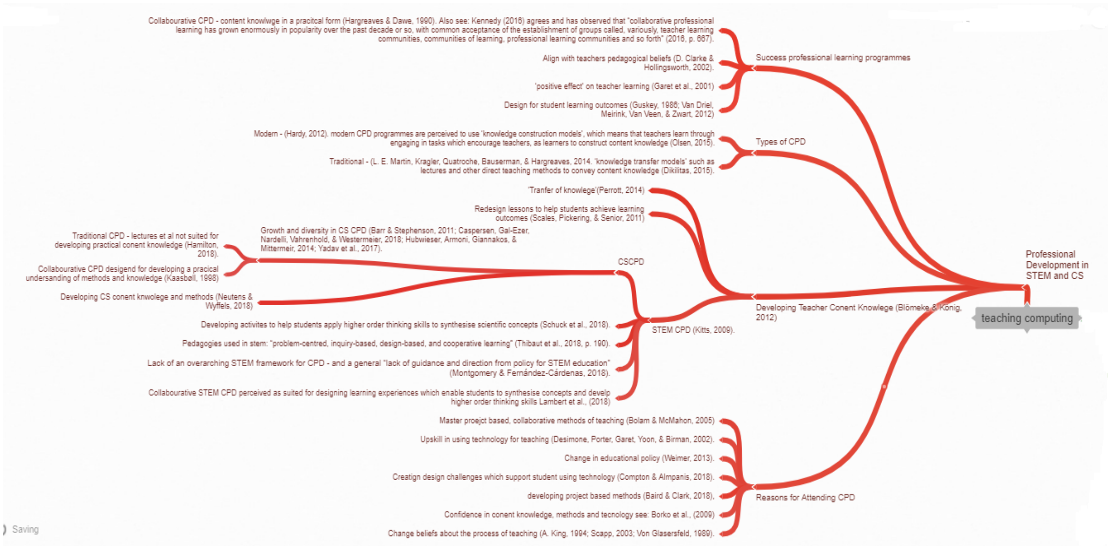
9.15 Visual Literature Maps

9.15.1 Educational Technology Context

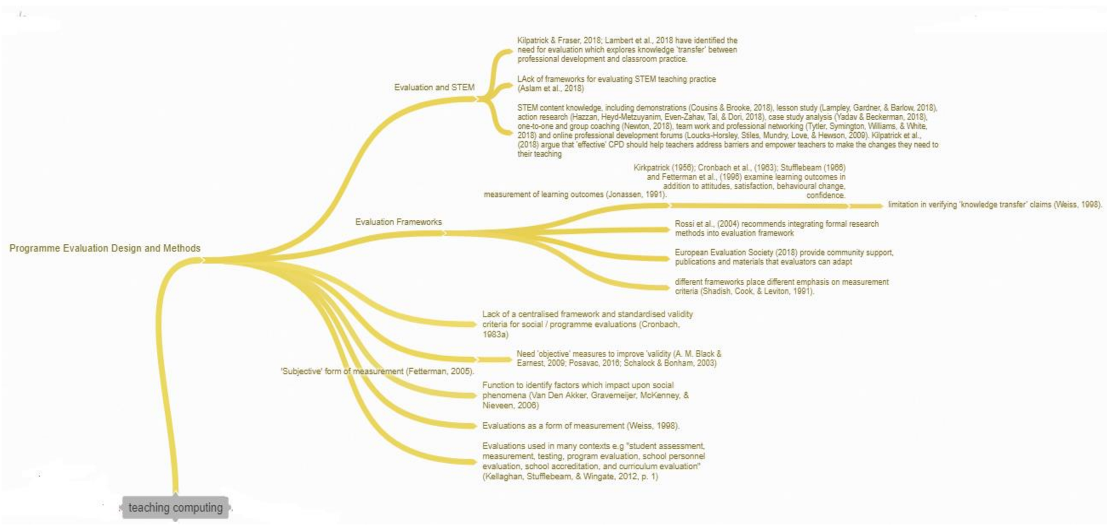


9.15.2 Learning and Teaching Programming

9.15.3 Professional Development in STEM and CS



9.15.4 Programme Evaluation Design and Methods



9.16 Publications

Fisher, Byrne, and Tangney (2015a). Exploring Teacher Reactions Towards a 21st Century Teaching and Learning Approach to Continuing Professional Development Programme in Computer Science. *7th International Conference on Computer Supported Education; Lisbon, Portugal, CSEDU; pp.353-362 doi:10.5220/0005432603530362*

Byrne, Fisher, and Tangney (2015a). Computer science teacher reactions towards raspberry Pi Continuing Professional Development (CPD) workshops using the Bridge21 model. *10th International Conference on Computer Science & Education; Cambridge, UK; IEEE; pp.267-272. doi:10.1109/ICCSE.2015.7250254*

Byrne, Fisher, and Tangney (2015b). Empowering teachers to teach CS — exploring a social constructivist approach for CS CPD, using the Bridge21 model. *2015 Frontiers in Education Conference; El Paso, Texas USA; IEEE; pp.1-9. doi:10.1109/FIE.2015.7344030*

Fisher et al. (2016). Teacher Experiences of Learning Computing using a 21st Century Model of Computer Science Continuing Professional Development. *8th International Conference on Computer Supported Education; Rome, Italy*; CSEDU; pp.*273-280. doi:10.5220/0005906702730280*

Byrne et al. (2016). A 21st Century Teaching and Learning Approach to Computer Science Education: Teacher Reactions. In S. Zvacek, M. T. Restivo, J. Uhomoibhi, & M. Helfert (Eds.), *Computer Supported Education* (pp. 523-540). Cham, CH: Springer International Publishing Switzerland. doi:10.1007/978-3-319-29585-5_30