CS in Schools Evaluation:

An industry-school partnership supporting secondary teachers to teach computer programming

Dr Nicola Carr
Dr Grant Cooper

RMIT School of Education
Shaping Education, Shaping the Future

Acknowledgments

The authors would like to acknowledge the funding provided by *CS in Schools* and the staff for their comments and assistance in the production of this publication. The authors would also like to thank the students, school teachers and industry partners that have participated in the *CS in Schools* initiative.

This report may be cited as:

Carr, N. & Cooper, G. (2019). *CS in Schools Evaluation: An industry-school partnership supporting secondary teachers to teach computer programming.* Melbourne: RMIT University.

About the authors

Dr Nicola Carr *is a Senior Lecturer and Program Manager of the Bachelor of Education programs at the School of Education, RMIT University, Australia. Her background, and interests in the role of technology in school and teacher education have recently coalesced to focus on pre-service teachers' conceptualisation and actualisation of STEM Education.*

Dr Grant Cooper *is a Lecturer in Science and STEM education at RMIT University, Australia. He is an educator, researcher, learner and maker. His research interests include the examination of emerging STEM education discourses, pre-service teacher preparation of STEM-related literacies/perceptions and how digital technologies such as VR have the potential to transform teaching and learning spaces. Grant is also interested in the use of mixed methods research, especially the application of advanced statistical methods.*

Table of Contents

Executive Summary

The aim of this document is to evaluate the pilot of the *CS in Schools* initiative. This evaluation provides information about the delivery and implementation of the *CS in Schools* pilot, considering the perspectives and values of different stakeholders, including teachers and industry volunteers. The document also examines the aims of the *CS in Schools* program, including factors that act as barriers or facilitators of the program and identifies ways to potentially improve the efficacy of the program.

A key aim of the *CS in Schools* program is to help high school teachers develop their confidence and competence in teaching computer science. In our evaluation, there was evidence to indicate that teachers in the study typically increased their self-efficacy to teach computer programming, with the support offered in the program commonly acting as a kind scaffold for in-service teachers develop their skills and knowledge of coding language and programming. Teachers generally held positive views of the pre-designed resources-inclusive of its scope, clarity and alignment with the curriculum. Moreover, they also frequently liked the in-classroom immediate access to expertise from industry volunteers. This element of *CS in Schools* speaks to the untapped value of industry-school partnerships in an effective, contemporary STEM education school syllabus. Conversely, some teachers in the study viewed the explicit pedagogy, which mostly underpins the design of the *CS in Schools* teaching resources, did not align with the pedagogical philosophies they espoused or wanted to facilitate in their learning environment. Other teachers commented that particularly for more advanced students, that the pacing constrained some students in the pilot. Teachers' lack of familiarly with the content was another concern raised by participates.

In relation to the industry volunteers, there was often an altruistic element to their underlying motivations to volunteer in *CS in Schools*, together with a perception that was a lag or deficit in the use of digital technology in schools and what industry trends. Other motivations for some to participate in the program included an eagerness for a professional challenge and the potential to network with others.

Addressing several barriers such as network hardware, software configuration and platforms (e.g. firewalls, password access/ management) in addition to adapting the program to align with individual school needs (e.g. timetables, educator expertise) is likely to improve the efficacy of the *CS in Schools* program. As the *CS in Schools* initiative is in its relative infancy, it's expected that this document will be useful for future iterations of the program and may helpful in addressing perceived areas of improvement and informing future directions of the initiative.

Setting the scene

The fourth industrial revolution and the importance of digital literacy

We have entered the early phase of what has been described as the Fourth Industrial Revolution (IR 4.0). IR 4.0 encompasses a range of technologies that will fundamentally alter the way we study, work, and play, representing new ways in which digital technologies are embedded and used within societies (Schwab, 2017). Relevant examples of this technological shift include the ever-increasing use and sophistication of artificial intelligence, robotics, immersive virtual reality, the Internet of Things, 3D printing and quantum computing (Schwab, 2016). Computer programming is the language that powers the hardware individuals use to solve problems and create. Akin to the change in other domains, schools need to adapt as digital technologies impact different facets of our lives. However, The Programme for International Student Assessment (PISA) results indicate that around one in four Australian 15-year-olds (27%) demonstrate low proficiency in digital literacy (Foundation for Young Australians [FYA], 2016). In the same report, the FYA report a 212% increase in the demand of digital literacy in early career jobs between 2013-2016 and considerably higher wages for jobs that specifically request such skills (FYA, 2016). These figures and the increasing digitisation of our communities suggest the need to re-think the emphasis and teaching of digital literacy in schools, including learning models that challenge the status quo.

CS In Schools

1.1 The program

CS in Schools is a philanthropic venture, founded by Professor Hugh Williams and funded by donations from Australian technology leaders, is focused on helping teachers in Australian secondary schools teach computational thinking (inclusive of computer programming and coding) to support the implementation of the Digital Technologies Curriculum, specifically to parts of grades 7 and 8. A core aim of the *CS in Schools* program is to provide an innovative approach to teaching coding and programing that works by leveraging school-industry partnerships whereby teachers and computer professionals collaboratively co-teach.

Similar to other countries around the world, Australia is experiencing a a shortage of teachers qualified to teach IT subjects in secondary schools (Weldon, 2015). Typically male-dominated subjects such as maths, physics and IT are taught by an ageing workforce but with no net growth in supply. There are limited teachers who feel qualified or sufficiently confident to teach much of the relatively new Digital Technologies curriculum introduced in 2016, particularly the computational thinking and coding required in the *Creating Digital Solutions* strand of the Digital Technologies Curriculum. As a consequence, teachers with limited experience and knowledge of computational thinking (inclusive of coding and programming) are attempting to teach this section of the new curriculum. A study into out-of-field teaching, that is where teachers are teaching into disciplines in which they have no formal qualifications[1], showed that more than half of teachers taking computing and IT classes in years 7-10 across Australian secondary schools are not formally qualified in that field. (Weldon, 2016).

In some cases, schools are unable to implement some sections of the Digital Technologies curriculum at all due to lack of teachers with the relevant skills and confidence to teach coding, or outsource delivery to after school coding clubs or commercial providers, including Grok Learning[2] and CodeClub[3].

An important aim of the *CS in Schools* program is to increase teachers' self-efficacy relating to the teaching of coding and the Digital Technologies curriculum. Self-efficacy, or the strength of a person's belief in their ability to master a practice has been shown to play a significant role in teachers' practices and in student learning. Bandura defined self-efficacy as "the conviction that one can successfully execute the behaviour required to produce the outcomes" (Bandura, 1977, p. 193). In other words, self-efficacy is a teacher's belief (or confidence) that he or she possesses the required teaching skills and knowledge (or competence) and that, when they apply these skills and knowledge, they lead to the desired student outcomes (Soodak and Wodell, 1996). Self-efficacy becomes particularly important when or teaching out of field or teaching new curriculum, such as the Australian Digital Technologies curriculum.

Improving teachers' self-efficacy through increasing their knowledge of and personal mastery of coding and programming, as well as their understanding of different pedagogical approaches to teaching coding and computational thinking, should improve the quality of their teaching in this new curriculum. Recent research suggests that self-efficacy is improved when new knowledge is combined with mastery experiences and observing and participating in new pedagogical approaches (Jaipal-

---

[1] Teachers are regarded as formally qualified to teach a subject if they have studied it above first-year university level and have also been trained in teaching methodology for that subject.

[2] Grok Learning is an initiative coming out of the University of Sydney that provides online courses and modules to teach programming, using a subscription service. Grok Learning also implements the NCSS Challenge, a five week programming competition for schools based on Python the includes access to online self-paced modules to teach programming https://groklearning.com/

[3] CodeClub is nationwide network of volunteer-led coding clubs, based in schools, libraries and other community centres, with sponsorship from the Telstra Foundation https://codeclubau.org/

Jamani & Angeli, 2017). Recent research into the characteristics of effective teacher professional learning (Jacques, Behrstock-Sherratt, Parker & Bassett, 2017) have confirmed that the most effective professional learning occurs when it is specifically relevant to the teacher at the time, grounded in day to day teaching practice, sustained over at least six months, onsite, reflective, and has clearly identifiable links to the curriculum.

CS in Schools uses a model of partnering experienced computer industry professionals with high levels of skills in coding with classroom teachers who have the responsibility of implementing the Digital Technologies Curriculum in schools. *CS in Schools* is based on the successful Microsoft TEALs model that operates in the United States, and in British Columbia, Canada. Computing professionals, typically someone currently working in the IT sector and with a background in computer science and programming, work side-by-side with teachers in the classroom over two terms, helping them to become confident and competent at teaching coding and computational thinking. Having volunteers from the computer science profession has been proven a successful strategy to provide in-class professional development for teachers in the US through the Microsoft TEAL project (Granor, DeLyser & Wang, 2016).

CS in Schools provides detailed lesson plans and teaching resources, assignments, software, and hardware where required. *CS in Schools* also provides training for volunteers, and on-call support for schools and teachers.

CS in Schools has designed and developed approximately twenty hours of coding lessons, based on the use of Python programming language, and is aimed at Year 7 students. The curriculum is designed to be delivered in two contact hours per week for approximately 10 weeks. Schools are provided with an industry volunteer, someone from the IT sector with strong computer programming knowledge, who take the lead in the first term teaching the pre-designed lessons, then remains in the classroom in the second term to mentor and support the classroom teacher as they take on the lead teaching role. The materials are provided free of charge, and schools and teachers have ongoing access to the materials under a creative commons license.

The *CS in Schools* 2019 pilot was intended to trial and evaluate the model with a view to extending the program to more schools in more locations in 2020 and beyond.

1.2 The Pilot

The first stage of the *CS in Schools* program was piloted during Terms 1 and 2 of 2019 in eight Victorian secondary schools, across public and private sectors and in metropolitan and regional locations. The schools that participated in the 2019 pilot included Mount Erin College, McClelland College, Toorak College, Gippsland Grammar School, Sale Catholic College, Greensborough College, Eltham High School and Haileybury. Fourteen volunteer computing professionals mentored ten secondary school teachers across the eight schools in their classrooms.

The programming language used in *CS in Schools* is REPL (Read–eval–print loop). REPL reads user's Phython commands (R), evaluatives entered code (E), Prints results to see the response from the computer (P) and loops back to read in the first step (L). The interactive nature of REPL may suit school environments and novice learners especially because there is relatively quick evaluation of code, including declarations, statements and expressions (Poirier, 2015).

The Pilot commenced in January 2019 with a two-day training workshop for the fourteen volunteers designed to orient them to the Victorian secondary school context. Volunteers participated in a range of workshops on topics including:

- what to expect in the classroom
- classroom management
- creating connections and building relationships between students and teachers (Phil)
- how children learn
- presentation skills
- vocal skills
- outline of the *CS in Schools* syllabus

The workshop culminated in each volunteer teaching a mock coding lesson to a small group of Year 7 students. The *CS in Schools* team allocated volunteers to participating schools. In some cases, one volunteer worked with a single teacher, although in some schools two volunteers worked with a single teacher either in different classes or in the same class but on different days.

The intended delivery model designed by *CS in Schools* is that the Volunteer teaches the content of the packaged curriculum to Year 7 students in Term 1 with the classroom teacher providing support on classroom management and organisation. The classroom teacher was meant to work alongside the volunteer and participate in the same activities as the students, building their knowledge and confidence of the Python language as they work through the curriculum and learning activities. In the subsequent term, the intent was that the roles would be reversed with the classroom teacher taking the lead on teaching the *CS in Schools* curriculum, whilst the volunteer would take a more supportive, background role being in the classroom to provide expertise in Python as and if required by the teacher. The volunteer was also intended to provide additional Python expertise to the more advanced students to deepen and extend their learning.

Throughout the pilot, teachers and volunteers were able to communicate in an online Slack[4] community as a way of providing mutual support and information exchange during the implementation of the pilot project.

During the implementation of the Pilot program in Term 1 it became evident to teachers and volunteers that the program was too content-dense and schools were struggling to cover the provided content within the term. Feedback was taken on board and modifications were made to the content to be delivered in Term 2 of the pilot, reducing the level of content to reflect the feedback provided.

[4] Slack is a proprietary collection of searchable online collaboration and communication tools including chat rooms or channels, file sharing and messaging.

2. Program evaluation

This evaluation provides information about delivery and implementation of the *CS in Schools* pilot, taking into account the perspectives and values of different stakeholders, including teachers and industry volunteers. The evaluation examines the aims of the *CS in Schools* program, the delivery of the *CS in Schools* program including factors that act as barriers or facilitators of the program and identifies and documents ways to potentially improve the effectiveness of the program.

The evaluation also reports on the impact of the pilot project, examining intended and unintended outcomes of the program and the extent to which stakeholder needs and expectations have been met. Specifically the evaluation reports on:

- Participating teachers' self-efficacy (confidence and perceived self-competence) in teaching computational thinking and coding to Year 7 students and the extent to which the *CS in Schools* program contributed to their self-efficacy
- The efficacy of the learning resources developed for the pilot
- The quality of the training workshops
- The quality of the supports to participating teachers and industry volunteers.

At this stage the focus of the evaluation is firmly on the experiences and perspectives of the participating teachers and volunteers. Research into the impact of *CS in Schools* on students and their learning is beyond the scope of this study.

2.1 Research design

There are numerous studies and approaches to measuring teacher self-efficacy, most of which use a series of self-efficacy scales desgined for large population samples. There are a number of recent studies that attempt to investigate teacher and pre-service teacher self-efficacy in relation to STEM, robotics, and computational thinking (Cooper & Carr, 2019, Nadelson, Callahan, Pyke, Hay, Dance & Pfiester, 2013; Yadav, Mayfiled, Zhou, Hambrusch and Korb, 2014; Jaipal-Jamani & Angeli, 2017; Rich, Jones, Belikov, Yoshikawa & Perkins, 2017). Similar questions across all of the scales used focus on teachers' confidence and beliefs about their teaching in the particular field under study. Questions are designed to elicit teachers' perceptions about their attitudes towards teaching in the specified field, self-perceptions about their ability to teach in that field, propensity to respond effectively to student questions and to try new pedagogical approaches. However, the *CS in Schools* pilot program involved a small number of participants in a relatively small number of schools. Hence, a quantitative approach to evaluating the program isn't suitable in terms of the aims of this evaluation previously identified at the beginning of this document.

Schools are also not all the same - the ways they are structured and organised and the approaches they take to curriculum implementation vary from one school context to another. For a program such as *CS in Schools* that is intended to grow to a significant number of schools, it is important to understand the context in which the *CS in Schools* program is delivered and to develop a deeper understanding of how effectively the structure of the program might be adapted to these different contexts. Experience with the roll out of initiatives designed to support some aspect of schooling would suggest that a one-size-fits-all approach is rarely successful and that the more successful programs are those that can be adapted to specific school contexts. These factors suggest a more qualitative research design for the evaluation of the Pilot program.

However, a limitation of qualitative research is that the findings are not generalisable. In other words, the findings of this evaluation cannot be assumed to apply to all other schools, teachers and volunteers who may participate in future iterations of the CS in Schools. These findings provide in depth insights into the experiences of the *CS in Schools* pilot programs in the schools that participated in the Pilot and from the perspectives of the teachers and industry volunteers who participated. However, much can be learned from the richness of these experiences and can be used to inform future iterations of the *CS in Schools* program.

2.1.1 Data collection

For this evaluation, it was considered important to understand the experiences of the key stakeholders in the *CS in Schools* Pilot – the teachers and industry volunteers. The aims of *CS in Schools* are really targeted at teachers, so it was important to understand their lived experience of the Pilot program in their specific context. Industry volunteers also play a pivotal role in the delivery of *CS in Schools* so it was vital to understand the program from their perspective as well. All teachers and volunteers who participated in the Pilot were invited to be a part of the evaluation research. Eight of the ten participating teachers and seven of the volunteers agreed to be part of the evaluation project.

All fifteen participants in the evaluation were interviewed at the end of the Pilot Program in the final two weeks of Term 2, 2019. Semi-structured interviews were conducted that lasted between 30-60 minutes. Interviews were held face-to-face or via telephone at the discretion and convenience of the research participants. Interviews were recorded and transcribed.

The semi-structured interviews gathered data about the participants and their backgrounds and context. For industry volunteers this included questions relating to their professional role and experience in the IT sector as well as any relevant educational experience they had. Teachers were asked about their teaching background, their prior experience with coding and their confidence to teach coding before participating in the *CS in Schools* program. A key focus in the semi-structured interviews was on the participants' perceptions of the program design, resources, organisation and support as well as enablers and barriers to program implementation. All participants were asked to discuss their motivations for participating in the program.

Teachers were also asked to outline the school context and how the CS in School program was implemented and how this may have deviated from the intended implementation model. Teachers were asked about their own perceptions of any changes to their efficacy in teaching programming as a result of their participation in the *CS in Schools* program; on the effectiveness of working with an industry volunteer in the classroom and the impact of working alongside volunteers on their confidence and competence to teach coding and computational thinking. The nature of the questions in the self efficacy scales mentioned above have informed the range of questions included in the semi-structured interviews.

Industry volunteers were asked a series of questions relating to the efficacy of the two-day preparation workshop for preparing them for the classroom context, focusing on the quality and relevance of the content of the workshops, the duration and pace of the workshop, the quality of materials and how engaging the workshop leaders were. Industry volunteers were also asked questions designed to elicit their perceptions about the value they added to the work of the teacher and their views about the impact and value of their participation in the program.

2.2.2 Data Analysis

Data was documented in the form of transcripts and notes made by the researchers during and after interviews. Qualitative data analysis is an iterative and reflexive process that begins as data are being collected (Stake, 1995). Notes made by the researchers during and after interviews together with interview transcripts were read and re-read to identify recurring themes and to draw meaning from the data (Schutt, 2012). The research questions and the questions that framed the semi-structured interviews were used as an initial guide to the analysis to identify themes or patterns in the responses from the evaluation participants. Frequencies of occurrence of themes, words and ideas also formed a critical component of the data analysis. Data was organised into concepts and themes in a series of tables and matrices to help clarify emerging themes within and across participants to identify interrelationships and patterns. This approach addressed some of the challenges resulting from the open-ended questioning that is used in qualitative research, where participants may provide lengthy and complex responses, digress, or discuss multiple themes (Belotto, 2018).

3. Teachers' experiences of CS in Schools

3.1 School contexts

The schools participating in the *CS in Schools* pilot included schools from across sectors – government, catholic and independent schools – as well as schools from regional and metropolitan locations. Schools IT infrastructure varied with the independent schools having superior network reliability and access to a wider range of ICT resources within the schools. The IT infrastructure at the participating government schools was at times problematic (see section 5). The schools also served quite different communities which impacted the cultural capital that students brought with them to their participation in the program and the ability of schools to garner support for their IT programs. In some of the government schools, but not all, teachers commented on the lack of awareness of the potential of the IT sector as a career option for students, whereas in other schools, particularly the independent schools and government schools located in more middle-class areas of metropolitan Melbourne, there was a much higher awareness of IT as a career pathway for young people and parental pressure and support relating to initiatives within the school to foster improved IT skills.

Each of the participating schools took slightly different approaches to the implementation of the Digital Technologies curriculum, which has proven challenging for many schools to implement. Two of the teachers from government schools made comments about the challenges of implementing a new curriculum area, particularly a curriculum like Digital Technologies. These schools, like many others, had attempted to embed the Digital Technologies curriculum across other discipline areas, but with limited success. Simon summed up the challenge succinctly:

> *it's difficult to try and get other people on board with something that they view as (1) not their problem, (2) not their domain area's issue and (3) something they can't really understand anyway. [Simon]*

Many computer-based units that had been implemented in the junior classes in the participating schools were focused on learning key programs, cybersafety, web-design and databases, or had relied on block programming languages, such as Scratch, to address programming. Most of the participating schools were beginning to embed the Digital Technologies curriculum in newly developed STEM programs, alongside robotics and in a limited number of schools, 3D printing, game design and makerspaces[5], seeing computer programming as an essential

[5] Makerspaces are creative, self-directed learning spaces where students can create, invent and learn, often using a combination of everyday materials, but also commonly using electronics.

component of a STEM education agenda and one that until recently has not been addressed in junior secondary schools. As one of the teachers, Matt, expressed 'we want to help the school community understand we want to get into more depth around the computer science part of computing and digital technologies.' However, identifying appropriately qualified teaching staff to teach computer programming was also a challenge, particularly in two of the government schools.

The intended approach for implementing the *CS in Schools* program is outlined in section 1.2 above. However, in some cases the implementation of the *CS in Schools* program deviated from the intended model and was adapted to suit the specific context and needs of the school. For example, in some cases the pilot was implemented in Year 8 classes, rather than the intended Year 7 classes, depending upon what the focus of the existing curriculum had been. Some schools had only one volunteer, others had more than one. Some schools implemented the pilot in only one class, whereas other schools had multiple classes where the *CS in Schools* program was implemented. In one school, the *CS in Schools* curriculum was delivered over two terms rather than one, due to the challenging nature of the particular class. In a different school the *CS in Schools* volunteer worked with one teacher in one class in Term 1 and a different teacher in a different class in Term 2. However, in nearly all cases, the volunteers took on the key teaching role in Term 1 and then switched to a supportive role for the classroom teacher in Term 2. In only one case did the Volunteer not act in the supportive role in Term 2. In some cases volunteers were unable to complete the second term, due to changes in timetable and conflicts this generated with their work responsibilities.

The varied ways that schools implemented the *CS in Schools* project showed that there is some versatility and flexibility with the model and that it can be adapted to suit most school's context and still stay relatively true to its main aim.

3.2 Teachers' IT backgrounds

Teachers were asked a series of questions related to their own teaching background, their background with computer science and with computer programming specifically. Table 1 summarises the responses to these initial questions.

The teachers who participated in the *CS in Schools* pilot program have teaching careers that extend from 8 years to 25+ years of teaching. The majority of the teacher participants were not originally trained in computer science – indeed their original backgrounds included languages, music and PE, with some coming from a media and visual communications or design background, rather than a background in computer programming. Only one of the teachers has a formal qualification in computing. Only two of the teachers who participated in the evaluation have undertaken additional formal training in computer programming or in computers in education. For the most part, the teachers in this project who were teaching IT related subjects were mainly self-taught in computer programming. This pattern reflects national data about the prevalence of out-of-field teaching which shows that more than half of teachers taking computing and IT classes in years 7-10, are not formally qualified in that field (Weldon, 2016). One of the teachers interviewed for this evaluation, Simon, commented n the extreme challenge his metropolitan government school faced in trying to employ appropriately qualified high level IT specialist teacher stating that 'the school didn't get a single applicant.'

Table 1: Teachers' backgrounds

Teacher	Years teaching	IT Qualification	Teaching methods	Computer programming background
Peter	8 yrs	Self taught	Vis arts/media	Scratch, HTML Minimal Python
Steve	20 yrs	Self taught	Media	Scratch. HTML Minimal Python
Matt	8 yrs	Self taught	Primary PE	Scratch (block coding) some Minimal Python Robotics.
Nellie	25 yrs	Post grad Computers in Education	Music IT	Logo, Scratch, Visual Basic Minimal Python
Gary	15+ yrs	Engineering (Manufacturing) Self taught	Maths and Science (Physics)	Pascal, C++ No Python
Simon		Self taught	PE Biology	No Python
Isabel	25+ yrs	Grad Dip IT	Languages	No Python
Stella	15 yrs	B Sci – Applied Physics and Computing.	Physics and IT methods	Many Minimal Python

3.3 Teachers' self-efficacy

3.3.1 Pre *CS in Schools* self-efficacy to teach computer programming

Questions were also asked to elicit the teachers' beliefs about their confidence to teach coding to junior secondary school students prior to their participation in the *CS in Schools* pilot program. As discussed earlier, many teachers who are given the responsibility for teaching coding or computational thinking or other sections of the Digital Technologies curriculum are not qualified in IT, as Susan observed in the following comment:

> *But who teaches it? and it's then put on to people who are not necessarily - don't have a computing background, may not be confident. [Susan]*

Whilst only one of the teacher participants in this evaluation was highly confident in teaching Python specifically, all but two of the teachers expressed a reasonably high degree of confidence in teaching the fundamentals of computer programming to junior secondary students.

Six of the teachers who participated in the evaluation also held roles within their schools as ICT Coordinators. For many, their journey to this role grew out of the previous emphasis in schools to embed the use of ICT across the curriculum where ICT was positioned as a learning tool that would enhance engagement and learning in other disciplines of learning. However, since the introduction of the Digital Technologies curriculum in 2017, there has been a significant shift in emphasis to focusing more on IT as a discipline of study in schools, requiring a different knowledge set. All of the teachers in this evaluation recognised the growing requirement to address those areas of the new curriculum that focus on computer programming. For those teachers without formal training in IT this meant a great deal of 'tinkering' with programming, often in their own time. Four of the teachers had relied on the block programming language Scratch for teaching programming in the junior secondary years, but acknowledged that there was a complexity jump between block programming and text only programming languages. Three schools had used existing online coding programs such as Grok Learning as their computer programming curriculum.

Despite the lack of formal training amongst the teachers interviewed for this evaluation, none were complete novices in relation to computer programming although very few of the teachers had more than a passing knowledge of the Python programming language, which is the basis for the *CS in Schools* program. All, except Simon and Isabel, felt they were reasonably confident about their capacity to teach computer programming in junior secondary years. All, except Simon and Isabel, reported having sufficient knowledge and skills to teach coding and computational thinking and felt able to explain why a program a student has written might not work. All, except Simon and Isabel, felt confident that they could respond to students' questions and problems relating to coding and computational thinking, although most, except Susan, felt that in some cases they may need to take such questions on notice and conduct some out of class research to work out the solutions. Nellie and Susan were both highly confident of being able to troubleshoot in the classroom.

When asked specifically about teaching programming using Python, Matt expressed the highest level of confidence in teaching programming using Python, but only because he had been teaching Python at his school for the last two years. Susan was also highly confident in her ability to teach programming, making comments such as 'I feel very confident with coding' [Susan] reflecting her years of experience in teaching IT at senior levels and her higher level of formal IT qualifications.

Nellie, an experienced teacher of the IT discipline at senior levels, admitted her confidence in her ability to program was quite low:

> *My confidence in programming is pretty low. I've taught software development for years but I've never actually programmed because in Year 12 software development [Year 12 studies] you don't need to program. [Nellie]*

Nellie's perspective was that Year 12 was focused on ensuring that students had an environment that enabled them to succeed in their Year 12 studies and get marks. She believed that a more 'drag and drop' programing environment was more suitable since the 'weaker kids' could get marks by dragging and dropping buttons and things into place. She felt that a text based coding program like Python would act against the 'weaker' students in the VCE years. So her experience was limited to Visual Basic, which she believed offered students of all ranges of ability the best opportunity to obtain a higher ATAR score.

Susan and Nellie both reported that their knowledge and understanding of computer programming and related concepts was high but their personal programming skills could still be improved. Susan stated:

> *if you asked me to write a program I would probably still get it all wrong but in terms of all the concepts I would get that and I've seen lots of different programming languages so I understand that you've got to be careful with your syntax and all that sort of stuff. [Susan]*

However, for Susan and Nellie, their lack of familiarity with Python specifically was not considered by them to be a barrier since they understood principles of programming and were confident that they would be able to work it out as Nellie stated, 'I'm pretty good at doing things on the fly so I would have worked it out.'

Even Gary, who had not taught an IT class prior to the *CS in Schools* program, felt 'pretty happy, yeah pretty confident.' Gary's confidence stemmed from his belief that, despite not doing any coding since his university days, ' a lot of the structures [of programming languages] are the same…with each new program what you're noticing is less about the language and more about the structures.' Gary also had a high sense of confidence in his ability to problem solve and teach himself what was necessary - 'I'm quite useful around a computer I suppose. I take an interest in it I think would be the way I'd describe it. So if I've got a problem I try and solve it myself.'

Steven and Peter likewise had a high sense of self-efficacy related to their capacity to teach themselves the content prior to teaching it to their class. Both commented that if they had enough time and access to online or other 'self-help' resources they would 'be able to skill up' but without preparation time their confidence would be quite low, as Peter pointed out in this statement:

> *but if you said I have to teach it tomorrow I probably would have died. But if I had a holiday break to skill up I would be fine. [Peter]*

In contrast, Simon and Isabel reported low levels of confidence in teaching computer programming of any sort.

> *I have low coding skills. In my IT classes I might have a few kids who code as a hobby who help me [Simon]*

Despite her additional studies into IT and her use of online coding programs in the classroom, Isabel's confidence to teach programming was still very low

> *I'm not confident to teach a whole class and I have no platform for the coding [Isabel]*

Simon shared his perspectives on his confidence, arguing that the very nature of the Digital Technologies curriculum is problematic for teachers who don't have an IT or programming background. Just making sense of what the curriculum is asking schools to put in place is challenging:

> *I don't think the curriculum is too friendly for people who don't have a programming background. The elaborations are good, but you've still got to have an understanding of what's going on before you can even understand what they need too. [Simon]*

In summary, the majority of the teachers in the *CS in Schools* pilot had very limited knowledge of teaching coding using Python, but the majority did not see this as a key barrier. Most of the teachers interviewed believed they were sufficiently experienced in teaching programming previously, or sufficiently knowledgeable about the principles of computer programming and confident that, given time and access to supporting resources, they could develop the skills and knowledge required to teach programming to junior secondary students. However, two of the teachers declared very low levels of self-efficacy in relation to teaching computer programming.

3.3.2 Post *CS in Schools* self-efficacy to teach computer programming

Teachers were asked to report on the impact their participation in the *CS in Schools* program had on their competence and knowledge about teaching coding. All teachers reported increases in their confidence and competence to teach coding as a direct result of their participation in the *CS in Schools* pilot. Even those teachers who were experienced and highly confident in coding reported that their knowledge of coding generally and their knowledge of the Python programming language improved as a result of their participation in the *CS in Schools* program.

Mark was one of the more confident teachers in Python and programming going into the *CS in Schools* pilot program having taught a Python based coding unit for the last two years. However, Mark reported that even with a relatively high level of confidence his knowledge of Python and of coding increased to the extent that he saw new possibilities for teaching and allowing him to extend the students' learning about coding:

> *What we did extended me because some of the content that we did in the CS in Schools program I had not done before. I learned new things. I was conscious that I haven't thought to do some of these things with the kids because I didn't have the knowledge myself. I definitely think it's given me a much better understanding and shaping the different possibilities and ways to do things. My confidence is much higher than it was.*

Despite Nellie's relatively high level of confidence with teaching programming prior to her engagement with CS in Schools, Nellie reported that her participation in the *CS in Schools* pilot has contributed a lot to her confidence and personal competence levels, as well as her motivation to extend her skills:

> *I have been meaning to hit Python for the last four years. My skills are really basic, I don't think I'm any good at it. I can do a little bit more than the Year 7s but other than that I'm now really motivated [Nellie]*

Peter and Steven had expressed a strong sense of confidence about being able to teach computer programming and to 'work it out' on their own, given time, but both expressed an appreciation of how the *CS in Schools* program provided a better quality teaching and learning experience, as Peter points out in this comment:

> *I had an understanding of Python but nowhere near as far along as it is now. Without the CS in Schools program I would have been able to teach [Python] but not as well as this. The [CS in Schools] program has helped immensely and now I will be able to teach in a really well structured way [Peter]*

For Simon and Isabel the jump in self-efficacy was substantial:

> *My participation in the CS in Schools program has contributed massively to my confidence. Now I am much more confident, especially using the CS in Schools curriculum program [Simon]*

> *My skills are much better. Much better. [Isabel]*

The program's main aim is to provide in situ professional learning about teaching coding to secondary school teachers. This aim has clearly been met. The teachers who participated in this evaluation all had IT qualifications ranging from formal computer science degrees to informal, self-taught IT skills and knowledge. The teachers had a range of experience in teaching coding and a range of confidence levels from exceptionally high to very low. Yet all reported positive impacts on their self-efficacy to teach coding as a result of participating in the *CS in Schools* pilot program. If the teachers in the pilot are 'typical' of the range of skills and confidence of teachers who are responsible for teaching the digital technologies curriculum, then the *CS in Schools* model does potentially provide much needed professional development for those teachers with responsibility for teaching the Digital Technologies curriculum, the majority of whom are doing so out-of-field.

3.4 Content of *CS in Schools* teaching/learning resources

3.4.1 Comprehensive and clear

The *CS in Schools* curriculum was based on a pedagogy of explicit instruction. The first part of each lesson involved teacher-led exposition of the key concepts to be addressed in that lesson, followed by a hands-on activity undertaken by the students that would be conducted in the REPL environment. A comprehensive set of Google presentation slides was provided to each teacher and industry volunteer for each week of the 10-week curriculum as part of the *CS in Schools* program. Teachers and volunteers were able to access the teaching resources online. For some of the weeks, there were up to 70 slides made available to support the exposition or explanatory component of each lesson.

All of the participating teachers made positive remarks about the comprehensive and clear nature of the teaching and learning content of the *CS in Schools* program and the logical progression of the content from simple to more complex programming operations.

> *It [the resources] was very well balanced. There were points where I would look at the slides and think, everything you are saying and your lessons you've planned has a slide for every single thing that you're actually saying. [Peter]*

> *I didn't realise how in-depth and how good the resources were going to be [Susan]*

For Matt, the impact of the highly detailed and comprehensive lesson outlines was that his students probably had 'a greater understanding of programming and Python as a whole.'

3.4.2 Scope of content

However, in the first iteration of the *CS in Schools* pilot in Term 1 it became obvious that there was potentially too much content. The *CS in Schools* team had designed and developed resources for a 10-week term, since this is generally how long each school term runs in most Australian schools in most years. However, the reality in schools is that a teacher would rarely have ten weeks to teach their classes due to interruptions caused by students missing classes due to illness or extra-curricular activities, school camps, productions and other school priorities. Steven and Peter explained:

> *We have excursions, we've got the sport, we've got the drama, we've got the plays, we've got those constant interruptions that will throw us out for a week or two. There was an interruption with camp. Then [students] come back and you go, now who remembers what we did last lesson? They have no idea, right? [Steven]*

> *You never get 20 weeks in a semester – 16 for that, right? 14 for sure. It was a revelation for the volunteers that students would miss classes [Peter]*

Apart from not having all ten weeks of a term available to them, the teachers reported that different groups of students made different progress through the content. For some classes, the pace to cover all the content in the original *CS in Schools* curriculum was too fast for many of the students in the class. This may have been a reflection of behavioural issues in the class or of the capacity and interest of the students to engage with the content. Most of the teachers worked with their volunteers to modify the presentations and the sequence of lessons to adjust the pace and depth to the learning needs of their students, as Peter explained:

> *There's a lot of content to get through. We had to really amend it for both of our classes. We didn't get through all of it and we took parts out and rejigged it and Nina worked with us to edit those lessons. [Peter]*

For some teachers, the scope of the content was simply too much for their class to cover in a term and some did not manage to cover all the topics provided.

Following feedback to the *CS in Schools* team about the content, the content for Term 2 was pruned significantly whilst retaining the core concepts of the curriculum.
Teachers reported that the second iteration of the *CS in Schools* program was much more realistic and more readily implemented.

3.4.3 Alignment with pedagogy and philosophies about teaching and learning

Explicit teaching, which lies at the core of the *CS in Schools* program, is a very traditional pedagogy and used extensively across Australian secondary schools, however not all teachers and schools will want to adopt this approach. For some teachers in the evaluation, the pedagogy embedded into the *CS in Schools* program did not necessarily align with their pedagogical preferences and contemporary trends emerging in education, particularly in STEM education, where many schools are beginning to situate their coding units. For example, Matt observed that even though the *CS in Schools* content 'was so detailed and well done' the students at times would 'lose interest.' Matt would have preferred a more active learning approach with less exposition and more activities-based hands on learning. One of the schools also reported that they were moving more towards integrated, challenge or problem-based learning where explicit instruction, the pedagogy that *CS in Schools* is based on, is secondary to more student-centred inquiry pedagogies centred on real-world scenarios. It seems like there might be a future opportunity for the *CS in Schools* resources to include a more varied range of teaching and learning strategies that align more with student-centered, social constructivist pedagogies.

3.4.4 Pacing

Some of the teachers, including Susan, commented on their perception that the style of teaching encapsulated in the *CS in Schools* approach was constraining for the more advanced students:

> *I actually feel like there were some kids that were probably held back [Susan]*

Susan offered a suggestion to make the slides available to the students, either as slides or as self-paced learning modules, so that they could work through them more independently. Her role would be to assist those students who needed more explicit instruction or to troubleshoot issues with students' hands on learning.

Isabel also commented that some of her students mastered the concepts more quickly than others and were ready to move on more quickly than she was prepared to. Isabel was not comfortable with the idea of multiple students making progress at different rates using the self-paced approach suggested by Susan. Her experience of more personalised and differentiated teaching in her other teaching discipline suggested to her that such an approach resulted in a great deal of additional workload on teachers. Isabel preferred the more lock-step approach the *CS in Schools* program was based on. The step-by-step approach seemed to suit her lower levels of confidence, allowing Isabel time to stay one step ahead of her students.

3.4.5 Lack of familiarity with content

One minor limitation identified by three of the teachers was the challenge inherent in delivering content that had been developed by someone else. Whilst the teachers were unanimous in their praise of the comprehensive and clear nature of the teaching materials, some commented on the difficulty of delivering the content when you are not deeply familiar with it. Steven's comment was typical:

> *It is difficult using other people's content because you know you've not written the words. You don't necessarily know what's going to come next without you sitting down and looking at the teacher's guide as well as the classroom teacher resources. [Stuart]*

3.5 Other benefits

In addition to the evident positive impact on participating teachers' self-efficacy relating to teaching coding, this evaluation has identified additional benefits to schools offered by the *CS in Schools* program.

3.5.1 The pre-planned teaching resources

A key component of the *CS in Schools* program is that it offers schools a curriculum that is already planned out in detail, with all required teaching and learning resources provided and a platform into which students can code. The program can be immediately implemented in a classroom over a typical term without teachers needing to spend considerable amounts of time developing a scope and sequence, lesson plans, assessment tasks and associated teaching and locating or creating teaching and learning resources. Susan and Simon's comments are indicative of a shared sentiment amongst the teachers:

> *I loved that they just had it all; the presentations, the resources. It was just there [Susan]*

> *So I think one of the main benefits of CS in Schools thing was the activities were there, the curriculum was followed, it was done and all you really had to do was learn it and implement it. [Simon]*

Teachers in Australian schools are time poor and under pressure. Any way to save time is seen as a benefit , as Simon argued:

> *Teachers don't really have endless hours to go funneling through a million attachments on not the best-designed website trying to find stuff that might actually fit into what they're doing. It saved me a lot of hassle from having to actually come up with all the activities as well. [Simon]*

Gary's comment echoes Simon's view of the time-saving aspect of CS in Schools, pointing out:

> *Brendon has probably saved me hours of chasing down the knowledge and the solutions. It's created space for to come in and tinker [Gary]*

Gary's comment suggests that because everything was already prepared, this freed up his time to 'tinker,' to explore the Python program and focus on the development of his own skills, in the belief that 'It's very hard to teach Python if you cant' do it' [Gary]. Gary also commented that

> *What CS in Schools have done well is that they've supplied the resources for teaching. It's the type of activities they've brought and the slides they've done. Because it's structured it makes it really, really easy to teach' [Gary].*

Susan observed that teachers are often forced to teach a class without appropriate time for preparation:

> *Teachers are very busy people and sometimes you don't have time to prep. You just have to walk in and wing it [Susan]*

For teachers with high levels of confidence and subject knowledge, 'winging it' is part of everyday teaching practice. However, for teachers who teach out of field, such as half of those who teach Digital Technologies curriculum, 'winging it' can result in an inadequate or incomplete learning experience for students. The shortage of appropriately qualified and confident IT teachers in secondary schools can lead to ineffective implementation of the Digital Technologies Curriculum. As Simon explained:

> *I think staffing it is a big problem because most people who have a lot of experience with coding and whatnot generally aren't teachers. So I think a lot of schools I suppose I don't want to say are ignoring it, but just don't have the capacity at the moment to actually do it effectively. [Simon]*

Being able to rely on the resources provided as part of the *CS in Schools* program provides teachers with greater confidence in their ability to cover the required curriculum. As Isabel pointed out, 'the slides are very descriptive, with step by step. You can't go wrong.' [Isabel]

3.5.2 Access to skills and knowledge on the spot

The teachers were unanimously positive about the knowledge and skills the industry volunteers brought with them into the classroom. As Nellie pointed out 'the great part is the fact that they know what they're doing' [Nellie], a reference to the fact that even the more experienced programmers amongst the teachers lacked specific knowledge about Python and computer programming.

For teachers such as Susan, someone who was highly confident and experienced in teaching IT in schools at all levels, but not particularly familiar with the Python language, one of the major drawcards of the *CS in Schools* program is having the volunteer in the class. Having that extra expertise on hand meant that errors could be identified and resolved on the spot, as Susan commented:

> *what I liked about the CS in Schools program, was if I got stuck there was somebody else who was more experienced in coding and they might be able to figure out where the error was [Susan].*

Simon reinforced this view with his reflection:

> *it's a lot quicker when you've got someone that can I suppose just help you out in real-time rather than than spending an hour or two trying to work something simple out. [Simon]*

Likewise, Isabel appreciate the 'on-tap' support:

> *It was - it was amazing just to be taught and I could ask things but what about that, and if this does - if this happens or I want this how do I do this? This is the bit that I don't like about [online programming modules] because if you're stuck, you're stuck. If you - especially the beginner and you get stuck, who do you ask if you don't have anybody that can help you [Isabel]*

Again, the *CS in Schools* program saved teachers the time they would otherwise have to spend after school or between classes identifying the students' coding errors and investigation possible solutions. On the spot troubleshooting was seen as a significant benefit the *CS in Schools* program brought.

Another advantage of having the industry volunteers in the classroom was their ability to draw on their experience to provide authentic examples of how coding is applied in the larger world to augment the curriculum. The 'real life examples Nina shared of what coding might look like in the real world – they [students] loved it' was seen by Steven as much more engaging than if he alone had taught the content. The authenticity of knowledge and experience was also valued by Nellie who admitted to 'borrowing' her volunteer to take him into her Year 10 class. The students in this class were 'really excited to have hi there, they could ask him questions about their computer programming and show him their code' [Nellie]. Nellie's students also asked questions about IT careers and opportunities and the interactions with the volunteer generated a high level of motivation amongst the students.

Three of the teachers commented on the benefits of having a second adult in the classroom, believing that the students always learn more when there are two 'teachers' in the room, since they have greater access to someone who can help them. There is also more of a chance of being noticed if they have not been engaged in their learning tasks. Susan's comment was typical:

> *the fact that we had another volunteer in the class, that was the biggest seller because the kids are always, always going to learn more when there's a second person in the room because when there's two people that* can help them - it's brilliant. [Susan]

3.5.3 In situ professional learning

The *CS in Schools* program provides teachers with in situ form professional learning for teachers who have been allocated responsibility to teach coding but who may have limited coding skills and experience. For Susan the biggest advantage of this form of professional learning was its authenticity and the fact that to attend this professional learning didn't involve taking time out of the classroom.

> *The fact that somebody was coming in and they were going to teach us was the selling point. They were going to teach us during class, which meant we didn't need to take time out, we didn't need to go off on a PD and leave work for our classes. They were going to come into our class and teach us during our class time what we need to teach for our next class. Isn't that just - how can you say no to that? [Susan]*

In other words, the industry partner acted as a 'scaffold' for the teacher in terms of their expertise of the subject content.

3.6 Overall impressions

Overwhelmingly the teachers participating in the *CS in Schools* pilot had highly positive experiences and believed in the efficacy of the program. All of the teachers interviewed indicated a desire to continue to implement the program in the future. The following comments were reflective of the overall impressions given by the teachers:

> *The resources are great. Please keep doing it. Please keep sending volunteers. [Susan]*
>
> *Amazing to have volunteers teach and be able to ask them questions to clarify. [Isabel]*
>
> *Grateful our school chose to put up their hand to do the project, I would love to continue with it. [Isabel]*
>
> *I think it was really beneficial for us to be a part of the pilot. I would like to see it happen again maybe next year. [Peter]*
>
> *I'd like to deepen the relationship and the partnership. [Gary]*
>
> *I'd happily continue [Matt]*
>
> *Yes, I'd do it again [Simon]*

Sustainability of the school-industry partnership created by the *CS in Schools* program was a theme amongst half of the teachers. These teachers made references to the time it takes for teachers to feel comfortable with teaching new content confidently and that ongoing relationships between schools and *CS in Schools* was important in building teachers' self-efficacy. For example, Isabel and Susan believed that many teachers need two to three 'passes' at new content to develop the necessary confidence to teach in a new area. In addition, some of the teachers in larger schools wanted to see more teachers included in the *CS in Schools* approach, so that the load for teaching computer programming was spread and more teachers developed the capabilities to teach programming. Matt wanted to see a deeper relationship develop between his school and CS in Schools. Similarly, Gary was focused on a longer term, more sustainability of the program, encapsulated in his comment:

> *I think the idea of bringing in industry has a great potential to actually create real and sustaining partnerships. But it's got to be a 10-15 year commitment. It needs to be that long because it takes five years in some ways to change a culture and then it's about the next five years that you can actually really develop and grow. That's when lots of the interesting stuff is done and becomes part of the DNA, part of the organisational structure, part of the group think of the school. [Gary]*

4. Industry Volunteers' experiences of CS in Schools

4.1 Industry volunteers' backgrounds

The people who acted as volunteers in the *CS in Schools* Pilot program ranged in experience from a recent graduate from a computer science qualification to two volunteers with more than 20 years experience in the IT industry. Table 2 below provides a summary of the volunteers' ages, IT experiences and roles as well as their main reasons for participating in the *CS in Schools* Pilot program. Their ages ranged from early 20s to mid 50s. Only one of the seven industry volunteers who participated in the evaluation research was a female. The majority of the volunteers defined their professional role as software developer or software engineer. Other professional roles included product manager, data warehousing, software and automation testing. All of the volunteers had used or used Python programming languages as part of their professional roles and were highly confident in their understanding and knowledge of the language and of coding more generally. All but one of the volunteers were currently employed in IT divisions of companies or in IT companies (although the distinction between these is blurring!). The exception to this, Josh, had recently left the IT industry after a 25 year career to pursue a teaching career and combined his volunteer work for *CS in Schools* with his teaching studies.

Table 2: *CS in Schools* Pilot 2019 Industry volunteer backgrounds

Volunteer	Age	IT sector experience	Professional Role	Reasons for participating
Brendon	Under 30	2 yrs	Software developer	Pervasiveness of IT sector and career opportunities for young people
Jason	Under 30	5 yrs	Software developer	Helping young people to learn about IT
Darryl	30-35	8 yrs	Software engineer	Helping young people to learn about IT; perceptions of deficits in education in relation to IT
Josh	Over 40	20 yrs	Programmer, data warehousing	perceptions of deficits in education in relation to IT; career change to teaching
Nina	Under 25	Less than 1 year	Software developer	Pervasiveness of IT sector and career opportunities for young people; gender imbalance in IT
Kieran	Over 50	25+ yrs	QA, software and automation testing	perceptions of deficits in education in relation to IT; helping young people to learn about IT; personal challenge
Jeremy	30-35	10 yrs	Product manager	Pervasiveness of IT sector and career opportunities for

				young people; personal challenge

4.1.2 Education experience

None of the volunteers had any first hand experience of teaching in schools and classrooms, with the exception of Josh who was currently studying to become a teacher. Kieran and Josh had limited exposure to the current education system and curriculum through their own children's schooling but had not taught or been particularly involved in schools. Jason had been an instigator of a peer mentoring and tutoring program during his university studies where he developed a passion for sharing his knowledge and 'teaching' but had not had experience of schools beyond his own school education. In fact, the majority of the volunteers admitted that their knowledge of the current education system and schools was limited.

4.1.3 Reasons for participating

The *CS in Schools* program is heavily reliant on volunteers from computer science and the IT sector. It is important therefore to understand what motivated the volunteers to participate in the program in order to inform the ongoing recruitment of volunteers.

A strong sense of altruism permeated conversations in the interviews about why the volunteers chose to participate in the *CS in Schools* Pilot program. All of the volunteers spoke in various ways about genuinely wanting to be part of a program that positioned IT more positively within the education system. Six of the seven volunteers were employed by two companies that actively encouraged their employees to participate in some form of volunteer program and gave employees time away from work to participate in volunteer programs. These two companies had also facilitated volunteer recruitment presentations by *CS in Schools* team members . Volunteering was an integral part of their workplace culture.

The sense of altruism shared by the volunteers appear to coincide with two key drivers that emerged from the interviews – a strong shared belief in the pervasiveness and importance of the IT sector to the current and future economy, as well as perceptions of deficits in how the education system and schools in particular are currently preparing young people to understand and be able to participate in a society and economy where IT underpins so many aspects of day to day life.

Importance of IT sector

Brendon, Nancy and Jeremy expressed strong beliefs about the importance and pervasiveness of IT, and programming specifically to future economic and social prosperity.

> *Where you look at where companies are heading everything is digital. Companies that use software as their primary product grow much faster. I'm sure this trend will continue. If we want the next generation of the workforce to be apt and knowledgeable in those skills they need to learn what coding means. But also help them remove some of the stigma associated with programming and computer science [Jeremy]*

Jeremy identified a 'stigma' associated with computer science which he felt may dissuade young people from considering an IT career. This was a theme repeated by Brendon, who also believed that programming and its pervasive role in many aspects of day to day life are not well understood by young people but represents potential career opportunities that he was happy to help demystify:

> *try and show [young] people that it's achievable, and not magic because it's about to be everywhere. [Brendon]*

The aims of CS in Schools, focused as it is in Years 7 and 8, aligned with Kieran's beliefs about the importance of exposing students to IT and its opportunities early to help them make informed choices about career options.

> *I think if you're not giving the kids the option to have a look at it early on, then they can't now make an informed choice. [Kevin]*

Nina is very passionate about the IT industry and argued that 'everything requires coding'. She identified that there are relatively few females in technology and that IT in particular is perceived of as being male-dominated, logical, even boring and entirely linked to sitting at computers – the stigma perceived by Jeremy and Brendon. In contrast to this mis-conception, Nina sees IT, in particular coding, as providing significant opportunities for expressing creativity and for offering opportunities for more females to be active participants.

Deficit view of IT in education

A key theme that emerged from the interviews with the industry volunteers was a shared perception that the current education system was not adequately representing the role that computing plays in the contemporary world. Some of the volunteers were highly critical of the schooling system for not keeping pace with the rest of the world in relation to computing. For example Darryl felt that:

> *education moves very slowly in comparison to the rest –if you look at how quickly businesses transform their practices and processes to accommodate new technologies and new thinking, and then you compare that to the education system, I think there's a massive lag there [Darryl]*

Further, Darryl also believed that STEM-related subjects, including computer programming, 'don't get enough attention' and were not a priority, although admitted that this was largely based on his own schooling experiences of ten year ago. The two volunteers with secondary school-aged children, Josh and Kieran, also commented on perceived deficits of the current schooling system in relation to the role of IT and coding. Josh, currently studying to become a teacher, observed that the level of confidence in teaching programming/coding amongst current teachers he worked with during his professional experience placements was low and that IT subjects were restricted to spreadsheets and data bases. His perceptions about the lack of attention paid to coding/programming from his professional placements was reinforced by his own children's limited coding/programming experiences in their current secondary school education.

The perceptions held by the volunteers shared here are not grounded in research but reflect personal perceptions of the current state of play in school education with respect to STEM education. They are not therefore necessarily accurate. However, they speak to the drivers that are acting on the volunteers in their decision to participate in programs such as CS in Schools.

Other reasons

Other reasons for participating largely related to personal and professional goals. Volunteers mentioned they were looking for something to 'challenge them' or to provide a different context for developing and applying generic skills that they believed would assist their career. For example, Kieran wanted 'wanted something that would really challenge me personally' whereas Jeremy believed there were strong parallels between teaching and his leadership role, both of which he thought involved taking 'relatively complex concepts and break them down to help people become competent with that knowledge.' For Jeremy the *CS in Schools* project was an opportunity to practice those skills in a different context.

The founder of CS in Schools, Hugh Williams, was also a motivating factor in many of the interviewed volunteers' participation in the pilot program. Hugh is highly respected in the IT industry and gave personal presentations to staff at two large companies he had connections with. The opportunity to make professional connections with Hugh was also a reason cited by four of the volunteers for their participation. However, the potential for connections with Hugh was a secondary motivation in all cases and not sufficient in and of itself for the volunteers to participate.

Recruiting volunteers will be a critical element of the future success of the program. The volunteers in the pilot were motivated by their strong beliefs about the potential for the IT sector to provide interesting and rewarding careers for more young people. Volunteers were therefore keen to assist teachers and schools to position computer science as a positive career option for young people. Volunteers were also eager to demystify computer science and to make a positive contribution to how schools could more authentically reflect the reality of the IT sector. In future iterations of *CS in schools*, it seems important to communicate more effectively about the time commitment required to effectively participate in the program, including the importance of preparation in order to meet the individual needs of students. Not all classes progress at the same pace; industry volunteers reported having to develop examples, mini-activities; rehearse and prepare. Moreover, factors like travel time to and from the school are important matters to discuss. A small number of volunteers believed that there was transferability of skills they developed from their volunteer role to their professional role. Emphasising these aspects of the volunteer role may assist in attracting the number of volunteers required to expand the *CS in Schools* program into the future.

4.2 Volunteer Training Workshop

CS in Schools developed and ran a two-day weekend 'Boot Camp' workshop for all industry volunteers to orient them to the *CS in Schools* program and help prepare them for the realities of the secondary classroom environment. The Boot Camp weekend included presentations from *CS in Schools* staff, including those responsible for the development of the *CS in Schools* curriculum package as well as current teachers from some of the participating schools who presented on a range of relevant topics. The Boot Camp culminated in each Industry Volunteer teaching a small group of secondary school children some Python coding. Each Industry Volunteer was evaluated on their teaching by the presenters (and the children) and given feedback.

Industry volunteers were asked a series of questions about the Boot Camp to gauge its efficacy in preparing them for the school and classroom environment as well as preparing them to become familiar with the content of the *CS in Schools* curriculum package. These questions related to the timing of the workshop on a weekend, the duration of the workshop and the pace of delivery of the workshop. Industry volunteers were also asked about the content of the workshop – its relevance to supporting them to teach in a classroom context, presentation quality, and engagement. Questions about the expertise and engagement of the presenters were also asked.

Overall impressions of the workshop by all of the industry volunteers was that is was highly engaging, well structured and delivered effectively by very experienced people.

Comments made by Josh and Darryl were typical of the positive impressions the volunteers had of the duration, 'Only two days and given the constraints they covered a good amount' [Josh] and the pace of the two-day workshop, 'I think the pacing was pretty good' [Darryl]. Whilst there was a lot of material covered in the two-days, it was 'not overwhelming' [Darryl].

The high level of interactivity that characterised the workshop sessions was seen as a positive, as suggested by Brendon's comment 'It was really interactive which you can't help but find engaging.'

The volunteers were equally enthusiastic about the quality of the presenters during the two-day workshop, finding them knowledgeable, engaging and well prepared, as reflected in these comments:

> *Teachers and speakers were people who had real knowledge of the classroom [Nina]*
>
> *Well prepared and engaging [Jeremy]*
>
> *They knew what they were talking about and did a good job. They were really, really good. [Brendon]*
>
> *The teachers were amazing [Jason]*

The volunteers were unanimously enthusiastic about the relevance and quality of the content covered in the workshop. In particular, the volunteers appreciated the insights into how classrooms are managed, with many of the volunteers commenting about how they could apply the approaches they were exposed to in the workshop to the classroom environment, for example:

> *There's heaps of stuff they mentioned that we tried, like how to engage students when asking them a question, how to talk to kids when they answer something incorrectly. [Brendon]*
>
> *Once I was in the classroom a lot more of the lessons [from Boot Camp] became applicable when in front of the class [Kieran]*

A highlight for the majority of the volunteers was the opportunity to receive feedback on their presentation and speaking skills in two different sessions – a skill they all felt was transferable to their workplace, or as Nina put it, 'helpful beyond the classroom,' as well as assisting them in their classroom role. For Jason, the nature of the feedback on his presentation skills was 'not something you learn anywhere else I don't think. I found it super useful, even for work, like presentations in work, it helped me quite a bit.'

For all but one of the volunteers the most valuable activity in the two-day workshops was the delivery of a mock lesson to some Year 7 aged students. Each volunteer had to teach some coding using the *CS in Schools* materials to a small group of students. This activity in particular, together with the other sessions that gave insights into current secondary school classrooms, gave the volunteers a stronger sense of confidence as typified by Josh's, Jason's and Jeremy's comments:

> *There were practical examples and tips to the extent that people [volunteers] developed a sense of "I can do that" [Josh]*
>
> *Certainly gave me a little boost of confidence going into my first class [Jeremy]*
>
> *Teaching in front of those students was the best for me [Jason]*

Two of the volunteers, Brendon and Darryl, were the only volunteers to make slightly negative comments about the workshop. Both volunteers were placed in classrooms in Term 1 that were characterised by high levels of inappropriate and disruptive student behaviour. In hindsight, they both believed that while the workshop was good, it was 'not really the same as an actual classroom' [Darryl]. AS Brendon pointed out:

> *The difference between that [mock lesson] and the regular classroom is just – it does mislead you a little. The way I taught that lesson is really different to the way I'm about to teach my class, I have a troublesome, difficult class in my school, whereas in the mock lesson I had four well-behaved volunteers [Brendon]*

However, both Brendon and Daryl acknowledged that it would be difficult to re-create a real classroom in a training workshop - 'there was no real substitute to the actual experience' [Darryl]. For Darryl and Brendon Workshop was not as relevant as it did not prepare him for the particular context of his classroom (see also 5.9).

5. Industry-school partnerships as one element of a contemporary STEM education syllabus

5.1 Industry-school partnerships

One of the more interesting and distinct elements of the *CS in Schools* initiative is its scope to act as a 'conduit' between industry and schools. An emerging focus in the STEM (Science, technology, engineering, maths) education literature is the use, efficacy, and challenges associated with industry-school partnerships. As Hobbs (2018) wrote, for schools, the reality of creating and maintaining industry partnerships can be challenging, they need to be meaningful for all stakeholders. From the evidence collected in this evaluation, it seemed that it was meaningful for school educators-certainly in terms of the scope, clarity and alignment of the teaching resources used in the program. Equally, volunteers' altruistic motivations and their commonly held perceptions that schools were not doing enough in this space certainly suggested that they thought it was a worthwhile endeavor. Considering the difficulty that many schools have initiating and maintaining industry links, it certainly appears that this is a significant strength of the *CS in Schools* program. Commonly school teachers and leaders are so time poor and burdened with other responsibilities. The capacity of *CS in Schools* to act as a link between industry and school certainly seems one of its more valuable and defining features, one that potentially adds significant value to a school's STEM syllabus and its links to community.

6. Barriers

Teachers and volunteers identified a number of barriers during the implementation of the *CS in Schools* program. These largely related to the availability of appropriate IT network infrastructure, hardware, software, access issues and student IT literacy. Important but lower impact barriers related to the nature of the rooms in which the *CS in Schools* project was delivered. Many of the barriers related to setting up the volunteers into the school network, setting up students' access to the Python learning environment, so were high impact barriers in the first week or two of the program. However, once these issues had been resolved their impact was low. Table 3 below outlines the relative impact of these barriers on the implementation of the *CS in Schools* program. Issues were classified as high impact if nearly all the participants identified them as problematic and if the issue was a persistent one throughout the implementation of the pilot. Issues were classified as low impact if they were reported by only one or two participants and were one-off or very infrequently occurring issues. Issues were moderate if some of the participants reported them, or if they were reported by many participants but were short-term barriers and once resolved had minimal impact on the implementation of the *CS in Schools* program.

Table 3: Summary of barriers to implementation of CS in Schools

Barrier	Impact		
	High	Moderate	Low
Network infrastructure	x		
Browser interface & software		x	
Password access and management	x		
Firewalls		x	
Access to hardware	x		
Timetable		x	
Facilities			x
Student IT literacy		x	
Response times		x	

6.1 Infrastructure

There was a significant difference between the experiences of volunteers in the independent and Catholic schools compared with those who were allocated to government schools. The independent schools had significantly faster and more reliable networks than the Government schools resulting in fewer interruptions to the program's implementation with fewer dropouts. Teachers and volunteers in government schools developed a series of workarounds to run with in case of network reliability issues, including volunteers resorting to their mobile phone hotspots to connecting to the Internet.

A related issue faced by some of the volunteers in government schools was initial issues in getting access to the school WiFi network. The security systems in place did not allow for instant connection to the school network, with volunteers having to go through school IT technicians to be connected. Delays such as these are not experienced in the corporate sector and were considered inefficient, with time being lost in the first week of the program because volunteers could not access the school's network.

CS in Schools is designed around the use of an online platform, REPL, where students carry out their programming. REPL requires a certain amount of latent bandwidth to operate effectively. At times schools' networks would slow and drop out in some parts of the school, particularly if the students were all working in the online REPL environment at the same time, which is quite normal practice in a computer science class. The consequence of the poor network experience was that students' coding activities couldn't upload to the REPL environment properly. As Jeremy observed:

> *If too many people were accessing REPL everything would slow down. You end up giving a poor experience to the students and that's their first impression.*

Some teachers, such as Gary, would have preferred not to have to rely on the use of REPL as the coding platform, since lack of reliability of the school's network acted as a barrier to learning at times:

> *I'm not a big fan of REPL because it just another technological barrier and set up and processes that teachers need to do.*

Nellie also was initially not in favour of the REPL environment because of its reliance on the network to use. In contrast, Simon, at another government school did not experience the same network access issues or any technological barriers related to REPL as his colleagues.

Most of the teachers and volunteers in other schools were more positive in their comments about REPL, particularly in relation to the capacity of the REPL environment to create and house assignments, monitor students' achievements of milestones and provide feedback to the students. However the alternative of using Python alone was not considered a viable one, since students would potentially be working on different versions of Python whereas within REPL everyone is using the same version of Python. Further, students would need some type of environment to submit their completed coding assignments, either through an existing learning management system or via email, creating a different set of compatibility and access issues.

Teachers in Government schools seemed to be less concerned about network reliability issues since this was their normal experience, although Gary's preference for an offline approach is clearly related to the poor reliability of his school's network. Gary's comments that 'a good teacher will know how to circumnavigate and get around those barriers' suggests an acceptance of network issues, an inevitability that a workaround solution will need to be found at some point. Nellie commented that:

> *The volunteers are beyond frustrated. Because I've been in schools a long time, I'm like, well, okay, this is what it is, so let's just think of ways to work around it. [Nellie]*

However, many of the industry volunteers who were in Government schools were surprised and appalled at the lack of network reliability. Their experience reinforced their perceptions that the education system lagged significantly behind the corporate and industry sectors in relation to IT. As Jeremy observed:

> *There's a really big gap between what facilities and infrastructure industry tends to invest in and take for granted because that's just how we need to have it to do our job, to run our business effectively*

Volunteers were disappointed that network issues resulted in a poor learning experience for students, which may have had adverse effects on students' interest in coding, exactly the opposite effect of why most of the volunteers chose to participate in CS in Schools.

6.2 Browser interface and software platforms

A significant delay to progress at the beginning of the each iteration of the *CS in Schools* program was the need for students to download alternative browsers such as Chrome as their browser. Many of the schools had prescribed bring your own device (BYOD) programs in place where students are required to purchase or lease specified laptop computers. If the devices used a browser that was incompatible with the online REPL platform, such as Internet Explorer that comes with Lenovo computers, or Edge with Surface Pros, students needed to download Chrome as the preferred browser. In some schools this proved to be an initial barrier as time spent installing Chrome meant a delay in teaching the content of the *CS in Schools* curriculum, as exemplified by these typical observations from Josh and Jason:

> *We had to spend more time getting students to download Chrome so they could use REPL [Josh]*

> *Most of the kids were using Internet explorer because that's what the computer comes with so we had to spend the first class just making sure everyone got Chrome installed [Jason]*

Nellie, a teacher in a government school, was surprised that many of the Year 7 students in Term 1 did not know what a browser was, and that there were different browsers, let alone know how to download and install one.

Two schools reports additional issues with the REPL environment where students had to download code from a library to complete their programming activity. Nellie all reported that:

> *We've got problems with our networks and the IT guys are just not [heavy sigh]….we can't view images so when kids have to run a particular program they can't see the code which sort of defeats the purpose. I'm modifying things heavily to accommodate the limitations in our school [Nellie].*

For Peter and Steven, teachers at one of the independent schools, their volunteer quickly found a solution to the errors that students were receiving when using code from a REPL library.

6.3 Password access and management

One of the biggest frustrations for teachers and volunteers alike was the challenges of managing password access for students. In Term 1 particularly a lot of time was lost in helping new Year 7 students establish connections to the school network as well as to the REPL environment. Many Year 7 students were unfamiliar with emailing, with having to remember passwords and with network structures in general. The need to familiarize these students, who are new to secondary school as well, with the school network reduced the amount of time initially spent on implementing the *CS in Schools* curriculum. As Brendon pointed out:

> *Students can't access their student emails or verify their accounts, they can't retrieve their passwords very easily. The schools' system got in the way [Brendon]*

Kieran also reported that the very first lesson was entirely devoted to trying to get students connected to the school network, when they did not know their email address or their student numbers.

Setting up connections to the REPL environment also posed significant challenges and may have contributed to some teacher's initial concerns about using REPL. REPL requires a separate log in and passwords, but teachers (and volunteers) do not have access to the students' REPL passwords. If students forgot their REPL passwords, which they did, then they would need to go through the entire password re-set process, again losing learning time. Issues with passwords continued throughout the term, as Kieran reported:

> *A lot were able to log on but then forgot their passwords, so had to get password re-sets. We'd ask them to write their passwords in their diaries and they didn't write it down or would forget their diaries. Constantly have these issues [Kieran]*

Simple solutions to managing passwords were implemented by some teachers, for example using a written template and requiring students to record their initial passwords at the beginning of the term, then ensuring that template was in each class throughout the term.

Most of the issues relating to password and email management came from teachers and volunteers in the government schools. This was not an issue that was seen as significant in the non-government schools.

6.4 Firewalls

Schools all have firewalls in place for normal network security reasons as well as to prevent students from accessing inappropriate sites. During the delivery of the *CS in Schools* program volunteers would at times need to access external sites to download additional pieces of code to supplement the curriculum, or to demonstrate and provide authentic examples of coding in practice. However, to do so meant having to arrange ahead of time for these sites to be unblocked from the firewall. Again, delays in being able to download a small piece of code that would solve students' problems interfered with the learning experience for the students.

6.5 Hardware

In some schools, particularly government schools that did not have a BYOD program, the computers students had access to were slow and old. Darryl and Kieran, volunteering in a metropolitan government school, observed a clear dichotomy between the access to hardware in the school they volunteered in and the access in one of the private schools:

> *The equipment they had was very old and very slow. A lot of time ended up being wasted in just students turning them on and then logging in and getting all their stuff ready. That just felt like I was basically in a first world versus third world kind of thing [Darryl]*

Even in schools with BYOD programs, access to laptops was varied, with students frequently forgetting their laptops, having flat batteries, or their computers were with the technicians as expressed in these typical comments from teachers and volunteers in government schools

> *I found that the kids constantly will not bring in their laptops. They'd have to grab a school one and there's not always enough [Brendon]*

Most schools had a class set of 'spare' laptops that would be distributed amongst students who did not have access to their own laptop or who had been allocated a desktop computer that wasn't working effectively, however, in many cases there would be students sharing a computer which meant that not all of the students had the opportunity in each lesson to work on their own coding projects. Further, shared laptops also meant more time in logging on and setting up.

6.6 Response times to solving issues

One of the frustrations that volunteers in the government schools experiences was the length of time it took to have set-up issues resolved. For example, early in Term 1 one of the volunteers and their teacher had requested access to some coding resources that required a site to be unblocked from the firewall. At the end of Term 2 the site was still blocked. The lack of availability of IT technicians in the government schools proved an irritation and acted against the effective implementation of the program.

6.7 Facilities

The majority of the participants in the *CS in Schools* pilot program did not make comments about the facilities they were teaching in. However, one volunteer commented in the inappropriateness of the teaching space for a coding class and the projection equipment he had access to. In one term the class he was allocated to was taught in a film room, with only a small TV screen to project from. He felt that students located in the back of the classroom would not have been able to easily see what he was projecting on the screen. In the second term, his teaching space was the woodwork room, again with a relatively small screen TV to project the slides.

6.8 Student (computer) literacy

Surprisingly, some of the teachers and volunteers made comments about the lack of computer literacy among some, but not all, of the Year 7 students. Surprising because assumptions are made about the digital literacies of young people. This was particularly the case with students who had used tablet devices or mobile phones in their primary school years and who were accustomed to touch screens rather than to computer keyboards, as Brendon and Jason pointed out in their comments:

> *The fact that they [students] just hate their laptops but do anything on their phones. The fact that they'd rather do something like coding on a mobile device blows me away. They don't know where everything is on their keyboard, how to navigate the cursor, copy and paste properly, they always want to click everything because they're use to touch screens. [Brendon]*

> *I found that most of the students struggled on a computer really. Like they play games with touch screens and they're pretty fluent with that, but I found their skills with the keyboard were missing. They don't know how to copy, paste, some basics that we assume in the program. So we spent half the class teaching them how to use the browser, save bookmarks, how to copy and paste, or how to use a URL. That was in first term so the kids were new to the school and new to computers as well. In term two the students were more fluent with a computer. [Jason]*

It would appear that computer specific skills are not necessarily being taught in primary schools and that this is resulting in a wide range of pre-existing knowledge amongst the students new to secondary school. As Brendon observed the skills gap posed challenges to him teaching the *CS in Schools* program initially:

> *The skills gap between my students is massive. I have students who can't put words to the screen. Then I have students who get bored no matter what I throw at them, because they should be lessons and lessons ahead. It's hard to deal with that difference [Brendon]*

Brendon was facing the need to be able to differentiate the curriculum to meet the needs of individual students and to reflect their specific learning goals and challenges. Whilst this is what experienced teachers deal with every day, it is a challenge for those with less teaching experience to be able to address.

Two teachers from two government schools observed that the children who struggled with learning coding were those students who also had issues with their general literacy and numeracy. Poor English literacy skills and numeracy skills make coding in any language much more difficult.

6.9 Student behavior

Student's behaviours varied from class to class, term to term and between schools. The overwhelming majority of teachers and volunteers experienced classes where students' behaviours were not perceived as problematic. However, in a small minority of cases, student behaviours were seen a disruptive to the intended learning. In those classes where inappropriate or distracted behaviours were high, the volunteers and teachers were less able to progress through as much of the planned curriculum as in classes where inappropriate behaviours were minimal. In classes where disruptive behaviours dominated, the classroom teachers spent more time managing the behaviours than engaging in the content of the *CS in Schools* program. In one case, the classroom behaviours resulted in a significant deviation from the intended program, with the volunteer working with small groups of more motivated and engaged students, while the classroom teacher gave alternative work to the more disruptive and disengaged students. In Isabel's case, the need to manage student behaviours had a detrimental impact on her capacity to engage in the intended professional learning embedded in the *CS in Schools* program:

> *I had to learn from him but I had to keep these kids in their seats and sit down and listen because you're not - you're now going to do your exercise and that was the thing that I found the hardest. That was really, really hard. [Isabel]*

6.10 Timetable

Each school timetabled their classes differently but the majority of classes participating in the *CS in Schools* Pilot program had at least one double period of coding each week. Teachers and volunteers alike commented on the loss of focus and concentration in the double periods. This seemed counter-intuitive, since double periods would give students more time to work on mastering their coding. However, the issues with diminishing concentration and engagement in double periods potentially reflects the explicit teaching pedagogy embedded in the *CS in Schools* program. If double periods are common, then a potential consideration is to reduce the proportion of exposition in the lessons and increase the proportion of hands-on learning activities.

7. Recommendations

As the *CS in Schools* initiative is in its relative infancy, its expected that the following nine recommendations will be useful for future iterations of the program:

a. **Adaption of the learning activities** to align more with contemporary understandings of effective pedagogical philosophies. For some teachers in the evaluation, the pedagogy embedded into the *CS in Schools* program did not align with their pedagogical preferences and contemporary trends emerging in education, particularly in STEM education, where many schools are beginning to situate their coding subjects.

b. **Authenticity of the program-** it was noted that it would be good to see more authentic examples that connect to the students' lives for them to see how coding relates to their world. Student-centered exercises (e.g. learning journal, metacognitive exercises) might help students see their learning progression more effectively

c. **Build in more flexibility to the model** – a one-size fits all approach does not work. Schools, teachers and industry volunteers subverted the model in ways to make it fit the specific school context – the way they structured and delivered their Technology curriculum, the timing and the approach to students and managing students' engagement and behaviours.

d. **Reduce the scope of the curriculum/learning package**. Needs to take into account that you don't get 10 weeks of learning in a term due to the range of interruptions to a teaching schedule by non-classroom activities (excursions, sports days, report writing and curriculum days etc)

e. **Consider not running *CS in Schools* in Year 7 Term 1** – many issues in many schools related to getting Year 7s computer literate and confident and competent with accessing school networks, emails and passwords. Ate into time for delivery. Instigate a template for recording emails, log in and passwords into School network and REPL environment to minimise disruptions caused by poor student self-regulation.

f. **Consider running the program in the same school more than once.** Different teachers come with different levels of prior knowledge of coding and Python and different capacities to learn – two terms may not be sufficient time to allow them to develop the required confidence and competence. Different teachers may be allocated to teaching in the digital technologies curriculum from year to year, indeed from semester to semester. Teachers not allocated time in the day to day life of a school to run PD for teachers who are new to coding generally or Python specifically.

g. **Be more forthcoming with industry volunteers about the time commitment** – need to understand that despite the detail of the provided curriculum/learning package that some preparation ahead of time is required to adapt to the needs of the students. Not all classes progress at the same pace; industry volunteers reported having to develop examples, mini-activities; rehearse and prepare; plus travel time.

h. **Run a weekend workshop for the teachers prior to starting**

i. **Consider including more upskilling in using the REPL environment during Boot Camp**

7.1 Future directions

As future possibilities are considered and how the program can be scaled up, there appear to be opportunities for *CS in Schools* to make a positive impact in regional and rural Australia, possibly with the use of online tutorials and videos as a scaffold for teachers where industry expertise may not be as accessible (e.g. remote Indigenous schools). Access to high speed internet in rural and regional Australia will however be an important consideration, perhaps videos uploaded to the cloud and downloaded by the teacher before class may be a practical solution to this (or the use of a USB). Consideration should also be given to the role that parents (who also might have expertise in computer science) can play in the program, considering that they are an important stakeholder in the school community that could value add to the program and might be highly motivated to participate. Of course, the expectations in terms of competence with the subject matter must remain, pragmatically, this might help the *CS in Schools* program run in places that are difficult for industry volunteers to access. Also worth mentioning in the involvement of senior leadership in the success of the program. When key decision makers are involved and excited about the potential of different initiatives in schools, they tend to receive more attention and success as a result. Ensuring that senior leadership in schools are actively 'on board' and part of all communications will be likely to positively impact the efficacy of the program in schools.

References

Bandura, A. (1977). Self-efficacy: toward a unifying theory of behavioural change, Advances in Behaviour Research and Therapy, 1(4), 139-161.

Belotto, M. J. (2018). Data Analysis Methods for Qualitative Research: Managing the Challenges of Coding, Interrater Reliability, and Thematic Analysis. *The Qualitative Report*, 23(11), 2622-2633. Retrieved from https://nsuworks.nova.edu/tqr/vol23/iss11/2

Catherine Jacques. C., Behrstock-Sherratt, E., Parker, A., Bassett. K., Megan Allen, M., Bosso, D. & Derek Olson, D. (2017). *Investing in What it Takes to Move From Good to Great Exemplary Educators Identify Their Most Important Learning Experiences* Washington: American Institutes for Research. https://files.eric.ed.gov/fulltext/ED577299.pdf

Cooper, G., & Carr, N. (2019). *Primary Pre-Service Teachers' Perceptions of STEM Education: Conceptualisations and Psychosocial Factors*. In T. Barkatsas, N. Carr & G. Cooper, STEM education : an emerging field of inquiry (1st ed., pp. 167-189). Netherlands: Brill Sense.

Foundation for Young Australians. (2017). *The new basics: Big data reveals the skills young people need for the new work order*. Retrieved from https://www.fya.org.au/wp-content/uploads/2016/04/The-New-Basics_Update_Web.pdf

Granor, N., DeLyser, L. A. & Wang, K. (2016). TEALS: Teacher Professional Development Using Industry volunteers, in proceedings of SIGSE 2016, Memphis, TN. DOI: http://dx.doi.org/10.1145.2839509.2844589.

Hobbs, L. (2018). *What now for STEM?* In T. Barkatsas, N. Carr & G. Cooper, STEM education : an emerging field of inquiry (1st ed., pp. 221-231). Netherlands: Brill Sense.

Jaipal-Jamani, K. & Angeli. C. (2017) Effect of Robotics on Elementary Preservice Teachers' Self-Efficacy, Science Learning and Computational Thinking, *Journal of Science Education and Technology*, 26, 175-192. DOE: 10.1007/s10956-016-9663-z

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: 10.1080/00330671.2012.667014

Poirier, Y. (2015). *JShell and REPL in Java 9*. Retrieved from https://blogs.oracle.com/java/jshell-and-repl-in-java-9

Rich, P. J., Jones, B. L., Belikov, O., Yoshikawa, E., Perkins, M. (2017). Computing and Engineering in Elementary School: The Effect of Year-long Training on Elementary Teacher Self-effiacy and Beliefs About Teaching Computing and Engineering, *International Journal of Computer Science Education in Schools*, 1(1). doi: 10.21585.ijcses.v1i1.6

Schawb, K. (2016). *The fourth industrial revolution: What it means, how to respond*. Retrieved from https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/

Schutt, R. (2012). *Investigating the social world : The process and practice of research* (7.th ed.). Thousand Oaks, Calif.: Sage Publications.

Schwab, K. (2017). *The fourth industrial revolution*. USA: Random House.

Sheffield, R., Blackley, S. & Moro, P. (2018). *A professional learning model supporting teachers to integrate digital technologies. Issues in Educational Research*, 28(2), 2018

Weldon, P. (2015). *The teacher workforce in Australia - supply, demand and data issues*. Camberwell: Australian Council for Educational Research. Retrieved 15 August 219 from https://research.acer.edu.au/cgi/viewcontent.cgi?article=1001&context=policyinsights

Weldon, P. (2016). *Out-of-field teaching in Australian secondary schools*, Camberwell: ACER

Yadav, A., Mayfiked, C., Zhou, N., Hambrusch, S. & Korb, J. T. (2014) Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Eduation*, 14(1), 5:1-5:16. doi:10.1145/2576872.

1. Materials
2. Barriers and issues
    i. Benefits
    ii. Future involvement

Para on traditional models of prof learning for teachers with some lit – largely off site but largely ineffective. This model brings the professional learning into the classroom environment in a highly supported way,

iii.

i. Experiences - workshop
ii. Benefits of participating
iii. Barriers to participating
iv. Future involvement

Research participants

1. Table of schools and participants

School	Context	Teacher	Industry Volunteer
School A	Metropolitan, independent school Haileybury	Matt Mark	Josh Jeff
School B	Metropolitan, Government school Eltham HS	Nellie natalie	Jeremy Julian Jason Jonatan
School C	Regional, independent school Sale Catholic College	Isabel Ineke	Zac??
School D	Metropolitan, Government school Greensborough HS	Gary Gus Susan Stella	Brendon Bryce
School E	Metropolitan, Government school Mt Erin College	Simon Sam	Darryl Darren Kieran Kevin

School F	Metropolitan, independent school Toorak College	Peter Phil Steven Stuart	Nina Nancy
School G	Metropolitan Government School	X	X