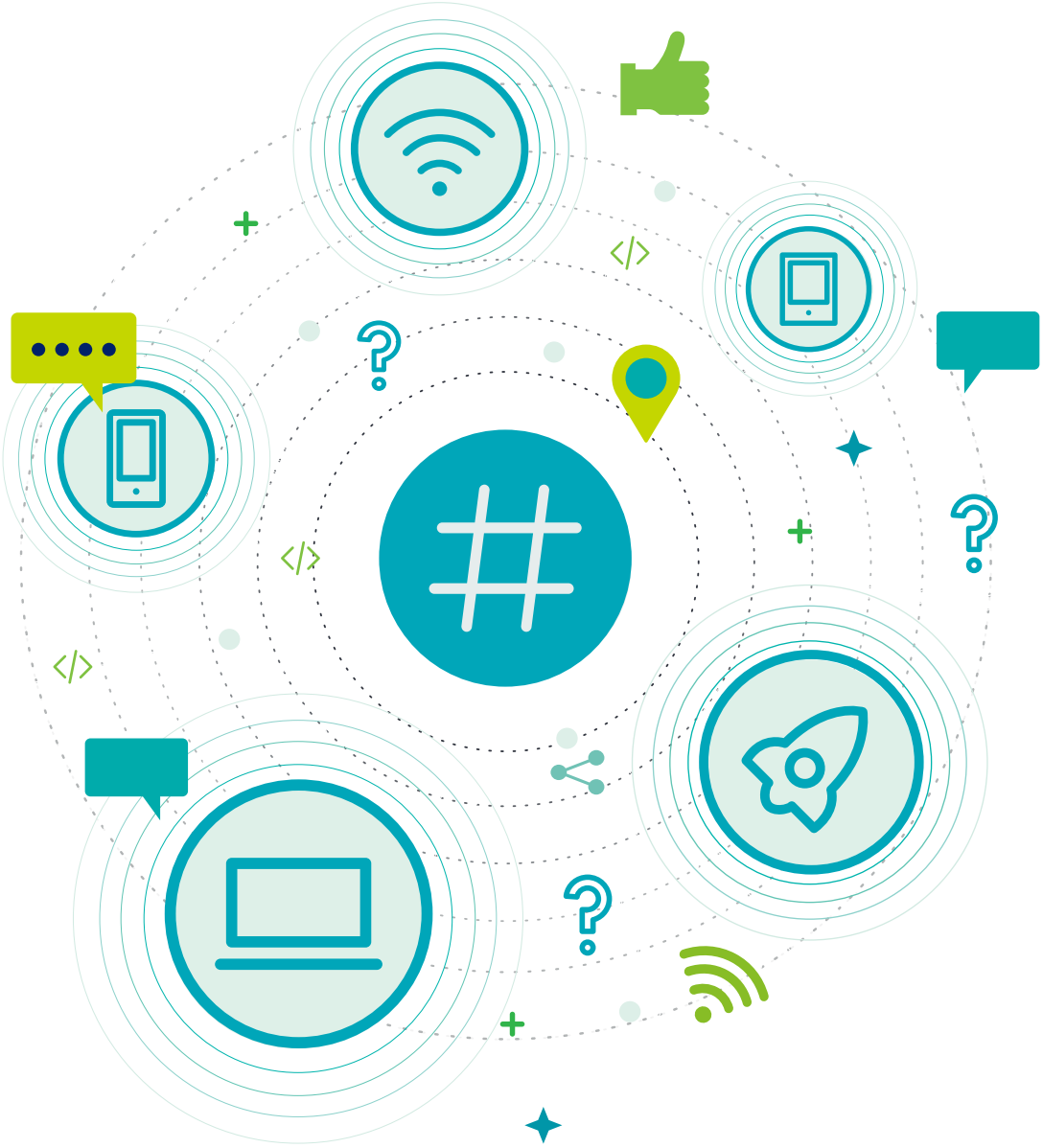
Deloitte.

GEELONG GRAMMAR SCHOOL®
EXCEPTIONAL EDUCATION



To code or not to code, is that the question?

2017

Deloitte Center *for* the Edge

Contents

Foreword

Digital technology has become an indispensable part of our social landscape. We are all struggling to manage its impact on our personal, professional and public lives and there is a palpable fear of being left behind if we cannot master it. As citizens, workers, employers, educators and parents, we want to respond but, how should we?

Adapting to digital technology, however, is not a problem to be fixed or a puzzle to be solved. The systemic nature of the technology also means that it is something the we need to respond to as a community, rather than individually. We need to work together to articulate the challenges and the opportunities that digital technology creates and decide how, as both a community and individuals, we can work together to create a future that maximises the benefits of technology while minimising the problems.

As the leader of Deloitte's national education practice, I have a deep interest in the ways in which education has and continues to be 'disrupted' by the emergence of digital technologies. It is a well-worn fact that digital technology is driving changes in education. Technology has facilitated the production of new models of teaching and learning that are shifting our definitions of what it means to be educated and also what it means to be an educator. At Deloitte, we are working with our education clients to assist them to keep pace with these changes. Our clients want us to base our advice on evidence rather than speculation. That is entirely understandable, but it is also important that we keep thinking beyond the horizon. One way that Deloitte ensures that it does this is through the work of the Centre for the Edge.

The Australian Centre for the Edge is an applied 'think-tank' supported by Deloitte; it is part of a larger Centre for the Edge network within Deloitte, based in Silicon Valley. The Centre explores the edges of what is known and understood; it speculates about the future, finding interesting ideas and bringing these back to the broader teams within Deloitte. The Centre's mission is to identify and explore emerging opportunities that aren't yet on our clients' agendas. The goal is to be speculative, and deliberately thought-provoking, but I want to emphasise that the work of the Centre for the Edge is grounded in data and observation.

Against this background, I was delighted when Peter Evans-Greenwood, a Fellow at the Centre for the Edge, and Tim Patston, Geelong Grammar School's Coordinator for Creativity and Innovation, decided to jointly turn their attention to education and the question 'To code or not to code?' Peter and Tim had noticed that while 'everyone should learn how to code' had become a catch cry, different stakeholders had their own interpretation of the phrase and its implications. There was no national, or even local, consensus on the meaning of the phrase, nor on how we should respond.

In response, Peter and Tim created a neutral forum where stakeholders from across public and private sectors – including educators, parents, employers and students – could gather to work through how digital technology will affect us all. It was also important for the forum to have a range of different stakeholder perspectives.

We are greatly appreciative of the participants who contributed their time, experience and opinions to the process, as without their contribution this project would not have been possible. It was also gratifying to see that participants approached each symposium with the spirt of cooperation that was intended, with open and robust discussions resulting in a positive and productive series of discussions.

The symposia showed that we were all struggling with the same challenges, as each conversation reached remarkably similar conclusions despite following very different paths. Digital technology was commonly seen as a valuable tool, but one which we're yet to fully understand and utilise. The symposia also highlighted the systemic nature of the challenge, and the need for a coordinated response if we want digital technology to be a positive force in our community.

We would like to thank Peter and Tim for their leadership on this project, and the Geelong Grammar School for the opportunity to be part of a unique collaboration the enabled us to explore important and timely questions. We hope that this conversation is the first of many as we work together to manage our transition to a more digital world.

Colette Rogers
Partner
National Leader, Education

Foreword

We live in a technological age. Students are surrounded by digital devices, from their 'smart' phone to their 'smart' watch. Teachers are being bombarded with digital products that claim to enhance the teaching experience.

While digital technology has become ubiquitous, the ability for it to be integrated into teaching has become more difficult. We run the risk that digital technology will subsume pedagogy, as information becomes cheaper and cheaper.
How should we respond to this rapidly changing world?

Geelong Grammar School has long striven to contribute to the national and global agendas of education. It was therefore very pleasing when our Coordinator of Creativity and Innovation, Dr Tim Patston, and Peter Evans-Greenwood from Deloitte's Centre for the Edge, proffered the idea of devising and running a national series of symposia to discuss ramifications of the digital age and how it is affecting government, industry and education. This partnership has lent a unique perspective to this project.

We felt that it was a very good time to have a close look at coding and digital literacy, which are increasingly influencing how and what we teach. It seemed logical to reach out to a diverse range of stakeholders from across Australia, to take a snapshot of where we are as a nation at this critical time of change.

We would like to express our gratitude to all of the participants, who gave freely of their time and expertise to ensure the success of this project. It is a credit to the individuals who took part, but also their organisations, from school and tertiary education, government and the private sector. The discussions were robust and fruitful, given in the true spirit of cooperation.

Perhaps most surprisingly, this report shows that all sectors are struggling with the issues of coding and digital literacy. Most organisations felt that, while digital technology could be very useful, it is still poorly understood and underutilised in many fields. As devices become smarter, it seems that we run the risk of people becoming less able to understand the processes behind the devices they are using and consequently less able at key skills such as creativity and problem solving.

This report shows that it is time to take stock. In the world of education, pedagogy must come first; giving our students' skills and attitudes that will enable them to flourish in an ever-shifting landscape of study and work. Technology, digital or otherwise, must be a tool that is understood and utilised for a greater purpose, not merely to do basic functions with ignorant operators.

Thank you to Tim and Peter for coordinating this project and to Deloitte for working with us to explore this valuable topic. We hope that this work will inspire a response from all stakeholders, which can improve how we best manage the digital world.

Stephen Meek
Principal, Geelong Grammar School

Introduction

Digital technology permeates the modern world. We can order dinner via an app on our smartphone or view a university lecture on a tablet, both from the comfort of our couch. We can call a taxi with a touch of a button. Indeed, we might even work for a computer[1] with a digital dispatch service directing us to pick up a passenger, meal or package, and our performance is rated by a combination of survey and algorithm. Many people are becoming increasingly concerned that we'll be replaced by computers, with artificial intelligence (AI) supplanting people for many complex tasks from playing Go (an abstract strategy board game) to writing news articles or providing investment advice. Digital technology is used in all aspects of our personal, professional, public and recreational lives.

In the last ten years – roughly since the birth of the modern smartphone – we have adopted digital technology at a startling rate. Smartphones and YouTube quickly flipped from novel to ubiquitous.

How we relate to the world has changed in the process, with the new digital tools becoming a lens (or, as some worry, the lens) through which we experience reality. For instance, why remember what you can Google?

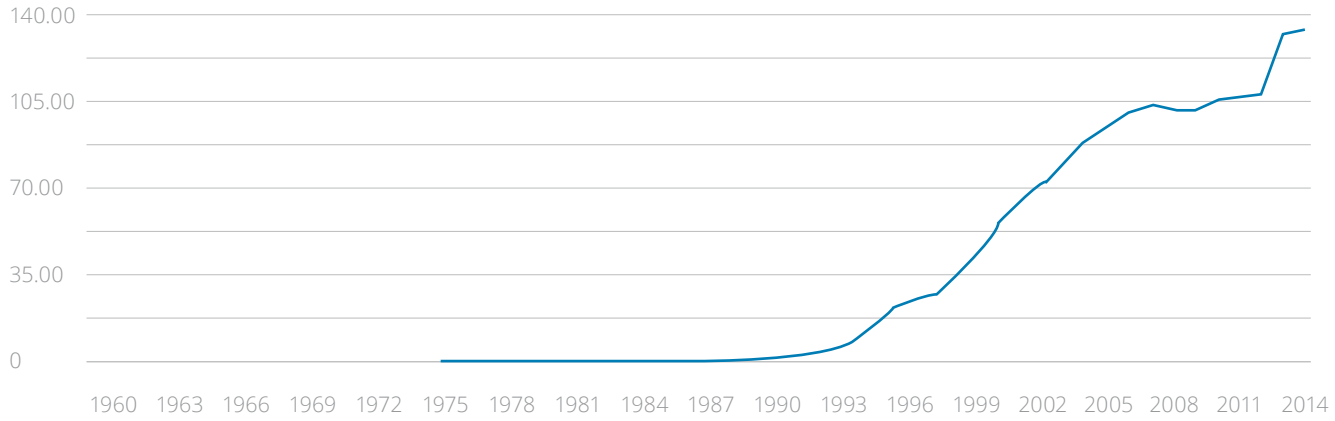
The digital environment that we're creating has led many people to conclude that we either create digital solutions or be replaced by them. There is also talk of the need to introduce a universal basic income[2] to support the people replaced by the robots. This raises the question: should everyone learn to code?

Over the last generation programming has moved from a niche activity – a challenging task that's the responsibility of a small digital elite – to a mainstream one. As digital technology becomes increasingly ubiquitous in everything from cars to coffee machines, is our education system keeping pace?

Fixed broadband subscriptions (per 100 people)

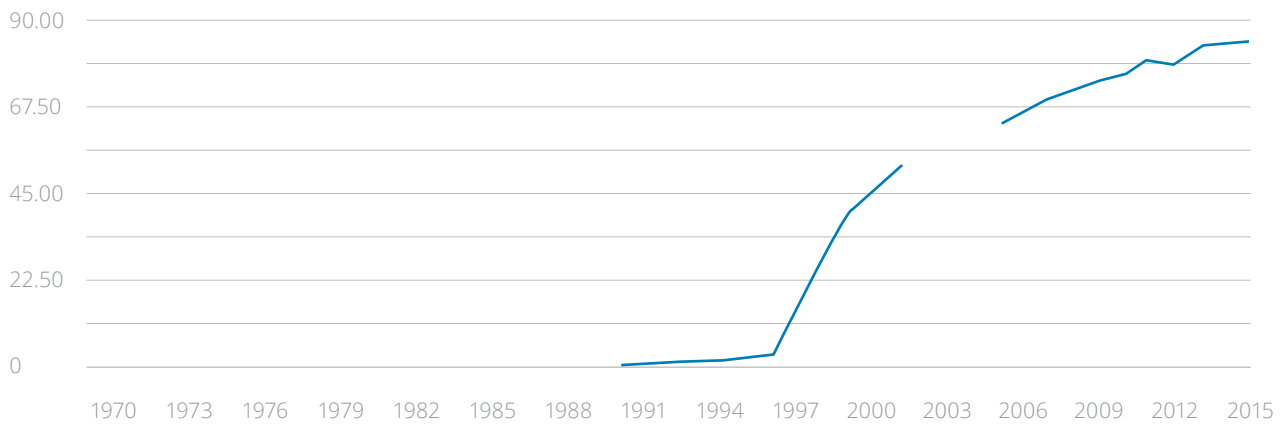
Source: The World Bank, World Development Indicators[3]

Mobile cellular subscriptions (per 100 people)



Source: The World Bank, World Development Indicators[4]

Internet users (per 100 people)



Source: The World Bank, World Development Indicators[5]

Our ability to shape our future relies on our digital competency – our ability to control computers and turn them to our will – and the best form of digital competency is coding, or so it's assumed. Why shouldn't learning to code be an important part of our formal education, possibly even supplanting learning a foreign language, as isn't coding the language of our digital future? Everyone from celebrities through to presidents and prime ministers and even educators has jumped on the 'everyone should learn to code' bandwagon.

Significant effort has been devoted to update the national curriculum and to meet the growing demands of digitally literate students who have a deep understanding of information systems. The Australian curriculum includes the general capability of information and communication technology (ICT) and the subject digital technologies, which incorporates coding.

ICT capability involves students learning to make the most of the digital technologies available to them, adapting to new ways of doing things as technologies evolve and limiting the risks to themselves and others in a digital environment.[6] Students learn computational thinking, design thinking, algorithmic thinking and systems thinking to cultivate digital skills. From year three, students begin to code using a visual programming language and by year 7/8 they progress to a general-purpose programming language to implement their solution designs.

Computer programming –known as coding – is the formulation of a problem into an executable computer programme. Different methods can be used to code; from imperative instructions using logical languages where problems are described in terms of axioms and rules to mathematics-based functional languages. Programming language can also vary from low-level languages, which are close to the detailed instructions used by a computer through to high-level languages that are closer to the problem domain. Concepts such as 'modules' and 'objects' are also used to help organise larger programmes.

However, while 'everyone should learn to code' is an attractive call to action, it might not be the whole solution. There are the practical concerns of which computer languages to teach and how to best teach them and the fact that our relationship with digital technology is evolving. Our frame of reference is the past, when there was a clear distinction between using a computer and making a computer.

We're forging a new relationship with digital technology. Today we might find ourselves working for a computer such as ride sharing drivers or pick-and-pack workers in a distribution centre. Or we might work with a computer, collaborating as peers, as humans do in freestyle chess, where success is not determined by our understanding of digital technology, but on our ability to work with the technology to find an optimal solution.[7] Or we might work on computers, where we are adapting and combining digital components, or encoding domain knowledge in smart digital components, to create a solution.

Our relationship with computers is evolving into a more collaborative and conversational one. For instance, in freestyle chess, humans collaborate with the computers, and with each other, as they work on the next move. Similarly, Lee Sedol, the second ranked professional Go player in the world, after being defeated by the Go computer AlphaGo, helped to improve the solution by conversing with it, training it. Many coding jobs are also being automated, either via the development of components that capture domain knowledge or via code synthesis, where a computer program is generated (by another computer program) based on a high-level description of the problem to be solved.[8] Indeed, Google is already undertaking to retrain its army of coders in this new 'noncoding' paradigm;[9] simply being a coder is no longer enough.

Learning to code can mean different things to different people and this was the reason we asked the question 'should everyone learn to code?' The conversation appears to have skipped directly from the problem – the need to remain relevant in a digital environment – to the solution – teaching everyone how to code – without reflecting on what we mean by coding. After all, the question seems simple. We all use digital technology on a daily basis (or it uses us), so it must be good. We use digital technology in education and in the workplace therefore we must be educated. And coding is the language of computers so it must be the essential skill in the digital age.

The goal of this project was to explore this thinking by asking:

- What are the intentions and aspirations behind 'learn to code'?
- What educational and social outcomes we should be striving for?
- Are there key skills from 'learning to code' not covered in the current curriculum?
- Is there a better definition for digital literacy?
- How does digital literacy relate to coding and the rest of computer science?
- How do we demystify digital technology and bring the community along?

The journey

While most stakeholders agreed that 'learn to code' is a vague, unquantifiable statement, many expressed different views on how learning to code will benefit the next generation. Computer professionals have a particular view based on their education and experience (and it's not uncommon to find one group of computer professionals that consider the work of another not to be coding). For parents, it represents the hopes and aspirations they have for their children in a society infused with digital technology. Employers have more practical concerns, and are looking for individuals with the skills they need be productive in an increasingly digital workplace. Educators are concerned with the practicalities of crafting tangible learning outcomes and packing them into a crowded curriculum, and have invested significant effort into breaking 'learning to code', and digital technology in general, into a well-defined and teachable set of concepts, knowledge and skills.

The challenge in understanding the statement 'everyone should learn to code' is not in forming an opinion. Each group has already done this independently. The challenge is to align the thinking and efforts of each group. What does the business community mean when they ask for graduates with digital skills? Are these the skills we are teaching in the curriculum? Does this address parents' angst for their childrens' future? Are we trying to develop competencies in the use and application of current (and soon emergent) software tools and apps? Or are we developing understanding, resilience and curiosity in digital competency so learners of all ages can span the boundaries in the pedagogical (and andragogical) domains?

Five symposia of ~20 people in five cities across four states providing ~100 points of view comprising:

- ~50 educators, K-12 through to tertiary
- ~15 from government
- ~40 from industry.

A series of symposia
In 2016 we held a series of symposia for educators, policy makers and industry to share their aspirations and experience, with the hope of reaching a consensus on what, as a community, we mean by 'everyone should learn to code'.

Each symposium was conducted as a round table using the Chatham House Rule to encourage a frank and open discussion. Participants were not required to identify themselves or their affiliations.

We developed a series of questions to initiate conversation and to ensure that the topic was adequately explored. The conveners did not participate to avoid prejudicing the conversation.

It's worthwhile noting that while each symposium reached similar conclusions, they all took quite different paths.

What do we mean by coding?
Each session began with the question 'What do we mean by coding?' Computer professionals typically highlighted how developing software should be seen as a team sport, with coding as only one of many skills required to create a finished solution: there's much more to coding than 'coding'. Other participants expressed how 'coding' was shorthand for a larger set of skills required to have a fulfilling and productive role in a digital environment.

Participants agreed that 'learning to code' should be interpreted as 'having the knowledge and skills required to use computers in the pursuit of work'. In this context 'computers' should be understood as any digital tool – either a physical device or a cloud- or network-based application – that can be configured and shaped by the user. This might range from configuring email folders, scripting an animation, creating a simulation or a website (possibly from a template) through to tabulating data in a spreadsheet or operating a 3D printer. 'Work' should be understood as 'any activity in pursuit of an outcome' and includes hobbies, interests and pastimes as well as paid employment.

There was a consensus that coding – programming a computer – is just one component of the broader concept of digital competency. There is a gap between the knowledge and skills, application and understanding required to be digitally competent that is represented by the desire that 'everyone should learn to code'. Similarly, learning a programming language is not enough. Students must also learn the affordances – or the properties of an object that make it clear how the object can or should be used such as the handle on a teapot, – provided by the application programming interfaces (APIs) and physical interfaces of a digital tool if they are to be digitally competent.

It was agreed that we need a 'computers' stream through the education system for the students who want to pursue computers in work. There is also an important role for a coding unit in the middle years of education, possibly late primary with a refresher in secondary, to demystify coding. Some students unfamiliar with coding will find the unit interesting and take up the computing stream. Others will gain the knowledge they need to work with coders. Several participants noted, 'This is more for the kids at the bottom or in the middle, than at the top'. Motivated students are typically ahead of the curriculum.

It was agreed that making all students competent coders is not the problem we need to solve. The challenge is to make students comfortable with digital technology and tools so that they can adapt them to their work. Younger students, primary students in particular, must discover that computers are a tool for learning. They must also understand the limitations of digital technology. Learning is a process that develops over time. Asking a computer to calculate a formula or find an article does not mean that you understand the processes behind finding the answer. Nor can you assume that the computer is infallible. We need to teach children from a young age that computers do not always 'know' the right answer.

What do we mean by digital literacy?
We asked participants to define digital literacy. Many expressed concern that digital literacy is usually interpreted as 'the ability to use computers' which was seen as vague and instrumental, rooted in a world-view that has us either using or making computers. A better approach is to define digital literacy in terms of knowledge and skills, but there was no clear consensus on how to frame these knowledge and skills.

What distinguishes digital technology from other technologies is that it is hard for the layperson to understand how it works, while the digitally literate can make digital technology do nearly anything. Consider this. It is easy to discover the affordances of a piano, the possible actions we can take. We can plunk a key or pluck a string. We can change the sound of a piano or how it is played for example by wedging cutlery between the strings or pieces of paper between the hammers. Digital technology is different. It can be difficult to discover the affordances of a digital tool, but at the same time digital tools are much more adaptable. Digital technology enables us to think about problems using different solutions. Working with digital tools is more akin to co-creation than utilisation, with the resulting solution a fusion of the user's knowledge and the knowledge embedded in the tool.[10]

Discussion centred on the need to provide students with the knowledge and skills to discover the affordances provided by digital tools, and how these tools can be reused, recombined and repurposed to solve the problem at hand. This concept is neatly expressed in the term bricolage, a word of French origin that means the creation of something from a diverse range of available things.

A common problem identified by participants was that 'digital literacy is commonly a consumption model'. Students are learning to use digital tools to create works, but they are not learning how to bricolage with digital tools, how to combine the affordances provided by different tools to create a new tool. Or as a participant suggested: 'Most employees can solve practical problems in the workplace, such as needing beer and snacks for Friday afternoon drinks – if I give them fifty dollars they can find beer and snacks. However, if there is a digital technology problem, even a basic problem with their device, they cannot solve the problem unless there is an app for it.'

Most technologies are simple, with few affordances. For instance, applying a clothes peg to a piano string requires us to observe the string within the piano, and source and apply a clothes peg. With digital technology, affordances are hidden and we have no knowledge of the materials used in construction, which makes digital tools difficult to understand and intimidating to the novice user. User-friendly interfaces and smarter devices alleviate this problem by reducing the numbers of affordances and making them more obvious, but they also constrain what the user can do. A common observation among participants was that 'the smarter the devices, the dumber the user'.

Digital literacy means much more than the ability to find, use, share and create content using technology. It is also the ability to create new solutions from digital tools and to adapt the tools to our purpose. It implies the ability to use ingredients and processes, combined with techniques to apply to a variety of different contexts, as well as spanning the boundaries between different knowledge domains, to bricolage.

'The inseparability of humans and media sustains the idea that students and tools are agents performing knowledge in coaction, while approaching mathematical problems. Looking at the solution to a problem is seeing a fusion of the solver's knowledge and the tool's built-in knowledge, rather than an aggregate of both or a complementarity between them'

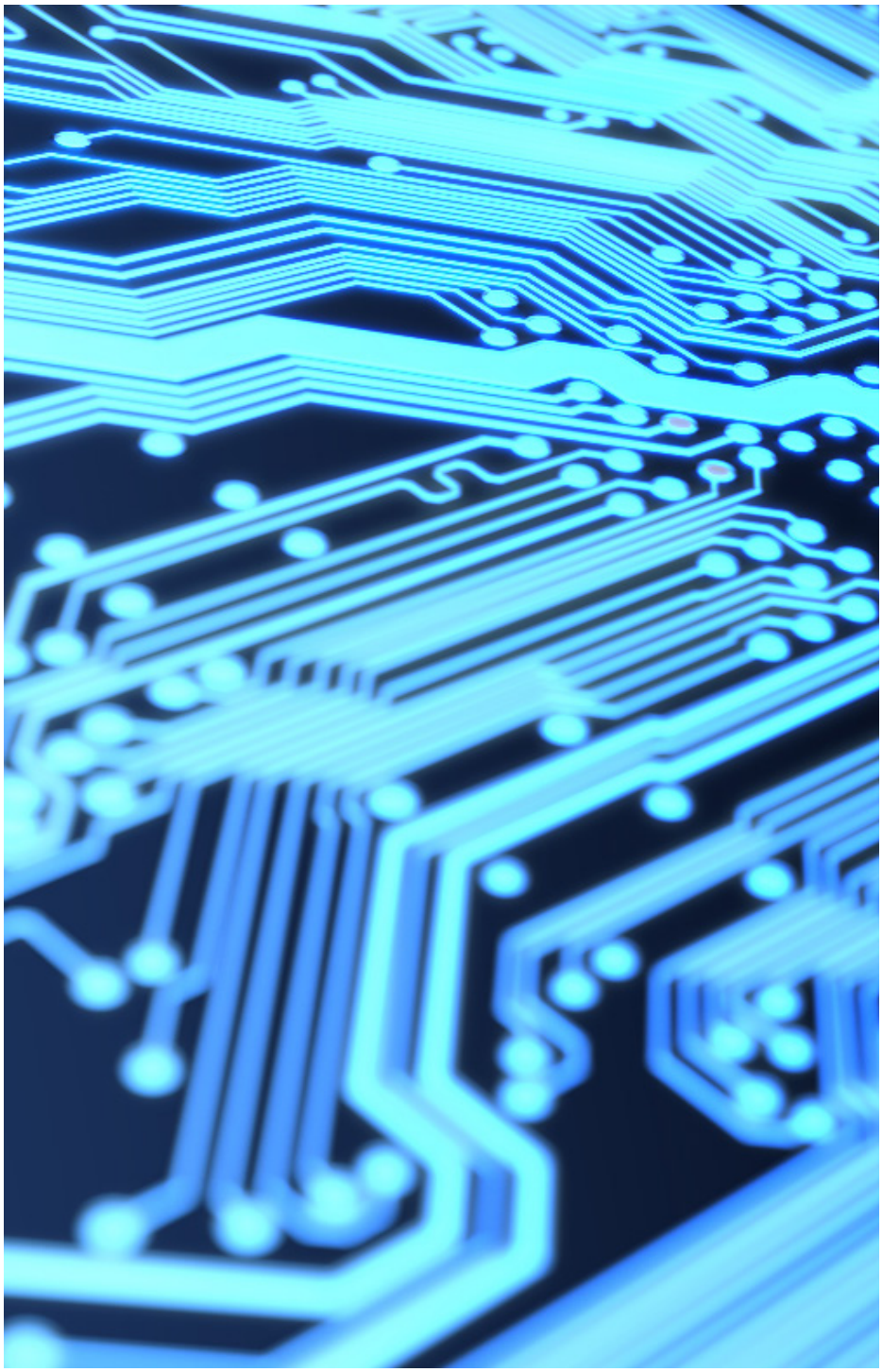
— Carriera, S et al. (2016), Youngsters Solving Mathematical Problems with Technology, Springer p. 276

Are there key digital skills missing from the current curriculum?

It was recognised that there is a gap between the digital skills and knowledge provided by current teaching and the needs of the workplace. Despite many participants discussing the concept of 'digital natives',[11] it was clear that this term is misleading, as familiarity with digital technology from a young age does not translate into digital competency. Employers highlighted what they thought were the most basic skills required by the workplace, 'can you type, use a browser, use a spreadsheet to add numbers, write relevant and appropriate email'.

This problem was not confined to the workplace. Universities also expressed frustration at the low level of digital competencies demonstrated by their students, in particular using spreadsheets and search engines. Too many people appear to use these tools as 'hammers', rather than developing a bricolage solution to a problem. This severely limits not only the ability to do basic tasks at university and in the workforce, but leaves people unable to construct effective solutions to problems. This idea is combined with the myth of the IT-technician as magician.

This basic lack of understanding leads people to a state of 'learned helplessness' in some key areas. As one participant put it, 'as technology gives black and white answers, people aren't open to failure'. There is also the sense of paralysis when digital technology does not do what people would like it to do. Rather than solve the problem, there is a tendency to wait for the magician to arrive and fix it. If we do not understand where technology comes from and how it works, then we can't navigate the technology landscape. What we are looking for is attitudes and behaviours, or values, in addition functional use.



Observations

Each symposium followed a different path, driven by the participants, though all ended up at a remarkably similar conclusion.

Should everyone learn to code?
There was strong agreement that 'everyone should learn to code' should be interpreted as everyone should learn how to make the best use of digital technology to create their own solutions. The form these solutions take, and the technologies they use will vary depending on the problem that the individual is prosecuting and the domain they are working in. Sometimes the solution will require the development of a software program – 'coding' in the traditional sense – though often it won't.

It is important for the K-12 curriculum to support a coding stream to ensure all students have some exposure to coding to help demystify digital technology.

Familiarity does not imply competency
It's assumed that familiarity with digital technology translates into digital competency. However, experience has shown that this is not true and that the familiar narrative about tech-smart young people, 'digital natives', is false. Participants pointed out 'we've been teaching [digital] technology like we teach to drive a car'. The problem is that for most of us a car is simply a tool, a means to get from A to B, and it wouldn't matter if the bonnet was welded shut or the car was completely autonomous. Digital technology is different to digital tools, and familiarity with digital tools does not confer digital literacy, nor does digital literacy confer digital competency.

The power of digital technology lies in its ability to be shaped, taking a collection of digital (and non-digital) tools and adapting them to a new purpose or moving them to a new context, bricolage. There is currently no teaching of the teachers in this in all sectors from primary through to post-secondary. Teachers expressed frustration at how quickly their organisations adopted new technologies but how slowly, if at all, they supported staff in adapting them to their specific context. It's easy to make sweeping statements about the need for a digital curriculum, but the task of integrating digital technology into individual subjects and topics is much harder, and more important. Digital technology can no longer lead pedagogy. Tools need to be selected appropriate for use, and teachers need to be provided with the time, support and inspiration required to adapt the tools and ideas to their own subject area and classroom.

Integrating digital content into non-digital subject areas is challenging
Students need digital skills and so do teachers. Both students and teachers also need to develop the ability to bricolage digital solutions in their chosen subject areas.

Education now has a strong vocational focus, but the rapidly changing workplace puts a premium on the prepared mind, and standardised testing does not support the practice of divergent thinking. A common complaint among participants was that 'we've perfected the skill of getting kids into university, but lost the skill of preparing them for university'. The goal must be for students to be able to integrate ever-changing digital technology into their work. This requires a new approach in education.

Significant work has already been done on capturing the concepts, knowledge and skills behind digital technology, and there are rich resources such as the Victorian Digital Technologies curriculum[12] and Australian Curriculum: Critical and Creative Thinking Capabilities.[13] The Victorian Digital Technologies curriculum was designed in a modular fashion to allow content to be exported into other subject areas.

The problem, however, is to understand how the digital content can be effectively integrated into a non-digital subject to provide the desired learning outcomes. Relevant core knowledge and skills must be retained: many are domain specific, but some 'digital skills' are not, as with the creative and critical thinking curriculum, and need to be sequenced. These must include 'resilience and appreciation of the need for iteration, developing the capacity to fail'. Unfortunately work in this area is in its infancy.

New attitudes and behaviours are required
There was a clear consensus that we must focus on instilling the attitudes and behaviours that the students will need to thrive in a world where the half-life of both knowledge and skills appears to be rapidly shrinking. 'The skills will change but we need to give them capability and confidence to succeed'. Students (and workers) need to develop processes enabling them to discover and adapt new ideas and tools to the task at hand, rather than relying on (periodic) formal education to provide them with the knowledge. We need to consider how we weave ICT, digital and creativity together.

Conclusion

Education is often seen to be lagging behind technological change. This is unjustified, as significant effort has been invested in packaging digital skills into the education curriculums. However, our relationship with digital technology is evolving and the community is still learning what it means to be digitally competent. This was reinforced in the different meanings read into the catchphrase 'everyone should learn to code'. Our aspirations are much broader than teaching everyone how to code: we want to ensure that everyone has 'the knowledge and skills required to use computers in the pursuit of work'. This means understanding the opportunities digital technology provides as well as its limitations. While only some of us might need to code, all of us need to be digitally competent, and our current approaches to teaching digital competency are insufficient. We can see this in three key points that were common across all symposia.

First, there is a common but incorrect assumption that familiarity with digital technology results in digital literacy and competency. This assumption is behind myths such as that of 'digital natives'. Being comfortable with a digital tool does not imply sophisticated use. Likewise, being digitally literate does not imply an understanding of the benefits and problems of digital solutions. Despite being repeatedly shown to be incorrect the myth persists that the young have an innate understanding of digital technology and don't need to be taught. This same assumption holds for teachers, with participants revealing that most 'teachers teaching digital technologies have less than one year of technology training'.

Second, there is a need for a digital technology stream in K-12 to support digital specialists, individuals who focus on developing digital technology and who want to learn how to code. While most students will not choose to become digital specialists, it is important that all students are introduced to coding to demystify the technology and to expose them to a career path that they might not have otherwise considered.

Third, and finally, we need to understand that it is not possible to teach digital competency as an isolated subject: it needs to be integrated into a broad range of subjects and across the various educational strata as the application of many digital skills is domain specific. While some students may know how to code, they don't have the skills to identify and solve problems or to understand or control the mechanics of the solution, nor do they understand the limitations of digital technology. Skills need to be exported from the digital curriculum and imported into other subjects where students can learn how to bricolage solutions. We need to address the broader issues of a lack of practical problem solving skills, the inability of many students to use digital technology to solve a problem unless 'there's an app for that'. The focus needs to be on providing students with the practical skills to enable them to solve problems in context, bringing the digital technology to the problem (where appropriate) rather than the problem to digital technology. The challenge is to 'create education components that live up to the reality of the kids' age', without simplifying the concepts and skills so the educational benefit is lost.

Participants highlighted the collaborative and cross-disciplinary nature of digital technology. Statements such as 'it's a collaborative rather than a creative subject' and 'one of the things we know about this stuff is that it shouldn't be taught from the textbook as it's non-linear' were common. Digital technology is not an island, but is often treated as one. There is a need for well-considered and well-taught integration into all subjects. Students need to understand both the potential and the limitations of digital technology. They need to be familiar with the technology's affordances, the opportunities it provides, and to have the ability to combine digital and non-digital components to make new solutions. They need to understand that failure and iteration are an essential part of the process. Formal education must provide students with a series of increasingly sophisticated challenges that require them to bricolage solutions using their knowledge and digital skills, integrated into their non-digital subjects.

These digital skills must also cover more than different ways of programming.[14] Emerging technologies such as functional programming and training-based approaches such as machine learning need to be included. Students need to explore the differences in these approaches (such as how any large collection of rules can be nondeterministic due to unexpected rule interactions, or how it's impossible to determine why a machine learning solution came to the conclusion it did), as these technologies will increasingly determine how they are measured in the workplace and their ability to engage with the broader community. Students must also learn the implications of working for, with or on a digital solution. As freestyle chess demonstrates, our ability to make the most of the capabilities and knowledge embedded in a digital tool such as a chess computer does not depend on depth of domain knowledge or knowledge of the underlying technology; it is a unique skill in and of itself.

The challenge of teaching everyone to code means that teachers need the support of the broader community. Employers and policy makers need to identify good examples of digital competency, and flesh out what it means to bricolage in different industries and domains. Teachers must be supported with targeted professional development to help them understand the why and how of what they're doing, and they must be trained in using the digital components within their subject areas.

While the groundwork has been laid and a sophisticated digital curriculum already exists, what is missing is an understanding of what digital competence means in practice; what application looks like in a world where society is forging a new relationship with digital technology.

Next steps

Participants agreed that the symposiums were a valuable and rare opportunity for stakeholders from different parts of the education sector to work together on a problem in which they are all deeply invested. Many participants want to keep the conversation going. This report is the first step in understanding whether everyone should learn to code. Deloitte Centre for the Edge and Geelong Grammar School will continue this conversation in a second series of symposiums to explore the question 'What does digital competence look like in practice, in different domains and across the educational strata, including in work'.

Participation

Andrew Baylis
Ben Beaton
Melissa Cadzow
Ivan Carlisle
Paul Clapton-Caputo
Benjamin Champion
Christopher Cheong
Andrew Chester
Paula Christophersen
James Dellow
David Dimsey
Amanda Dodson
Fiona Fitzgerald
Sanda FitzGerald
Daniel Groenewald
Raphael Hammel
Stephen Harvey
Jacki Hayes
Nick Heng
Renee Hindmarsh
Stella Jinman
Wendy Johnson
Therese Keane
Damian Keegan
Bernardine Knorr
Jennifer Mumford
Margaret Northcote
Tina Photakis
Carol Puddicombe
Kynan Robinson
Jessica Sashegyi
Kate Sonter
Kay Stacy
Lachie Stevens
Rosemary Stockdale
Elizabeth Stone
Charmaine Taylor
Grace Tuttle

Endnotes

1. This is an expansive definition of 'computer', including networked or cloud applications as well as servers and standalone devices such as personal computers, tablets and modern smartphones.

2. A universal basic income (UBI) is a form of social security that guarantees that all citizens or residents of a country receive a regular payment, regardless of other income or wealth.

3. The World Bank, World Development Indicators (2016), Fixed broadband subscriptions (per 100 people) [online]. Available at <http://data.worldbank.org/indicator/IT.NET.BBND.P2> [Accessed 27/02/2017].

4. The World Bank, World Development Indicators (2016). Mobile cellular subscriptions (per 100 people) [online]. Available at <http://data.worldbank.org/indicator/IT.CEL.SETS.P2> [Accessed 27/02/2017].

5. The World Bank, World Development Indicators (2016), Internet users (per 100 people) [online]. Available at <http://data.worldbank.org/indicator/IT.NET.USER.P2> [Accessed 27/02/2017].

6. Australian Curriculum, introduction to ICT general capability, available at <http://www.australiancurriculum.edu.au/generalcapabilities/information-and-communication-technologycapability/introduction/introduction>.

7. Freestyle chess is a chess competition that allows any combination of human and computer chess players to compete as two teams. When IBM's Big Blue beat chess champion Garry Kasparov in 1997 Kasparov wondered what would happen if instead of competing against one another humans and machines collaborated? The first freestyle chess competition was played in 2005 and resulted in an upset victory later reflected upon by Kasparov:

The surprise came at the conclusion of the event. The winner was revealed to be not a grandmaster with a state-of-the-art PC but a pair of amateur American chess players using three computers at the same time.

Their skill at manipulating and 'coaching' their computers to look very deeply into positions effectively counteracted the superior chess understanding of their grandmaster opponents and the greater computational power of other participants. Weak human + machine + better process was superior to a strong computer alone and, more remarkably, superior to a strong human + machine + inferior process.

Human strategic guidance combined with the tactical acuity of a computer was overwhelming.

Garry Kasparov (11 February 2010), The chess master and the computer, [online] New York Review of Books.

Available at <www.nybooks.com/articles/2010/02/11/the-chess-master-and-the-computer/> [Accessed 27/02/2017].

8. It's interesting to note that Deep Learning is also being applied to the problem of program synthesis, with Microsoft announcing progress using deep learning to enable 'non-coders' to create programs via interacting with a neural network.
Singh R., Devlin J., Mohamed A., and Kohli P. (21 April 2017), Deep Learning for Program Synthesis, Microsoft Research Blog. Available at <https://www.microsoft.com/en-us/research/blog/deep-learning-program-synthesis/>.

9. Levy, Steven. 'How Google Is Remaking Itself as a 'Machine Learning First' Company.' Backchannel, June 22, 2016. Available at <https://backchannel.com/how-google-is-remaking-itself-as-a-machine-learning-first-companyada63defcb70>. [Accessed 13/06/2017]

10. See Carriera, S et al. (2016), Youngsters Solving Mathematical Problems with Technology, Springer p. 276, for a discussion of the integration of digital technology into mathematical problem solving.

11. Marc Prensky coined the terms 'digital natives' and 'digital immigrants' in his essay Digital Natives, Digital Immigrants published in On the Horizon [online] (MCB University Press, Vol. 9 No. 5, October 2001). Available at <http://www.marcprensky.com/writing/Prensky - Digital Natives, Digital Immigrants - Part1.pdf> [Accessed 27/02/2017].
12. The Victorian Digital Technologies curriculum is available at <http://victoriancurriculum.vcaa.vic.edu.au/technologies/digital-technologies/introduction/rationale-and-aims> [Accessed 27/02/2017].

13. The Australian Creative and Critical Thinking curriculum is available at <http://www.australiancurriculum.edu.au/generalcapabilities/critical-and-creative-thinking/introduction/introduction> [Accessed 27/02/2017].

14. Peter Denning explores the distinction between coding and computational thinking in Denning, Peter J. 'Remaining Trouble Spots with Computational Thinking.' Available at <https://cacm.acm.org/magazines/2017/6/217742-remaining-trouble-spots-with-computational-thinking/fulltext>. [Accessed June 13, 2017.]

Contacts

Peter Evans-Greenwood
pevansgreenwood@deloitte.com.au
Deloitte Centre for the Edge
550 Bourke Street
Melbourne Victoria 3000

Tim Patston
tpatston@ggs.vic.edu.au
Geelong Grammar School
50 Biddlecombe Avenue
Corio Victoria 3214

Deloitte.

MCBD_HYD_08/17_054714