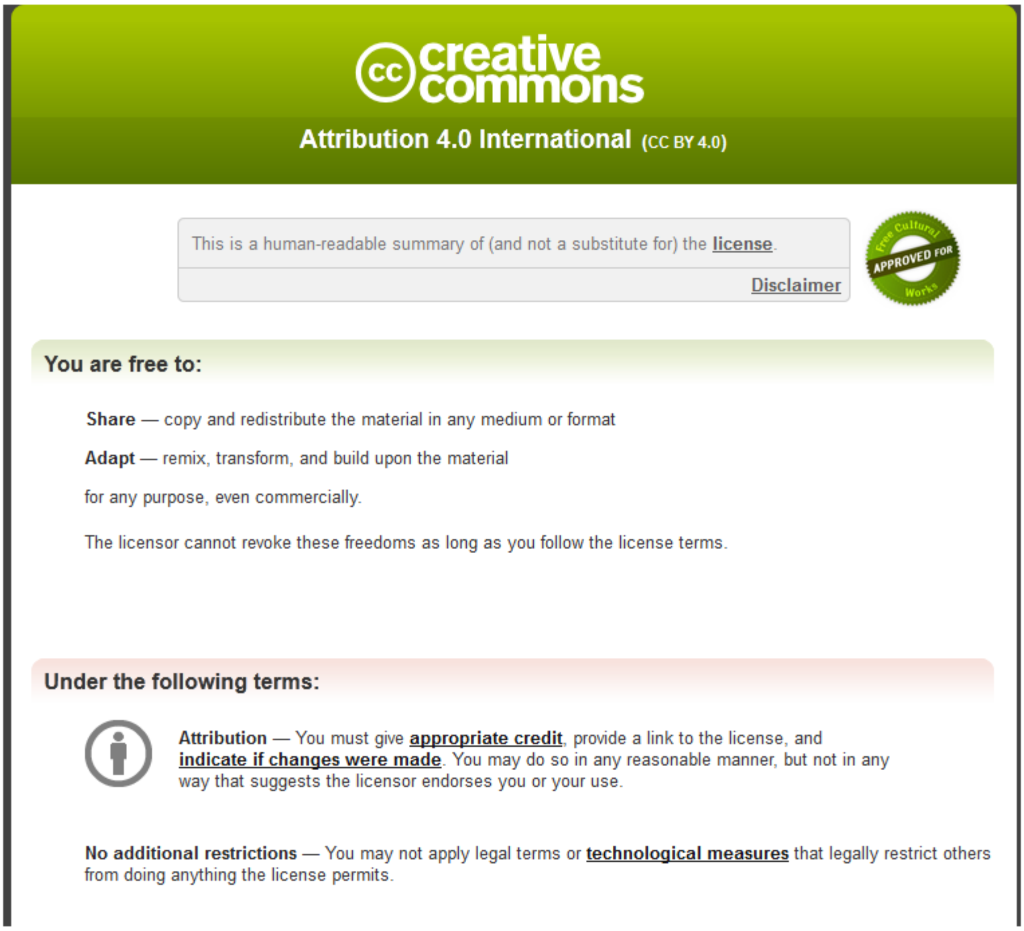

Free Cultural Works
APPROVED FOR

THE UNIVERSITY *of* ADELAIDE

Resource Review & Report

# Coding Across the Curriculum

CRICOS PROVIDER 00123M

Computer Science Education Research Group

A/Prof Katrina Falkner & Dr Rebecca Vivian

School of Computer Science,
The University of Adelaide,
Adelaide, South Australia 5005

c/o: A/Prof Katrina Falkner
Phone: (08) 8313 6178
Fax: (08) 8313 4366
katrina.falkner@adelaide.edu.au
http://blogs.adelaide.edu.au/cser

# Coding Across the Curriculum

## Resource Review

*Final Report*

March 2015

# TABLE OF CONTENTS

# INTRODUCTION

## Overview

On the 14th of October 2014, the Australian Government released the Industry Innovation and Competitiveness Agenda[1] that aims to strengthen Australia's competitiveness. One of the major announcements at this time was the introduction of the Coding Across the Curriculum Program.

This program will aim to support teachers to implement the Australian Curriculum in classrooms, to provide students across various year levels with the opportunity to learn about algorithmic thinking, computational thinking and coding in relation to the Australian Curriculum: Technologies and other relevant key learning areas.

The Australian Government has been working with the Australian Curriculum, Assessment and Reporting Authority (ACARA), who has been established to oversee the development of the Foundation (F) to Year 12 Australian Curriculum. One of the Australian Curriculum learning areas, that explicitly explores computational thinking, algorithmic thinking and coding, is Technologies.

## The Australian Curriculum: Technologies

The Australian Curriculum: Technologies will aim to ensure that all students benefit from learning about and working with traditional, contemporary and emerging technologies that shape the world in which we live.

The F-10 Australian Curriculum: Technologies has been fully developed and published on the Australian Curriculum website[2] as 'Available for use; awaiting final endorsement' and Australian states and territories can use the curriculum, as they choose. The Curriculum content descriptors are organised around a series of Year level 'bands', from Foundation to Year 10, with senior year levels (Year 11 and 12) awaiting final development. For reference, the following is a guide of age groups in accordance with the Australian year level bands:

- F – Year 2: typically students from 5 to 8 years of age
- Year 3 – 6: typically students from 8 to 12 years of age
- Year 7 – 10: typically students from 12 to 16 years of age

The Australian Curriculum: Technologies describes two distinct but related subjects:

- **Design and Technologies**, in which students use design thinking and technologies to generate and produce designed solutions for authentic needs and opportunities, and
- **Digital Technologies**, in which students use computational thinking and information systems to define, design and implement digital solutions.

The learning area encourages students to apply their knowledge, practical skills and processes when using technologies and other resources to create innovative solutions, independently and collaboratively, that meet current and future needs. Figure 1, is a representation of the relationship between the two subject areas and the thinking skills that students are to develop in the Australian Curriculum: Technologies.

[1] Australian Government 2014, Industry Innovation and Competitiveness Agenda. Available at: https://www.dpmc.gov.au/sites/default/files/publications/industry_innovation_competitiveness_agenda.pdf

[2] Australian Curriculum and Reporting Authority (ACARA) 2015, Australian Curriculum: Technologies. Available at: http://www.australiancurriculum.edu.au/technologies/rationale

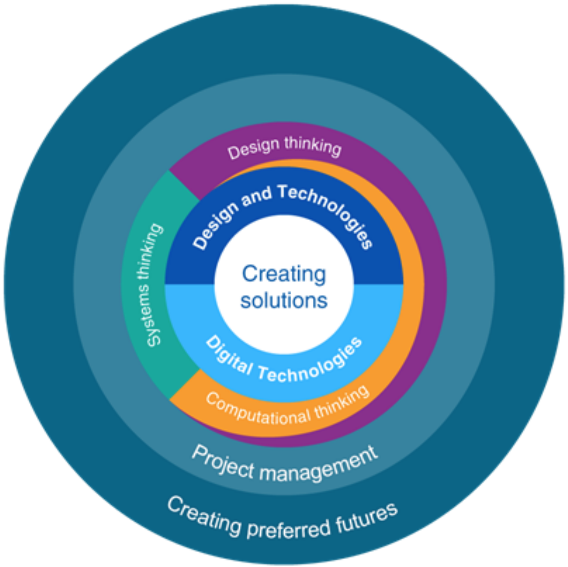


**Figure 1: ACARA 2015**[3]

The Australian Curriculum: Technologies aims to develop the knowledge, understanding and skills to ensure that, individually and collaboratively, students:

- investigate, design, plan, manage, create and evaluate solutions
- are creative, innovative and enterprising when using traditional, contemporary and emerging technologies, and understand how technologies have developed over time
- make informed and ethical decisions about the role, impact and use of technologies in the economy, environment and society for a sustainable future
- engage confidently with and responsibly select and manipulate appropriate technologies − materials, data, systems, components, tools and equipment − when designing and creating solutions
- critique, analyse and evaluate problems, needs or opportunities to identify and create solutions.

Relating to Computer Science, learning development from F-10 supports the understanding of the utility of technology, as well as the development of problem solving skills and an abstract understanding of Computer Science. The development of both digital literacy and computational thinking commences in the F-2 band. In F-2, learning is based around developing an understanding of the relationship between real and virtual worlds, the use of technology in communication and the importance of precise instructions (logical sequencing) and simple problem solving in the digital world.

In Years 3-6, students are guided to develop a wider understanding of the impact of technology, including family and community considerations, and are able to work on, and communicate about, more complex and elaborate problems and projects. From Year 3, students are introduced to visual programming languages, such as MIT's Scratch.

Across Years 7-10, students move beyond their initial community and are required to consider broader ethical and societal considerations. In this band, students should be able to solve sophisticated problems using technology, and develop an understanding of complex and abstract processes. From Year 7, students are introduced to general-purpose programming languages, such as Python and JavaScript.

[3] Australian Curriculum and Reporting Authority (ACARA) 2015, Australian Curriculum: Technologies. Available from: http://www.australiancurriculum.edu.au/technologies/content-structure.

# The Project

Although the development of the Australian Curriculum is the responsibility of all Australian governments, the implementation of the Australian Curriculum and the provision of teacher professional development is the key responsibility of state and territory government and non-government education authorities. This includes decisions that relate to the provision of resources that complement the teaching of the Australian Curriculum.

The Australian Curriculum: Technologies presents a significant opportunity for the next generation of Australian children to develop skill and mastery in the design and creation of technology, to position Australia as internationally competitive. However, the provision of high-quality learning and teaching materials are a fundamental component, to ensure successful implementation. The Australian Commonwealth has commissioned this project to identify the current landscape of Computer Science resources, suitable for the Australian Curriculum: Technologies, to inform the consideration of the curation and/or creation of resources under the Coding Across the Curriculum Program.

This report describes the methodology and results of a project undertaken by The University of Adelaide Computer Science Education Research (CSER) Group[4] for the Australian Department of Education. The project involved a review and analysis of currently available Computer Science resources for teachers and students, suitable for the Australian Curriculum: Technologies. The analysis explored two aspects: one part being the type of available resources and coverage of resources according to the content descriptors in the Australian Curriculum: Technologies. The second aspect involved analysis of the educational quality of the resources. The following section describes the project plan.

[4] CSER: http://blogs.adelaide.edu.au/cser/

# The Project Processes and Goals

A quality assessment framework was developed (Appendix 2), in partnership with the Department of Education's requirements and feedback. This framework was used to evaluate the quality of Computer Science resources – according to factors such as educational value, evidence of assessment, usability, and relevance to the Australian Curriculum: Technologies.

The CSER group applied semi-systematic review techniques[5] to locate currently available Computer Science resources online. The CSER group developed a survey tool, based on agreed indicators between CSER and the Department, to categorise the available Computer Science resources, and to evaluate the overall quality of each resource. The categorisation process involved mapping each resource to the content descriptors in the Australian Curriculum: Technologies, as well as the identification of the technologies and competence required to use the resource. Appendix 1 provides a more comprehensive overview of the item considered in the review process. The quality assessment framework (see Appendix 2), was used to explore characteristics, such as the openness of the resource, its appropriateness to the teaching context, and its ability to be immediately used within the teaching context.

This final report is produced for the Department of Education to describe the research findings. The report identifies resources currently available, areas of the Australian Curriculum: Technologies that are lacking available resources and the current level of educational quality of the resources. This report includes a series of recommendations to guide the future development of Computer Science resources to support the implementation of the Australian Curriculum: Technologies.

In summary, the project deliverables were to:

- Develop a Technologies quality assessment framework that can be used to evaluate the educational value and suitability of online and offline Computer Science resources for the Australian Curriculum: Technologies.
- Systematically curate and assess available online and offline Computer Science learning resources for teachers and students, mapping each to the areas and content descriptors of the Australian Curriculum: Technologies and recommending appropriate competency level for use.
- Produce a draft report and final high-quality report about the project's research findings and recommendations for the development of future Computer Science resources for the Australian Curriculum: Technologies.

A breakdown of the project plan, with major and minor milestones, can be found in Appendix 3.

[5] A. Booth, D. Papaioannou and A. Sutton, Systematic Approaches to a Successful Literature Review, SAGE Publications Inc, 2011.

# METHODOLOGY

This section describes the methodology and instruments applied to review current Computer Science resources that are relevant to the Australian Curriculum: Technologies. The resources were curated and mapped to the following:

- A set of resource attributes (Appendix 1)
- The Australian Curriculum: Technologies F-10 content descriptors (and senior years)[6]
- Indicators on a Quality Assessment Framework (Appendix 2).

## Search Criteria

The following inclusion and exclusion criteria were used for the review.

**Inclusion Criteria:**

- Primary and high school
- English language.
- Free and accessible.
- Developed for the purpose of learning and teaching.
- A complete resource package (not one-off).
- Computer Science resources.
- Relevant to the Australian Curriculum: Technologies.

**Exclusion Criteria:**

- Post-secondary or pre-school.
- Languages other than English.
- Pay-to-use resources.
- One-off resources (e.g. blogs).
- Resources specifically for other learning areas that partially link to the Australian Curriculum: Technologies.

## Process of Curating & Mapping Resources

An initial set of resources were located and recorded in a Google Sheet (See Appendix 4). The resources were identified based on an ongoing collection of Computer Science resources by the CSER Group, curated and pinned to the CSER Pinterest account[7]. Using the inclusion and exclusion criteria, relevant resources were selected and recorded in the Google Sheet. This identified an initial list of resources to be allocated for review.

A further, online semi-systematic review of Computer Science resources was undertaken to identify additional resources that had not yet been curated for analysis. The search was performed using the Google search engine. The following search terms and their derivatives were used.

**Key Search Terms**

- Computing
- Coding, code
- Computer Science
- Digital Technology
- Technology
- Programming

**Supporting Search Terms**

- Lessons
- Teacher
- Student
- Resources
- Unit Plan
- Professional Learning, Professional Development
- K-12, elementary, primary, high school, college.

[6] http://www.australiancurriculum.edu.au/technologies/rationale
[7] https://www.pinterest.com/edresearch/

## Instruments & Analysis

Once the Computer Science resources had been curated and recorded in the Google Sheet, the resources were to be mapped to particular attributes (Appendix 1) and a Quality Assessment Framework (Appendix 2).

The Quality Assessment Framework was developed by CSER, in consultation with the Department of Education. The Framework is based on Education Services Australia's [8](2012) standard for digital resources and Lloyd and Cochrane's (2006)[9] model of effective teacher professional development. We combine ESA's standards, which evaluate the educational value of resources, with Lloyd and Cochrane's elements of effective and sustainable professional development. In doing so, we devise a set of four principles that examine the Computer Science resources based on their *context*, *integrity*, *alignment* and *sustainability*. These principles determine the extent that resources address particular standards, such as whether resources are accessible, are relevant to particular audiences, contain appropriate disciplinary language, are current, include evaluation or assessment components, and are conducive of ongoing learning and development. The principles will be discussed in more detail in the Results section, along with the survey findings.

To map the resources to particular attributes, content descriptors and the Quality Framework, a survey tool[10] was developed using Google Forms. In doing so, details about the resources were being collected in a Google Sheet. These could then be exported and analysed for further analysis.

## Research Questions

The following research questions were investigated:

- What resources are available– according to year level, topics and content descriptors?
- What level of Computer Science knowledge are required to use the resources?
- What is the educational quality of resources according to the Quality Assessment Framework?

The goal of the investigation was to highlight what resources are required and recommendations for where efforts can be directed in terms of resource development.

The following section presents the research findings.

[8] Education Services Australia, 2012, Educational value standard for digital resources, version 1.01. Available: http://www.ndlrn.edu.au/verve/_resources/Educational_value_standard_for_digital_resources.pdf

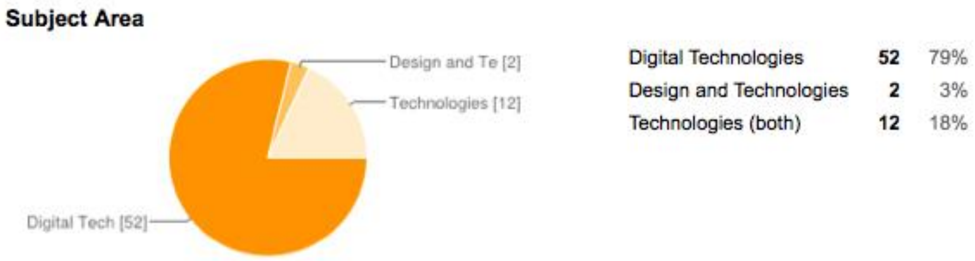
[9] Lloyd, M and Cochrane, J (2006), Celtic Knots: Interweaving the elements of effective teacher professional development in ICT, Australian Educational Computing, vol. 21, no. 2, pp. 16-19.

[10] Survey Tool sample: http://goo.gl/cfWvVt

# RESULTS

This section describes the results from the Computer Science resource review and mapping. A total of 79 websites were curated for review. Of the 79 sites, 65 Computer Science resources were identified as being relevant to the Australian Curriculum: Technologies and subsequently reviewed. A broad categorisation of the intended purpose of the resource was undertaken, including an assessment of expected background and required technologies, as well as an assessment of the quality of the resource according to our defined framework (Appendices 1 and 2).

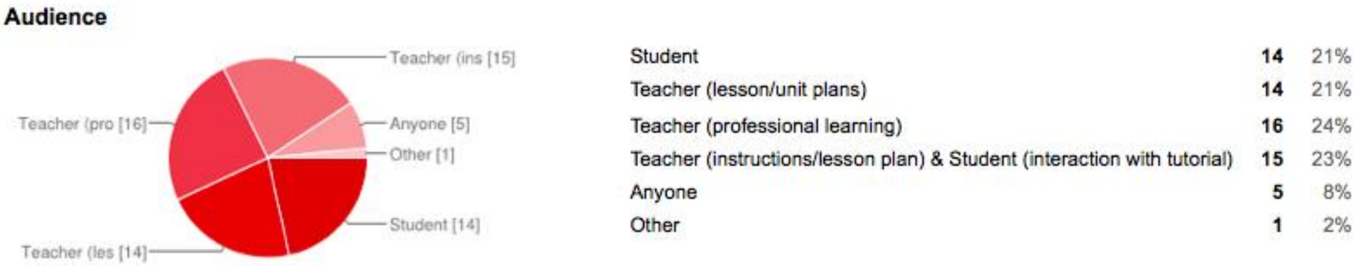
## Categorisation of Resources



| | | |
|---|---|---|
| Digital Technologies | 52 | 79% |
| Design and Technologies | 2 | 3% |
| Technologies (both) | 12 | 18% |

**Figure 2: Resource categorisation by broad subject area.**

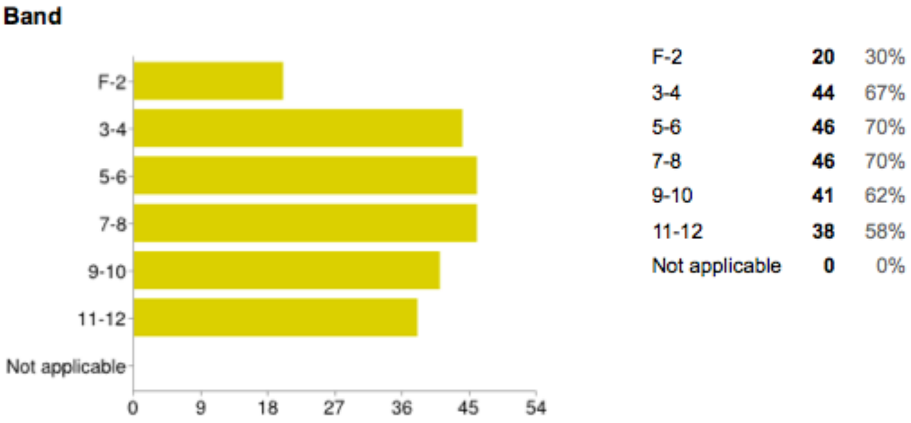
Overall, 79% of the resources reviewed focussed solely on the subject of Digital Technologies (see Figure 2). There is considerable potential for resources that address the intersection of Digital Technologies and Design and Technologies, facilitated by the growth in maker-space technologies that blend both understanding of materials, technology integration and computational thinking - 18% of the reviewed resources addressed both Design and Technologies and Digital Technologies. 2 (3%) of the resources, although identified through our searching methodology as addressing Computer Science and Computational Thinking, primarily addressed Design and Technologies.



| | | |
|---|---|---|
| Student | 14 | 21% |
| Teacher (lesson/unit plans) | 14 | 21% |
| Teacher (professional learning) | 16 | 24% |
| Teacher (instructions/lesson plan) & Student (interaction with tutorial) | 15 | 23% |
| Anyone | 5 | 8% |
| Other | 1 | 2% |

**Figure 3: Resource categorisation by primary audience.**

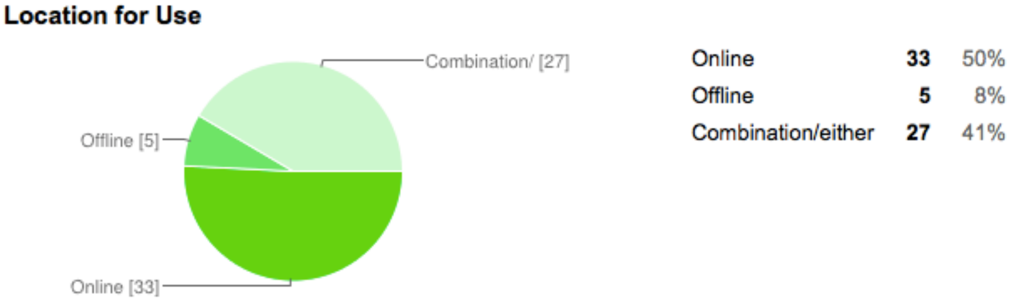
Our analysis of the resources identified a primary focus on teacher support, including professional learning (24%) but with a greater emphasis on the provision of lesson plans, either individual, unconnected lesson plans, or presented as a sequence through a structured unit (total, 44%). 23% of the resources reviewed provided support for both learners and teachers, with learner activities provided with accompanying lesson plan guidance.



| | | |
|---|---|---|
| F-2 | 20 | 30% |
| 3-4 | 44 | 67% |
| 5-6 | 46 | 70% |
| 7-8 | 46 | 70% |
| 9-10 | 41 | 62% |
| 11-12 | 38 | 58% |
| Not applicable | 0 | 0% |

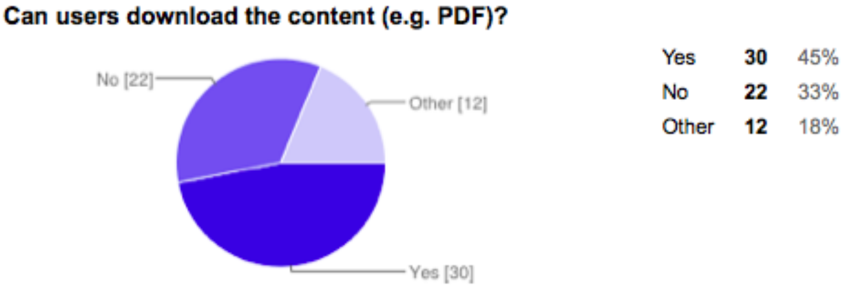
**Figure 4: Resource categorisation by intended grade band (as defined through their support for the Australian Curriculum).**

In our analysis, we categorised resources according to their intended grade band, through their support for content descriptors at specific year bands in the Australian Curriculum, and their degree of age appropriateness and contextualisation. The majority of resources provide support across multiple year bands, either through the provision of individual lesson plans or classroom resources at different year bands, or the provision of a sequence of lesson plans or classroom resources that move in a structured manner through the year bands. Figure 4 illustrates our categorisation, identifying the majority of resources target year bands 5-6, and 7-8, with 70% of resources providing support for year band 5-6, and also 70% of resources providing support for year band 7-8. Only 30% of resources specifically provide support for band F-2.



| | | |
|---|---|---|
| Online | 33 | 50% |
| Offline | 5 | 8% |
| Combination/either | 27 | 41% |

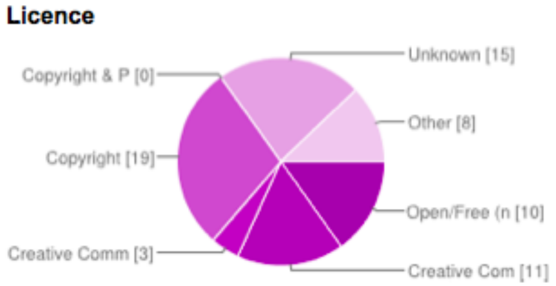
**Figure 5: Resource categorisation by location for use.**



| | | |
|---|---|---|
| Yes | 30 | 45% |
| No | 22 | 33% |
| Other | 12 | 18% |

**Figure 6: Is the resource downloadable for offline usage?**

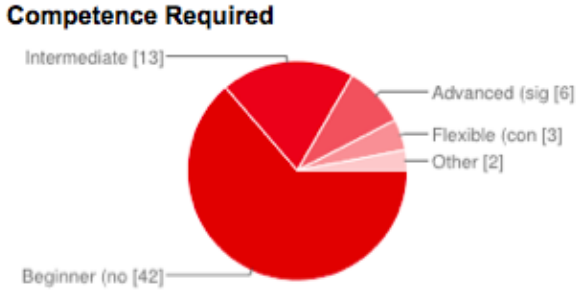
The availability of internet access is crucial to the usage of many of the reviewed resources, with 50% requiring online access for full usage, 41% available for both online and offline usage, while only 8% are designed as purely unplugged activities (Figure 5). However, further analysis shows that only 33% of the resources provide no form of useful resource that can be easily downloaded. 18% of the resources provided partial support for downloadable content, typically either through support for partial resource download, or download access restricted through additional registration requirements.



| | | |
|---|---|---|
| Open/Free (not listed as CC) | 10 | 15% |
| Creative Commons (Attribution ShareAlike) | 11 | 17% |
| Creative Commons (other than ShareAlike) | 3 | 5% |
| Copyright | 19 | 29% |
| Copyright & Purchase | 0 | 0% |
| Unknown | 15 | 23% |
| Other | 8 | 12% |

**Figure 7: Resource categorisation by resource license requirements.**

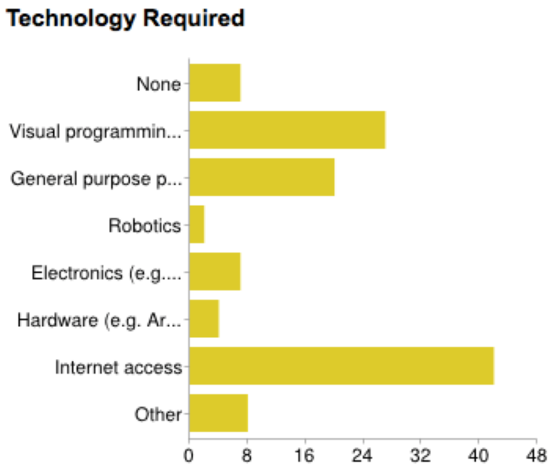
The majority of resources are made available under licensing agreements that support free and open usage, either through explicit licensing as such, Creative Commons licensing that support free usage and (somewhat restricted) modification, or through copyright licensing specifying open usage (see Figure 7). However, 23% of the resources reviewed did not clearly specify their usage requirements, while 12% provided a mixed approach with individual resources published under varying licensing agreements. These latter two approaches complicate the reuse process, with the potential for lack of clarity for teachers in the legality of their intended use of the resource, and the inability to classify a full resource as available for free and open usage.



| | | |
|---|---|---|
| Beginner (no knowledge) | 42 | 64% |
| Intermediate (some prior knowledge) | 13 | 20% |
| Advanced (significant knowledge) | 6 | 9% |
| Flexible (constructivist) | 3 | 5% |
| Other | 2 | 3% |

**Figure 8: Resource categorisation by degree of expected competence for usage.**

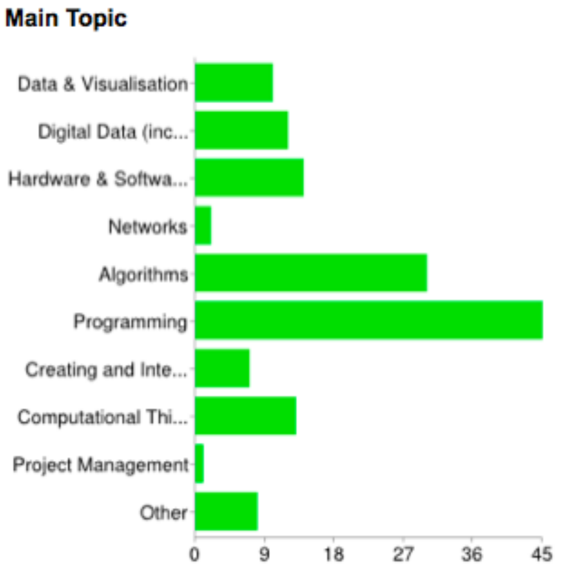
The vast majority of resources assume no background knowledge (64%), with 29% assuming either intermediate or advanced background knowledge (Figure 8). 5% supported a flexible range, with the provision of resources at each level. A closer analysis identifies a trend in the assumption of prior knowledge or experience. Approximately 30% of the resources reviewed focus primarily on F-6, while approximately 41% focus on 7-12, with the remaining approximate 29% supporting across all year levels, but typically with the majority of resources in the F-6 space. Within the F-6 year band, 65% of available resources assume beginner competence. Within the resources across all year bands, 88% of available resources assume beginner competence. However, within the 7-12 band, approximately 73% of the reviewed resources assume either intermediate or advanced competence.



| | | |
|---|---|---|
| None | **7** | 11% |
| Visual programming (e.g. Scratch) | **27** | 41% |
| General purpose programming | **20** | 30% |
| Robotics | **2** | 3% |
| Electronics (e.g. LittleBits, Makey Makey) | **7** | 11% |
| Hardware (e.g. Arduino) | **4** | 6% |
| Internet access | **42** | 64% |
| Other | **8** | 12% |

**Figure 9: Resource categorisation by technology required.**

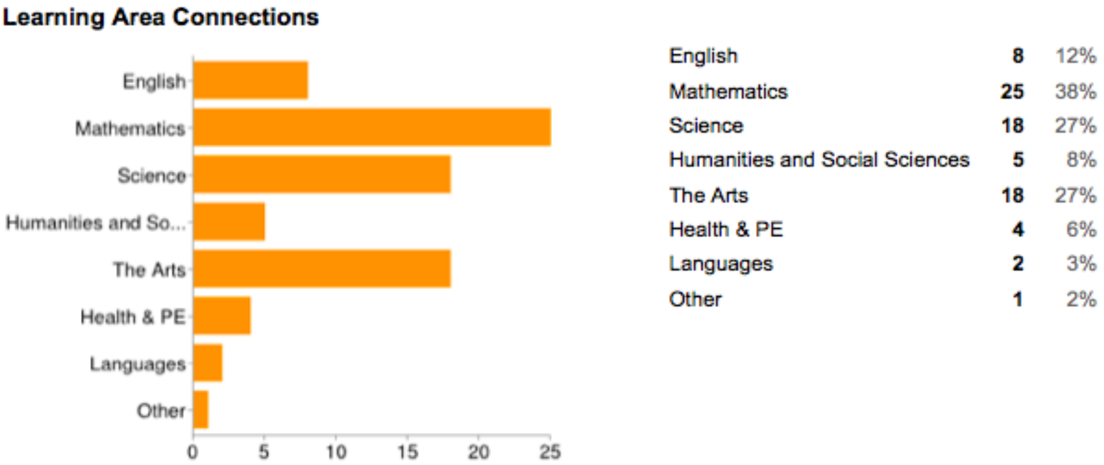
Figure 9 illustrates the different technologies assumed for use with the resources reviewed. 64% of the resources assume internet access for access to at least some of the support provided, while the remaining technology assumption is dominated by the expectation of the availability of either visual programming languages, or general purpose programming languages. A small number of resources support new forms of educational technology, including robotics and physical devices (e.g. Arduino, MaKey MaKey).



| | | |
|---|---|---|
| Data & Visualisation | **10** | 15% |
| Digital Data (incl. binary) | **12** | 18% |
| Hardware & Software (incl. interface design) | **14** | 21% |
| Networks | **2** | 3% |
| Algorithms | **30** | 45% |
| Programming | **45** | 68% |
| Creating and Interacting Online | **7** | 11% |
| Computational Thinking | **13** | 20% |
| Project Management | **1** | 2% |
| Other | **8** | 12% |

**Figure 10: Resource categorisation by main topic addressed.**

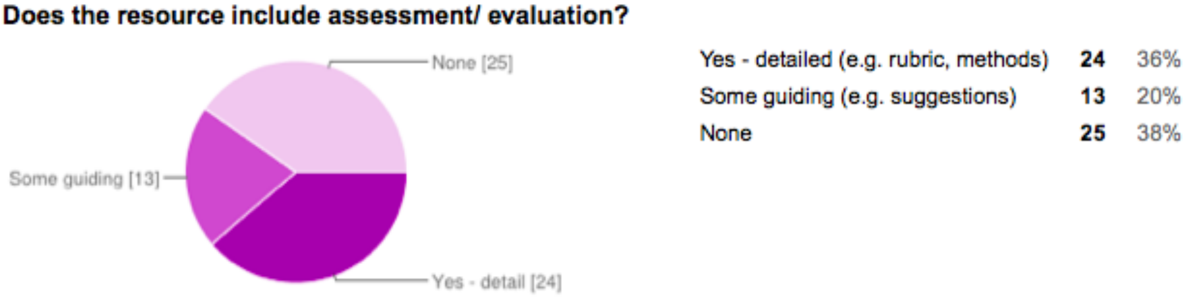
Within our categorisation, the resources reviewed have been mapped against the content descriptors of both Design and Technologies, and Digital Technologies at each grade band (formally, only 41% of the reviewed resources are themselves mapped to a curriculum standard, with only 9% mapped to the Australian Curriculum). In addition, we have undertaken a broad categorisation of the resources against the core topic areas identified within Digital Technologies, as listed in Figure 10. The dominant emphasis across the resources is supporting the development of programming skill (68% of resources), with the development of algorithm awareness a close second (45% of resources). Again, a trend can be seen here upon closer analysis based on year band. Resources targeting all year levels support a wide range of topic areas, while resources targeting primary or secondary only are more focussed on programming as their main topic, with in excess of 85% resources emphasising programming as their main topic.



| | | |
|---|---|---|
| English | **8** | 12% |
| Mathematics | **25** | 38% |
| Science | **18** | 27% |
| Humanities and Social Sciences | **5** | 8% |
| The Arts | **18** | 27% |
| Health & PE | **4** | 6% |
| Languages | **2** | 3% |
| Other | **1** | 2% |

**Figure 11: Integration with other learning areas.**

A minority of the resources reviewed made explicit connections with other learning areas (see Figure 11). Those that did make connection typically contextualised a lesson plan or a classroom activity within another learning area, but without integrating the activity to achieve content descriptors across both learning areas. The most common learning area for integration was Mathematics (38% of resources), while Science (27%) and the Arts (27%) were also commonly mentioned. Within the F-6 space, implicit integration with the Arts and English learning areas via contextualisation was the most common through the use of storyboarding or storytelling as a basis for visual programming, while Science and Mathematics were more popular in the 7+ level.



| | | |
|---|---|---|
| Yes - detailed (e.g. rubric, methods) | **24** | 36% |
| Some guiding (e.g. suggestions) | **13** | 20% |
| None | **25** | 38% |

**Figure 12: Resource categorisation by assessment or evaluation inclusion.**

Figure 12 illustrates the categorisation of the reviewed resources by their support for assessment or evaluation. Only 36% of resources provided detailed assessment methods, with some of these assessment methods restricted to registered users only. A further 20% of resources provided some guidance, in the form of suggestions for assessment or evaluation, while 38% provided no guidance. Where guidance was provided, the quality and inclusion typically varied across the lesson plan activities or classroom resources provided by the resource.

# Mapping to Curriculum Content Descriptors - Digital Technologies

The following figures illustrate the mapping of resources to content descriptors from the Australian Curriculum: Technologies (separated into Digital Technologies and Design and Technologies). For brevity, only part of the learning objective text is listed in each figure - the full detail is shown in the accompanying tables.

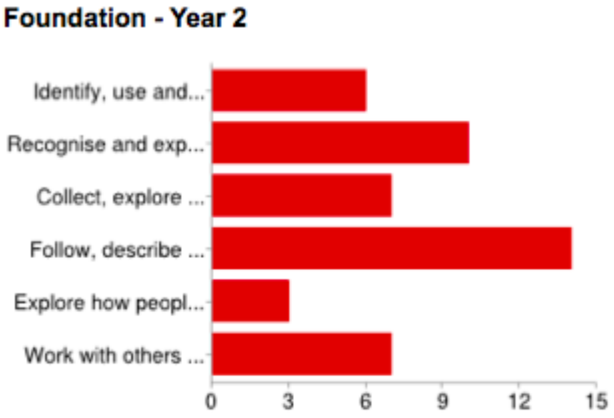


**Figure 13: Resources mapped to content descriptors - Digital Technologies (F-2).**

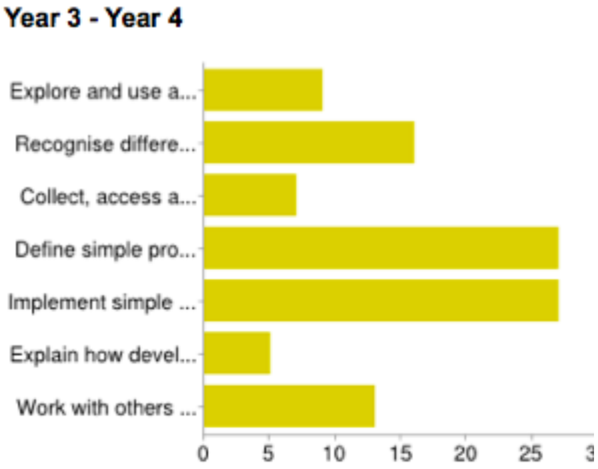


**Figure 14: Resources mapped to content descriptors - Digital Technologies (3-4).**

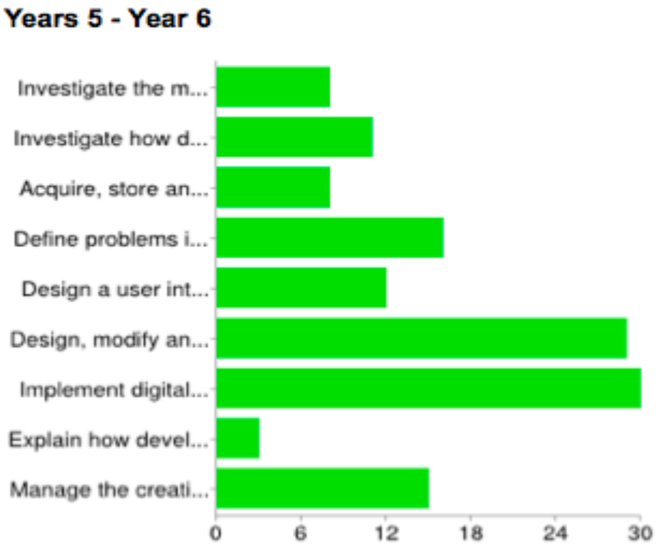


**Figure 15: Resources mapped to content descriptors - Digital Technologies (5-6).**

**Table 1: Mapping to Digital Technologies content descriptors (F-2).**

| | | |
|---|---|---|
| Identify, use and explore digital systems (hardware and software components) for a purpose (ACTDIK001) | 6 | 9% |
| Recognise and explore patterns in data and represent data as pictures, symbols and diagrams (ACTDIK002) | 10 | 15% |
| Collect, explore and sort data, and use digital systems to present the data creatively (ACTDIP003) | 7 | 11% |
| Follow, describe and represent a sequence of steps and decisions (algorithms) needed to solve simple problems (ACTDIP004) | 14 | 21% |
| Explore how people safely use common information systems to meet information, communication and recreation needs (ACTDIP005) | 3 | 5% |
| Work with others to create and organise ideas and information using information systems, and share these with known people in safe online environments (ACTDIP006) | 7 | 11% |

**Table 2: Mapping to Digital Technologies content descriptors (3-4).**

| | | |
|---|---|---|
| Explore and use a range of digital systems with peripheral devices for different purposes, and transmit different types of data (ACTDIK007) | 9 | 14% |
| Recognise different types of data and explore how the same data can be represented in different ways (ACTDIK008) | 16 | 24% |
| Collect, access and present different types of data using simple software to create information and solve problems (ACTDIP009) | 7 | 11% |
| Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them (ACTDIP010) | 27 | 41% |
| Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input (ACTDIP011) | 27 | 41% |
| Explain how developed solutions and existing information systems meet common personal, school or community needs, and envisage new ways of using them (ACTDIP012) | 5 | 8% |
| Work with others to plan the creation and communication of ideas and information safely, applying agreed ethical and social protocols (ACTDIP013) | 13 | 20% |

**Table 3: Mapping to Digital Technologies content descriptors (5-6).**

| | | |
|---|---|---|
| Investigate the main components of common digital systems, their basic functions and interactions, and how such digital systems may connect together to form networks to transmit data (ACTDIK014) | 8 | 12% |
| Investigate how digital systems use whole numbers as a basis for representing all types of data (ACTDIK015) | 11 | 17% |
| Acquire, store and validate different types of data and use a range of commonly available software to interpret and visualise data in context to create information (ACTDIP016) | 8 | 12% |
| Define problems in terms of data and functional requirements, and identify features similar to previously solved problems (ACTDIP017) | 16 | 24% |
| Design a user interface for a digital system, generating and considering alternative designs (ACTDIP018) | 12 | 18% |
| Design, modify and follow simple algorithms represented diagrammatically and in English involving sequences of steps, branching, and iteration (repetition) (ACTDIP019) | 29 | 44% |
| Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input (ACTDIP020) | 30 | 45% |
| Explain how developed solutions and existing information systems are sustainable and meet local community needs, considering opportunities and consequences for future applications (ACTDIP021) | 3 | 5% |
| Manage the creation and communication of ideas and information including online collaborative projects, applying agreed ethical, social and technical protocols (ACTDIP022) | 15 | 23% |

Starting with the F-6 context (Figures 13-15), analysis of this data highlights an emphasis on resources mapped to content descriptors associated with algorithm and programming skill development. While other content descriptors are also addressed, the focus of resources is very much on developing programming skill. A small number of resources map to content descriptors within the F-2 band - the most mapped objective, associated with algorithm development, is mapped by only 21% of resources. Within the F-4 context, resources map more broadly across the content descriptors, while in later year bands, resources are further focussed on programming and algorithms.

While data is identified as a topic area, and relevant content descriptors are able to be mapped to the reviewed resources, our analysis has identified a focus on specific exemplars of data-based activities. Specifically, in senior years (8+) the focus on data representation is on the understanding of binary representation, and activities to help explore binary representation. Within F-6, the focus extends from binary to the identification of patterns, and the visualisation of data through visual programming environments.

The support for the development of algorithmic skill, and explicit support for algorithm-based content descriptors is, at best, partial across the majority of resources as the development of algorithms, and experience in designing and thinking about algorithms is typically treated as an implicit skill developed while developing programming skill, rather than as a learning objective in its own right.

Areas that are not commonly mapped typically include objectives associated more so with the design process, rather than the development and application of programming skill itself. For example, while data is discussed in terms of visualisation, patterns and representation, it is rarely discussed in terms of its use in the problem solving process (ACTDIP003, ACTDIP009), validating different types of data (ACTDIP016), or in terms of explicitly analysing problems in terms of the data and functional requirements (ACTDIP017). Little attention is given to the analysis of the design of digital systems in terms of the relation of hardware and software. Finally, there is a lack of attention provided to the overall evaluation of system suitability and functionality.

At years 7 through to 10, the increased focus on programming skill becomes more apparent, with little emphasis on the remaining content descriptors of the Digital Technologies subject area (see Figures 16 and 17). There is a lack of resources mapping to the more advanced content descriptors concerning data (beyond binary representation), design skills and critical analysis and evaluation skills. Only one resource was identified as mapping to the project management area, despite this being a significant aspect throughout the Technologies subject area.

Significantly, resources that support the assessment of design skills, and evaluation and critical analysis skills are missing from our review of resources. While assessment guidance is provided for fundamental algorithm and programming skill development, it is typically defined in terms of correct functionality and understanding of a given algorithm, rather than in terms of an assessment of quality of the design and computational thinking process, starting from an analysis of data and functional requirements through to implementation.

While project management content descriptors are explicitly addressed by only one resource, we have identified opportunities for building project and teamwork skill across several resources. We have mapped these resources to the relevant content descriptors as demonstrating potential for the development of project management skills - this reflects the existence of quality lesson plans and classroom resources that introduce opportunities for project and teamwork skill development. While focusing primarily on informal project management practices, these may be supplemented by more formal resources to explicitly address project management related content descriptors.

While support for content descriptors associated with understanding privacy concerns and collaborating safely in online environments is confined to a comparatively small number of resources (7, or 11% as shown in Figure 10), the quality of these resources, their suitability for the teaching context, and inclusion of quality, immediately usable resources was high.

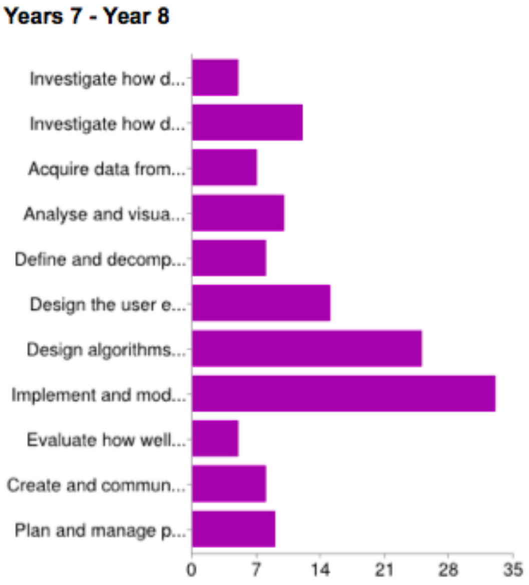


**Figure 16: Resources mapped to content descriptors - Digital Technologies (7-8).**

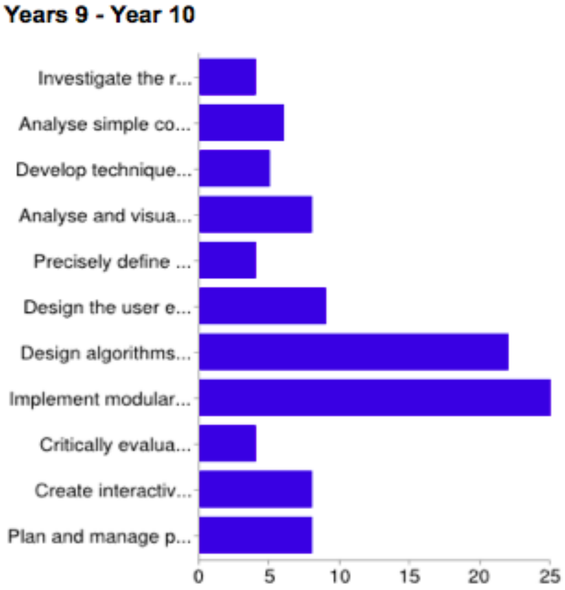


**Figure 17: Resources mapped to content descriptors - Digital Technologies (9-10).**

**Table 4: Mapping to Digital Technologies content descriptors (7-8).**

| | | |
|---|---|---|
| Investigate how data are transmitted and secured in wired, wireless and mobile networks, and how the specifications of hardware components impact on network activities (ACTDIK023) | 5 | 8% |
| Investigate how digital systems represent text, image and audio data in binary (ACTDIK024) | 12 | 18% |
| Acquire data from a range of sources and evaluate authenticity, accuracy and timeliness (ACTDIP025) | 7 | 11% |
| Analyse and visualise data using a range of software to create information, and use structured data to model objects or events (ACTDIP026) | 10 | 15% |
| Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints (ACTDIP027) | 8 | 12% |
| Design the user experience of a digital system, generating, evaluating and communicating alternative designs (ACTDIP028) | 15 | 23% |
| Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors (ACTDIP029) | 25 | 38% |
| Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language (ACTDIP030) | 33 | 50% |
| Evaluate how well developed solutions and existing information systems meet needs, are innovative and take account of future risks and sustainability (ACTDIP031) | 5 | 8% |
| Create and communicate interactive ideas and information collaboratively online, taking into account social contexts (ACTDIP032) | 8 | 12% |
| Plan and manage projects, including tasks, time and other resources required, considering safety and sustainability (ACTDIP033) | 9 | 14% |

**Table 5: Mapping to Digital Technologies content descriptors (9-10).**

| | | |
|---|---|---|
| Investigate the role of hardware and software in managing, controlling and securing the movement of and access to data in networked digital systems (ACTDIK034) | 4 | 6% |
| Analyse simple compression of data and how content data are separated from presentation (ACTDIK035) | 6 | 9% |
| Develop techniques for acquiring, storing and validating quantitative and qualitative data from a range of sources, considering privacy and security requirements (ACTDIP036) | 5 | 8% |
| Analyse and visualise data to create information and address complex problems, and model processes, entities and their relationships using structured data (ACTDIP037) | 8 | 12% |
| Precisely define and decompose real-world problems, taking into account functional and non-functional requirements and including interviewing stakeholders to identify needs (ACTDIP038) | 4 | 6% |
| Design the user experience of a digital system, evaluating alternative designs against criteria including functionality, accessibility, usability, and aesthetics (ACTDIP039) | 9 | 14% |
| Design algorithms represented diagrammatically and in structured English and validate algorithms and programs through tracing and test cases (ACTDIP040) | 22 | 33% |

| | | |
|---|---|---|
| Implement modular programs, applying selected algorithms and data structures including using an object-oriented programming language (ACTDIP041) | 25 | 38% |
| Critically evaluate how well developed solutions and existing information systems and policies take account of future risks and sustainability and provide opportunities for innovation and enterprise (ACTDIP042) | 4 | 6% |
| Create interactive solutions for sharing ideas and information online, taking into account social contexts and legal responsibilities (ACTDIP043) | 8 | 12% |
| Plan and manage projects using an iterative and collaborative approach, identifying risks and considering safety and sustainability (ACTDIP044) | 8 | 12% |

## Mapping to Curriculum Content Descriptors - Design and Technologies

A mapping of Computer Science resources was provided for content descriptors within Design and Technologies. Not all content descriptors are listed in our analysis below - only those for which a mapping was obtained, or for where an obvious connection with Computer Science resources was possible.

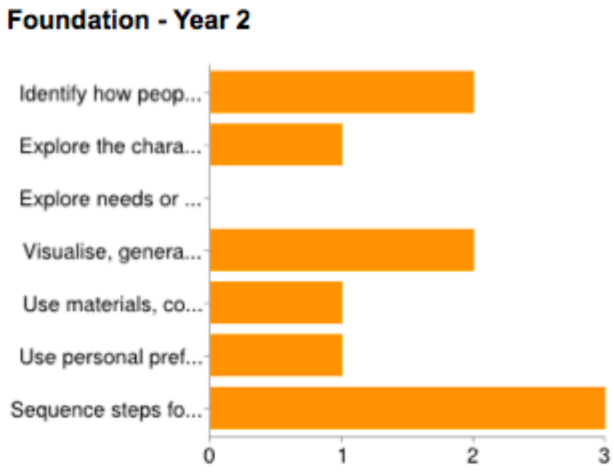


**Figure 18: Resources mapped to content descriptors - Design and Technologies (F-2).**

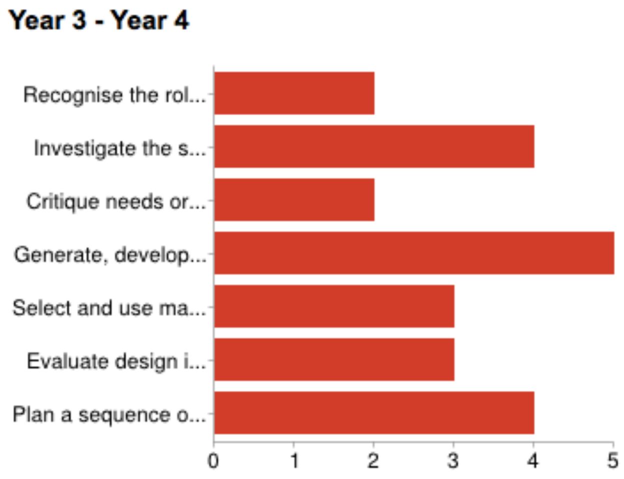


**Figure 19: Resources mapped to content descriptors - Design and Technologies (3-4).**

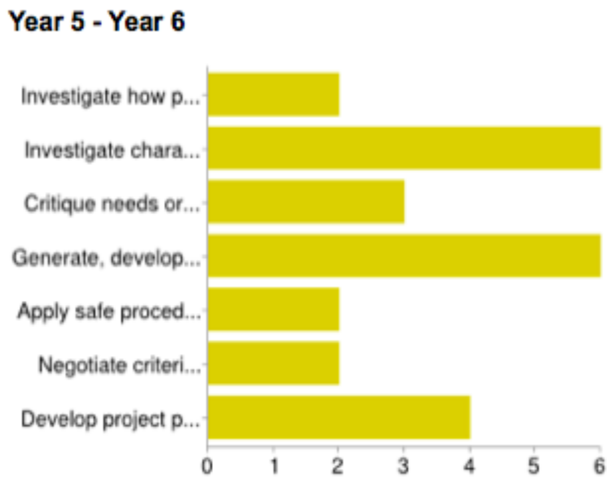


**Figure 20: Resources mapped to content descriptors - Design and Technologies (5-6).**

**Table 6: Mapping to Design and Technologies content descriptors (F-2).**

| | | |
|---|---|---|
| Identify how people design and produce familiar products, services and environments and consider sustainability to meet personal and local community needs (ACTDEK001) | 2 | 3% |
| Explore the characteristics and properties of materials and components that are used to produce designed solutions (ACTDEK004) | 1 | 2% |
| Explore needs or opportunities for designing, and the technologies needed to realise designed solutions (ACTDEP005) | 0 | 0% |
| Visualise, generate, develop and communicate design ideas through describing, drawing and modelling (ACTDEP006) | 2 | 3% |
| Use materials, components, tools, equipment and techniques to safely make designed solutions (ACTDEP007) | 1 | 2% |
| Use personal preferences to evaluate the success of design ideas, processes and solutions including their care for environment (ACTDEP008) | 1 | 2% |
| Sequence steps for making designed solutions and working collaboratively (ACTDEP009) | 3 | 5% |

**Table 7: Mapping to Design and Technologies content descriptors (3-4).**

| | | |
|---|---|---|
| Recognise the role of people in design and technologies occupations and explore factors, including sustainability that impact on the design of products, services and environments to meet community needs (ACTDEK010) | 2 | 3% |
| Investigate the suitability of materials, systems, components, tools and equipment for a range of purposes (ACTDEK013) | 4 | 6% |
| Critique needs or opportunities for designing and explore and test a variety of materials, components, tools and equipment and the techniques needed to produce designed solutions (ACTDEP014) | 2 | 3% |
| Generate, develop, and communicate design ideas and decisions using appropriate technical terms and graphical representation techniques (ACTDEP015) | 5 | 8% |
| Select and use materials, components, tools and equipment using safe work practices to make designed solutions (ACTDEP016) | 3 | 5% |
| Evaluate design ideas, processes and solutions based on criteria for success developed with guidance and including care for the environment (ACTDEP017) | 3 | 5% |
| Plan a sequence of production steps when making designed solutions individually and collaboratively (ACTDEP018) | 4 | 6% |

**Table 8: Mapping to Design and Technologies content descriptors (5-6).**

| | | |
|---|---|---|
| Investigate how people in design and technologies occupations address competing considerations, including sustainability in the design of products, services and environments for current and future use (ACTDEK019) | 2 | 3% |
| Investigate characteristics and properties of a range of materials, systems, components, tools and equipment and evaluate the impact of their use (ACTDEK023) | 6 | 9% |
| Critique needs or opportunities for designing, and investigate materials, components, tools, equipment and processes to achieve intended designed solutions (ACTDEP024) | 3 | 5% |
| Generate, develop, communicate and document design ideas and processes for audiences using appropriate technical terms and graphical representation techniques (ACTDEP025) | 6 | 9% |
| Apply safe procedures when using a variety of materials, components, tools, equipment and techniques to make designed solutions (ACTDEP026) | 2 | 3% |
| Negotiate criteria for success that include consideration of sustainability to evaluate design ideas, processes and solutions (ACTDEP027) | 2 | 3% |
| Develop project plans that include consideration of resources when making designed solutions individually and collaboratively (ACTDEP028) | 4 | 6% |

As mentioned, we consider there to be considerable potential for resources that address the intersection of Digital Technologies and Design and Technologies, through the growing importance and popularity of maker-space technologies, and increased availability of classroom technology that bridges this gap. Despite this, the dominant mapping to content descriptors in Design and Technologies from reviewed Computer Science resources came from resources that supported Digital Technologies and Design and Technologies separately, i.e. with separate resources for classroom activities and professional learning within the materials and material design spaces.

Within the year bands 7 onwards (Figures 21 & 22), more opportunities for mapping existing Computer Science resources to the Design and Technologies content descriptors were found. In F-6, lesson plans and classroom resources typically defined guided lessons, whereas in later years, it was common to identify projects requiring more independence and the requirement for design, and (informal) project planning and management. As per our mapping of Digital Technologies, we have mapped resources to content descriptors associated with project management due to their inclusion of opportunities for the development of informal project management skills.

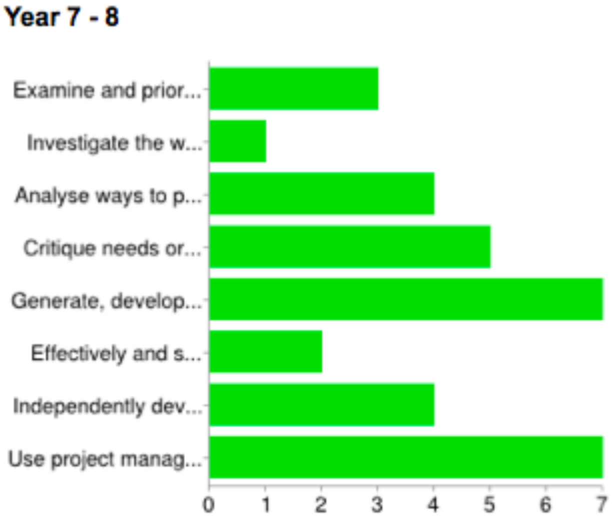


**Figure 21: Resources mapped to content descriptors - Design and Technologies (7-8).**

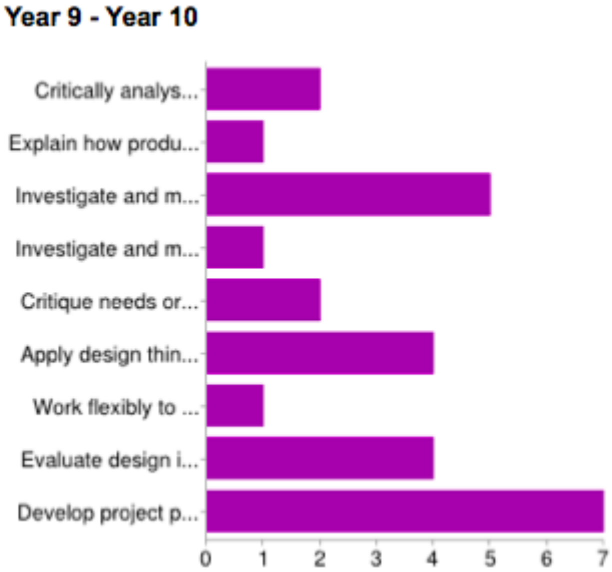


**Figure 22: Resources mapped to content descriptors - Design and Technologies (9-10).**

**Table 9: Mapping to Design and Technologies content descriptors (7-8).**

| | | |
|---|---|---|
| Examine and prioritise competing factors including social, ethical and sustainability considerations in the development of technologies and designed solutions to meet community needs for preferred futures (ACTDEK029) | 3 | 5% |
| Investigate the ways in which products, services and environments evolve locally, regionally and globally through the creativity, innovation and enterprise of individuals and groups (ACTDEK030) | 1 | 2% |
| Analyse ways to produce designed solutions through selecting and combining characteristics and properties of materials, systems, components, tools and equipment (ACTDEK034) | 4 | 6% |
| Critique needs or opportunities for designing and investigate, analyse and select from a range of materials, components, tools, equipment and processes to develop design ideas (ACTDEP035) | 5 | 8% |
| Generate, develop, test and communicate design ideas, plans and processes for various audiences using appropriate technical terms and technologies including graphical representation techniques (ACTDEP036) | 7 | 11% |
| Effectively and safely use a broad range of materials, components, tools, equipment and techniques to make designed solutions (ACTDEP037) | 2 | 3% |
| Independently develop criteria for success to assess design ideas, processes and solutions and their sustainability (ACTDEP038) | 4 | 6% |
| Use project management processes when working individually and collaboratively to coordinate production of designed solutions (ACTDEP039) | 7 | 11% |

**Table 10: Mapping to Design and Technologies content descriptors (9-10).**

| | | |
|---|---|---|
| Critically analyse factors, including social, ethical and sustainability considerations, that impact on designed solutions for global preferred futures and the complex design and production processes involved (ACTDEK040) | 2 | 3% |
| Explain how products, services and environments evolve with consideration of preferred futures and the impact of emerging technologies on design decisions (ACTDEK041) | 1 | 2% |
| Investigate and make judgments on how the characteristics and properties of materials, systems, components, tools and equipment can be combined to create designed solutions (ACTDEK046) | 5 | 8% |
| Investigate and make judgments, within a range of technologies specialisations, on how technologies can be combined to create designed solutions (ACTDEK047) | 1 | 2% |
| Critique needs or opportunities to develop design briefs and investigate and select an increasingly sophisticated range of materials, systems, components, tools and equipment to develop design ideas (ACTDEP048) | 2 | 3% |
| Apply design thinking, creativity, innovation and enterprise skills to develop, modify and communicate design ideas of increasing sophistication (ACTDEP049) | 4 | 6% |
| Work flexibly to safely test, select, justify and use appropriate technologies and processes to make designed solutions (ACTDEP050) | 1 | 2% |
| Evaluate design ideas, processes and solutions against comprehensive criteria for success recognising the need for sustainability (ACTDEP051) | 4 | 6% |
| Develop project plans using digital technologies to plan and manage projects individually and collaboratively taking into consideration time, cost, risk and production processes (ACTDEP052) | 7 | 11% |

## Quality Assessment

The Quality Assessment tool examined resources according to four major components (see Appendix 2). These included assessing the *context* of the resource, the *integrity*, *alignment* and *sustainability*. In order to make judgements at a deeper level, each component was further divided into four more factors to be assessed (discussed below). When reviewing the resources, the reviewers were to indicate if there were evidence of these standards in the resource by selecting either 'no', 'somewhat' or 'yes'. These strings were later converted to integers in order to analyse the average results for each component, with '0' being 'no', '1' being 'somewhat' and '2' representing 'yes'.

The average quality of the resources indicate that many had a reasonable level of integrity, however, they were reasonably low in terms of alignment. As the Australian Curriculum: Technologies is relatively new, there were some Australian resources mapped specifically to the Australian Curriculum but most (if present) were mapped to other international standards, such as the CSTA. A majority of resources had a solid context as they were relevant and able to be immediately used. In terms of sustainability, most resources were of reasonable quality but we identified opportunities where resources could be strengthened.

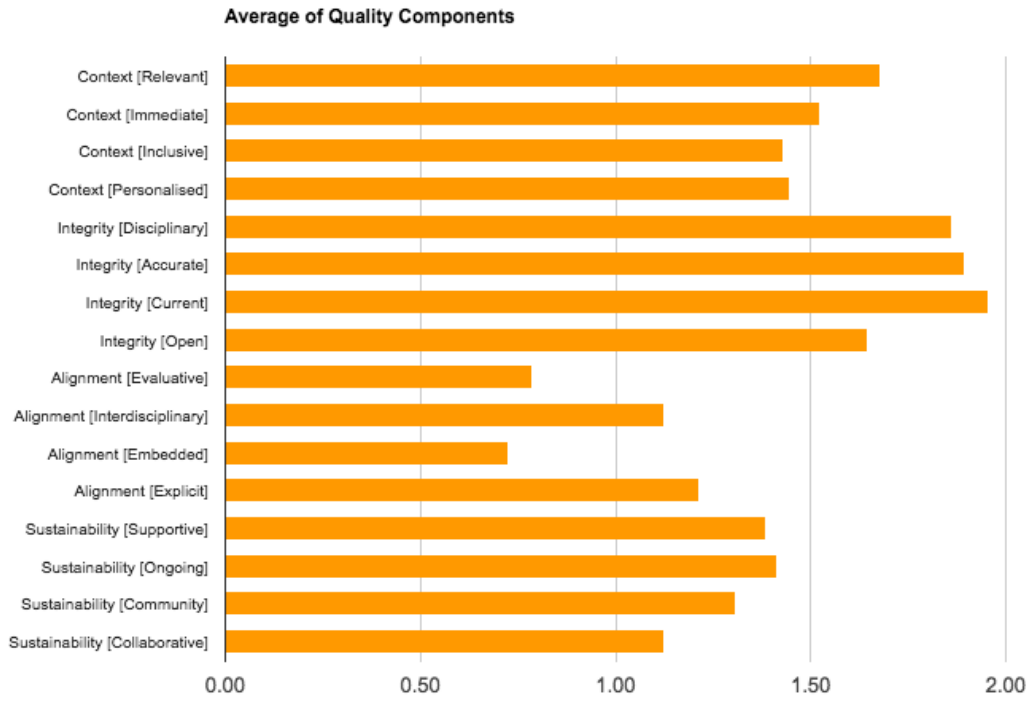
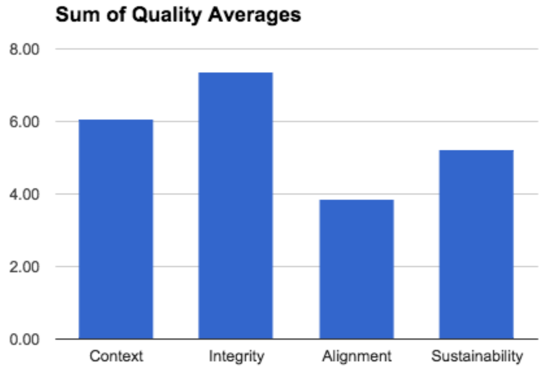


**Figure 23: Resources mapped to quality attributes - analysis.**



| | |
|---|---|
| Context | 6.08 |
| Integrity | 7.35 |
| Alignment | 3.85 |
| Sustainability | 5.23 |

**Figure 24: Sum of Quality Averages.**

The following section will describe the Quality Assessment Components and the findings about each component in more detail.

## Context

The context component involved examining whether the resource was an authentic learning experience and relevant to the context of the audience. Further, this component examined whether the resource were able to be immediately available to use and presented in a way that was inclusive of a variety of users.

**Table 11: Analysis of Quality Attribute Assessment - Context.**

| | Relevance | | Immediate | | Inclusive | | Personalised | |
|---|---|---|---|---|---|---|---|---|
| Yes | 47 | 72% | 39 | 60% | 32 | 49% | 33 | 51% |
| Somewhat | 15 | 23% | 21 | 32% | 29 | 45% | 28 | 43% |
| No | 3 | 5% | 5 | 8% | 4 | 6% | 4 | 6% |
| Total | 65 | 100% | 65 | 100% | 65 | 100% | 65 | 100% |

In terms of immediacy, we sought to find out whether resources were practical, able to be immediately accessed and implemented and whether they supported just-in-time learning. Over half of the resources (60%) were identified as being open and available for immediate use. Although, 40% of these resources were only 'somewhat available' or 'unavailable' for immediate use. Some online materials, such as lesson plans, were open and available to the public, whereas others required teachers to subscribe or request access. All of the resources reviewed for this report, that required a subscription or request to access, were free. In the case of technology-sites, it was noticed that, in some cases, a set of preliminary lesson plans were made available, with additional content requiring a price to access. Resources that were currently unavailable (n= 5) for immediate use included those that relied on release dates, such as the Bebras Computational Thinking Challenge, or online courses that had set timelines, such as those for teacher professional development on Coursera. Some online Computer Science professional learning courses, however, had been a one-off course and had ceased formal instruction but course materials were still open and available for use. This still allows teachers to use the materials for professional learning but it means that some of the course elements, such as assignments, instructor support and professional learning recognition are unavailable.

Evaluating the inclusivity of resources included factors such as whether materials could be downloaded, if they were able to be used by screen readers, if they were available in different languages and evaluating the extent that they were able to be tailored or adapted to suit particular contexts and needs (such as location, availability of resources and learners with disabilities). This question was quite broad as there are many aspects to consider in terms of usability and the extent that the resources addressed inclusivity varied. It was identified that a large number of resources (94%) were inclusive to some extent, however, there was still room for improvement to ensure resources are more inclusive for all users. As mentioned previously, 33% of resources were not able to be downloaded in some regard. Further, some resources were inconsistent in format, style, licensing agreement and whether or not they could be downloaded. Another potential issue across a number of resources is the accessibility of resources using screen readers. Some sites were not structured in a way that would complement an individual using a screen reader. Further, some of the online programming activities may be challenging to learners with particular disabilities or learning difficulties and further, many of the resources lacked any mention of alternative options.

A large majority of resources (72%) were relevant to the audience. Exemplary Computer Science activities were those that had a connection to real-world Computer Science applications, or the implementation of meaningful learning activities, particularly those that involved problem-solving, developing personalised projects or the creation of solutions to real problems (such as the creation of games or apps that address local issues). Resources that allowed learners to incorporate their own ideas, passions and develop a project that was their own stood out as positive and almost half of the resources were able to be personalised (51%) or to some extent (43%). Resources that were able to be personalised or tailored to context needs, allowed students and teachers to have greater choice. Resources for teachers that were able to be personalised involved tasks or materials that could be tailored to their particular teaching context. Some of these include integration with different learning areas, the ability to have plugged and/or unplugged activities or a blend of collaborative and guided learning.

In the case of teacher professional learning, resources that made connections between Computer Science theory, activities and practical implementation into the classroom were rated favourably. For example, online courses that involved teachers undertaking projects, developing classroom unit plans (or school action plans) provide meaningful and relevant opportunities for teachers to engage in Computer Science education. Although the inclusion of Computer Science pedagogy was not explicitly considered in the quality assessment framework, this was something that was considered when reviewing whether the resources for teachers. Overall, the resources for teachers did lack an element of pedagogy of Computer Science education. There was one resource though that this did quite well - a course for educators, where they wove pedagogy lessons between the Computer Science content and project activities.

Quality lesson plans were quite comprehensive - enough so that teachers could pick up the lesson and begin to implement it with little or no Computer Science knowledge or experience. Quality instruction-based tutorial activities and other resources were those that students could work through the activities from a beginner or appropriate level.

## Integrity

The integrity component involved judging whether the resource included the use of Computer Science disciplinary language and determining if it was an accurate, current and reputable resource. It included identifying whether the resource is open, accessible and free to use by anyone.

**Table 12: Analysis of Quality Attribute Assessment - Integrity.**

| | Disciplinary | | Accurate | | Current | | Open | |
|---|---|---|---|---|---|---|---|---|
| Yes | 56 | 86% | 58 | 89% | 62 | 95% | 43 | 66% |
| Somewhat | 9 | 14% | 7 | 11% | 3 | 5% | 21 | 32% |
| No | 0 | 0% | 0 | 0% | 0 | 0% | 1 | 2% |
| Total | 65 | 100% | 65 | 100% | 65 | 100% | 65 | 100% |

Many of the resources (95%) were up-to-date and contained accurate information (89%) from reputable sources. Further, many of the resources had used specific Computer Science terminology, where appropriate, and there were no issues with the language, accuracy or currency of the resources. Although many of the resources used Computer Science specific language, an issue did occur when the resource assumed teachers or students had the knowledge of the Computer Science concepts or programming languages (particularly general-purpose). When resources assume Computer Science knowledge it may be challenging for teachers to be able to explain and teach the concepts as well as guide students through the learning process. This issue was more common with high school resources, in which 29% assumed either intermediate or advanced background knowledge (Figure 8).

It was identified that the primary school resources involved a higher level of scaffolding. Primary school resources often introduced Computer Science concepts along with practical learning activities. However, the middle to high school resources in general-purpose programming had lost this structure and included more independent activities (such as general-purpose programming activities and 'makerspace' project work), which were removed from the explicit and guided teaching of Computer Science theory and concepts. In these cases, the resources assumed an adequate level of Computer Science knowledge by both teachers and students. Further, a number of the high school general-purpose programming resources also assumed a level of competency of general-purpose programming.

Where the resources did differ in integrity, was with respect to the openness. As reported earlier, although 29% of resources were listed under copyright, the majority of resources are made available under licensing agreements that support free and open usage (Figure 7). However, for 23% resources it was unclear what the licensing agreement was and 12% adopted mixed approaches, with individual resources being published under varying licensing agreements.

## Alignment

The alignment component involved identifying the extent that resources included an assessment or evaluation component and whether there were explicit links to other learning areas (interdisciplinary) and formal curriculum documents (embedded). It also involved judging the extent that the resource explicitly documents the intended content descriptors, outcomes and learning and teaching processes. As mentioned previously, alignment was identified as being one of the most under-addressed components, with quite a number of resources not having fully met these aspects or not having met them at all.

**Table 13: Analysis of Quality Attribute Assessment - Alignment.**

| | Evaluative | | Interdisciplinary | | Embedded | | Explicit | |
|---|---|---|---|---|---|---|---|---|
| Yes | 15 | 23% | 27 | 42% | 20 | 31% | 29 | 45% |
| Somewhat | 21 | 32% | 19 | 29% | 7 | 11% | 21 | 32% |
| No | 29 | 45% | 19 | 29% | 38 | 58% | 15 | 23% |
| Total | 65 | 100% | 65 | 100% | 65 | 100% | 65 | 100% |

Good resources were identified as having clear, explicit goals and objectives described in either teachers' learning and teaching materials, the course information or students' learning activities. 77% of resources were explicit or somewhat explicit in the content descriptors and outcomes but 23% of resources had no evidence of learning goals or explicit learning and teaching processes to follow. If the resources lack explicit content descriptors and teaching processes, this may be challenging for some educators who are unfamiliar with Computer Science concepts or the technology being used. Moreover, in terms of students' Computer Science resources, if the content descriptors are not explicitly described from the beginning, teachers may be uncertain whether the learning activities and online instruction are meeting students' learning needs.

Some 45% of resources were missing an assessment or evaluation component. Of the 55% of resources that did have, or somewhat had, an assessment or evaluation component, these varied greatly in terms of quality and detail. A large majority of these featured partial evaluation or assessment guidance through suggested discussion questions, rather than explicit assessment tools and rubrics. It was noted that some of the resources for students that involved one-to-one online interactive tutorial instruction, provided students with instant feedback about the correctness of their code. For example, sites such as Made With Code and Blockly Games will inform students when they have made a mistake and invite them to try again with hints. However, if using these during lessons, it is anticipated that these would also be paired with more rigorous formative and summative assessment tools developed by teachers. It was identified that some sites, such as Khan Academy and Code.org allow teachers to enrol their class and track students' progress through learning activities with progress charts.

In regards to the limited inclusion of learning and teaching assessment, it was noted in the review that many of the online resources had a primary goal to inspire, motivate and provide children and youth with opportunities to engage in Computer Science activities as many are derived from outreach programs and initiatives, such as Made With Code or intended for use as an after school club, such as the Code Clubs or CS First materials. Therefore, it is important to bear in mind the intended purpose for why these materials are developed in order to determine what supplementary materials might assist their implementation into classrooms for learning and teaching purposes. It appears that many of these high-quality resources would complement classroom learning and teaching and can improve on 'alignment' quality if supplementary assessment tools and support documents are created that align with the Australian Curriculum: Technologies.

A majority (58%) of resources have not been mapped to any formal curricula or documentation but 27 (42%) were mapped in some way. Upon closer examination of the findings, we noted that the following number of resources had been mapped to particular documentation:

- The Australian Curriculum (n= 7, with 6 being to the Technologies subject area)

- The Computer Science Teachers Association (CSTA) Standards (n= 7)
- England's Curriculum (n= 6)
- The Common Core Standards (n= 5)
- The National Council of Teachers of Mathematics (NCTM) (n= 1)
- The New Zealand Curriculum (n= 1)

As mentioned previously, a minority of the resources reviewed made explicit connections with other learning areas (see Figure 11) but overall, 71% of resources had potential to be integrated more explicitly as they were based on activities that are familiar or relevant to other learning areas. In our earlier findings we identified that most cross-curricula connections were made between Computer Science and English, Mathematics, Science and the Arts.

## Sustainability

Lastly, the sustainability component examined the support and community mechanisms that the resource included as well as the extent that the resource allowed for ongoing learning and collaborative learning opportunities.

**Table 14: Analysis of Quality Attribute Assessment - Sustainability.**

| | Supportive | | Ongoing | | Community | | Collaborative | |
|---|---|---|---|---|---|---|---|---|
| Yes | 37 | 57% | 35 | 54% | 34 | 52% | 25 | 38% |
| Somewhat | 16 | 25% | 22 | 34% | 17 | 26% | 23 | 35% |
| No | 12 | 18% | 8 | 12% | 14 | 22% | 17 | 26% |
| Total | 65 | 100% | 65 | 100% | 65 | 100% | 65 | 100% |

The review identified that many online sites are making use of community integration as a means to expand resources, ideas and projects, as well as provide learners with support. Such examples include MIT's Scratch Project Community and the LittleBits Community, where anyone can upload their project ideas for others to see and comment. These provide powerful opportunities for learners and educators to seek and receive help and advice, as well as suggest and receive ideas for adapting the resources for particular contexts and learner needs.

A large portion of resources had some element of support (82%). These included supportive elements such as 'help' features embedded in websites, additional supporting documents, glossary of terms, instructor support, discussion forums, social media integration and community (peer) support. We also note that there are other avenues of support, outside the scope of this review, that may provide sustainable and valuable support, if required, during the use of these learning resources. Some examples are the CSIRO Scientist in Schools Program, or the support relationships cultivated through the involvement in face-to-face or virtual clubs and Australian competitions. Further, there may be local or in-school sources of support that can be drawn on, such as experienced teachers, parents or students. It would be worthwhile exploring these sources of support as schools go ahead with implementing the new learning area.

Some 73% of resources involved collaborative learning opportunities. These included more implicit activities, where students could create projects together or share their projects online to receive feedback. Communities that appeared to be more successful were those environments, which not only provided support for learners and teachers, but also allowed for learners and teachers to have ownership over creation and community sharing.

# DISCUSSION

One intended purpose of this review is the identification of gaps within the available support for teachers in integrating the Australian Curriculum: Technologies, and the subsequent identification of development opportunities for new resources that are needed to best support teachers.

There exists a significant range of resources designed to support Computer Science development that may be useful in supporting teachers with the new Australian Curriculum: Technologies, including curated lesson plans and classroom resources, online courses, tools and teaching technologies and outreach resources. Our review identified that the existing resources represented high-quality resources, that were, in the majority, designed for immediate use within the education context. However, we also identified several gaps in terms of the support provided for the content descriptors of the Technologies curriculum area, and - accordingly - opportunities for the development of new and complementary resources.

Our review of these resources has identified a strong focus on the development of programming skills, with an emphasis on developing content knowledge and sharing lesson and activity ideas. However, when mapping to the Australian Curriculum: Technologies, it has become clear that there are significant areas of the curriculum that are either not addressed, or addressed only implicitly. Further, our analysis using our defined quality assessment framework has identified concerns over the completeness of the available resources, with a deficiency in explicit tools and advice on assessment and guidance as to appropriate pedagogy.

We have identified several gaps within the resources available for Computer Science and the Australian Curriculum: Technologies, that we feel represent opportunities for targeted development and investment. These are discussed here, and presented in summary in our next section. These recommendations may provide guidance to the development of new or complementary resources for both students and teachers that can support the effective integration and implementation of the Australian Curriculum: Technologies.

It was identified that there are a number of high-quality Computer Science resources available that would be suitable for the Australian Curriculum: Technologies. However, some of these high-quality resources may be lacking an element or two that supports their immediate implementation and ease of use in an Australian F-12 education context. For these high-quality Computer Science resources, it may be sufficient to invest in the development of supplementary resources to support their use, such as the development of assessment frameworks and materials, guiding pedagogy, lesson plan exemplars, or links to the Australian Curriculum: Technologies content descriptors.

**Recommendation 1:** That supplementary materials be developed to support the integration of existing high-quality Computer Science resources into the classroom, to support the Australian Curriculum: Technologies.

As seen in the results of the identified resources, the majority of resources across all year bands are associated with the algorithm- and programming-related content descriptors. Our analysis has indicated a stronger emphasis on programming, with implicit mention of algorithm development and computational thinking in most cases. This is particularly true within the 7-8 and 9-10 year bands. What we have observed within the available resources is a tendency to promote the learning of programming through the learning of a specific programming language. There is an opportunity to further address content descriptors associated with the development of design and algorithm development skills (i.e. the analysis of data and functional requirements, and the design and communication of algorithms) in new resources.

Unplugged activities that promote instruction identification and sequencing assist in addressing related content descriptors within the F-6 year bands, with a particular focus on F-2. However, there is still room for the development of complementary resources within F-6 to support the use of visual programming languages, where in existing resources these content descriptors are frequently addressed only implicitly.

This highlights an overall concern with the lack of resources that explicitly identify computational thinking as a holistic process, including explicit and assessed elements matching to data analysis, functional requirement analysis, algorithm design, implementation via programming and then evaluation of fit-for-purpose. This is of particular concern within the year bands 5-6, 7-8 and 9-10.

**Recommendation 2:** That resources associated with the explicit development and assessment of the computational thinking process as a holistic process across F-10 be supported, including:

- data and functional requirements analysis,
- algorithm design and evaluation,
- programming as an element of this process, and
- evaluation and critical analysis.

Assessment of programming as an overall evaluation component is an area that requires further support, particularly in the year bands 7-8 and 9-10. While programming is the focus for many of the resources made available, guidelines for how to assess programming, including both assessment of functional correctness and assessment of non-functional properties such as design quality, clarity and conciseness, are rarely found. Within the F-6 context, more guidance is made available, particularly in the very early years (F-2), however this decreases once visual programming becomes more dominant.

**Recommendation 3:** That resources designed to explore suitable assessment guidelines for programming be developed appropriate for each year band.

Further, building on recommendations 2 and 3, there is a need to bring this information together and to make it more explicitly clear to teachers the developmental process from Foundation to Year 10. We recommend that a developmental framework would be useful to guide teachers in developing learning activities by mapping the introduction of computing and programming concepts and skills over the duration of their F-12 schooling.

**Recommendation 4:** That a developmental framework(s) be created to map the developmental and overall sequential process of learning and teaching of programming knowledge and skills across the F-12 Australian Curriculum: Digital Technologies.

There is a need to address the lack of explicit resources associated with content descriptors associated with project management. While opportunities for project and teamwork are prevalent throughout the reviewed resources, there is little support for teachers in terms of specific guidance of the relevant degree of project management, types of appropriate project management per year band, assessment and appropriate pedagogy.

**Recommendation 5:** That resources associated with the explicit development and assessment of content descriptors associated with project management (F-10) be supported.

The overall number of resources that supported content descriptors within the Foundation-2 year band was comparatively low. Only one resource that exclusively addressed this year band was reviewed (ScratchJr), built around the use of a visual programming language designed specifically for this age group. The use of visual programming within this year band is not explicitly documented in the content descriptors of this year band, but may be useful as a complementary technology to the more unplugged style of activities promoted by the content descriptors and exemplars in this year band.

**Recommendation 6**: That resources designed for the explicit development and assessment of content descriptors associated with the F-2 year band be supported.

In our analysis, we identified a broad lack of reference to appropriate pedagogy. While lesson plans and classroom resources were often clear regarding their broad learning objective, and in some cases, relevant assessment techniques, it was rare to include discussion on relevant pedagogy for this discipline, for example the use of instruction versus play-based techniques, opportunities for collaborative learning or pair programming.

**Recommendation 7:** That resources associated with the explicit definition and examples of appropriate pedagogy across F-10, both general and specific to the Computer Science discipline, be supported. Further, that research into effective pedagogies for Computer Science be undertaken to support resource development.

As seen in our analysis, many of the resources available for year bands 7-8 and 9-10 require intermediate or advanced competency or background knowledge. In particular, the resources at these bands tend to focus more so on development of content knowledge rather than in the provision of detailed lesson plans and activities. Furthermore, the transition from visual programming to general purpose programming language within a classroom context is not addressed by any of the resources we reviewed. We have identified a clear need here for resources that address both of these concerns.

A case study for how this might be developed can be seen within the family of MIT Scratch resources, consisting of a specific programming framework targeting F-2, and a more complicated, but related, programming framework for 3-4, and 5-6. Accompanying both of these frameworks are a community for sharing basic ideas and examples and a teacher specific community supporting the development and sharing of more detailed lesson plans, structured curriculum, structured professional learning and assessment guidance. This integration of resources provides a well structured example of how resources can be directed towards specific year bands, while still maintaining a clear development and learning pathway that encapsulates all resources.

**Recommendation 8:** That resources associated with the transition from visual programming languages to general purpose programming languages be developed to support teachers with beginner competency.

The importance of community cannot be underrated. In our review, we found that resources that included an open community, i.e. a community where contributions and support from all contributors were welcomed, was a critical aspect in developing a sustainable and vibrant resource. Further, many high-quality resources had ‘help’ features or supporting resources to assist educators or students when requiring help, particularly when encountering issues or challenges in programming (visual or general-purpose). Where tools or resources are not explicitly available, educators could be made aware of communities, such as those on GitHub, where they, or students, may be able to seek or find technical support.

**Recommendation 9:** That developed resources be paired with an open community, providing opportunities for peer- and expert-assistance, and facilitating the development, evaluation and sharing of new resource components.

The growing prevalence and availability of physical technology designed for the classroom (e.g. Arduino, MaKey MaKey, robotics) opens opportunities for highly interactive learning activities within both Digital Technologies and Design and Technologies and opportunities for integration between these two subject areas. However, there is a lack of resources that address both of these learning areas in an integrated fashion, as well as resources that provide holistic support for how these technologies might be used within the classroom. While resources exist to demonstrate exemplars of use, additional resources are required to provide both tutorial-based support in their use, in addition to lesson plans and classroom resources illustrating their development, assessment and appropriate pedagogy.

**Recommendation 10:** That resources associated the use of innovative physical technologies (e.g. hardware and electronics), supporting Digital Technologies, Design and Technologies, and their integration be supported.

The reviewed resources were made available using a wide variety of licensing arrangements, with a significant percentage (total: 35%) that were either unclear regarding their usage arrangements, or provided mixed arrangements. In some cases, the provision of mixed licensing meant that individual lesson plans or classroom resources (i.e. individual videos, or demonstrations) were made available under different licensing arrangements, and identifiable only at the individual lesson plan or classroom resource level. This makes the process of reuse and adoption unnecessarily complicated for teachers, and may represent concerns for their inclusion in classroom teaching, regulated curations of resources and/or the adaptation of resources to suit new contexts and needs.

**Recommendation 11:** That a consistent and appropriate licensing agreement style be adopted for all resources to be developed, and that this licensing agreement style be promoted as that to be used for external, third parties when developing resources to be used within the Education sector.

As learner populations are increasingly diverse and the Australian Curriculum: Technologies is designed to be taught across all F-10 year levels, within a variety of schools and contexts, there is a need to ensure resources accommodate for a variety of learning needs, learning preferences and contexts. Resources that involved projects where learners had control over what they created opened up opportunities for personalised learning and potential for teachers to personalise learning activities to suit their contexts. Furthermore, it is suggested that student voice should play a role in the development of learning activities so that materials and projects are being developed that support student interests and needs.

This study was limited in terms of expert assessment of resources for learners with special needs, however, it is believed that a number of resources may encounter issues when assistive technologies, such as screen readers are required. Further, many resources required the Internet and not all provided alternative or ‘unplugged’ options, potentially posing challenges for educators with unreliable or no Internet connection at all.

**Recommendation 12:** That resources accommodate the diverse population of F-12 learners and contexts (e.g. learners with disabilities, contexts with limited Internet access, female engagement), taking advantage of personalised and inclusive learning design, alternative implementation approaches or the provision of supporting resources, where appropriate.

The above recommendation may also adopt the approach of recommendation 1, where supplementary resources are developed to accompany existing high-quality learning resources, to accommodate for varying context or learner needs. Further, building on this recommendation we propose that an introductory document, much like that produced by the Computing at Schools Group for England’s new Computing subject, be developed to introduce teachers to the subject area, provide an introductory overview and support teachers and leadership on how to “get started”.

**Recommendation 13:** That an introductory guiding document be developed to introduce teachers to the F-10 Digital Technologies subject, the key terminology, the context, some guiding examples, and how to get started.

Our recommendations present a combination of recommendations that could be initiated as individual projects, and recommendations that represent holistic guidelines across any initiated project.

In terms of structuring these recommendations, we would suggest that the following framework might support a suitable structure to guide initial implementation projects. We see an immediate need to help clarify scope and language within the curriculum. Recommendation 13 identifies the need to develop a scoping and guiding document for the curriculum that explains, with examples from the educational context, new concepts and terminology, extending beyond a glossary.

We also see benefit in pursuing two of the recommendations at an early stage: Recommendations 1 and 2. With the prevalence of high quality resources addressing the development of technical skill and content development within this learning area, we see that an immediate benefit can be provided through the development of a general framework guiding teachers through the problem solving process as a whole, encapsulating and placing the individual phases of data and requirements analysis, algorithm design and evaluation, programming and implementation, and evaluation and critical analysis. We see that this can be strengthened through the inclusion of additional case studies, where existing resources are placed within this framework and the process of scaffolding resources within the holistic problem solving process is modelled.

# RECOMMENDATIONS

In summary, the recommendations of this review report are:

**Recommendation 1:** That supplementary materials be developed to support the integration of existing high-quality Computer Science resources into the classroom, to support the Australian Curriculum: Technologies.

**Recommendation 2:** That resources associated with the explicit development and assessment of the computational thinking process as a holistic process across F-10 be supported, including:

- data and functional requirements analysis
- algorithm design and evaluation
- programming as an element of this process
- evaluation and critical analysis

**Recommendation 3:** That resources designed to explore suitable assessment guidelines for programming be developed appropriate for each year band.

**Recommendation 4:** That a framework(s) be created to map the developmental process of learning and teaching of programming knowledge and skills across the F-12 Australian Curriculum: Digital Technologies.

**Recommendation 5:** That resources associated with the explicit development and assessment of content descriptors associated with project management (F-10) be supported.

**Recommendation 6**: That resources designed for the explicit development and assessment of content descriptors associated with the F-2 year band be supported.

**Recommendation 7:** That resources associated with the explicit definition and examples of appropriate pedagogy across F-10, both general and specific to the Computer Science discipline, be supported. Further, that research into effective pedagogies for Computer Science be undertaken to support resource development.

**Recommendation 8:** That resources associated with the transition from visual programming languages to general purpose programming languages be developed to support teachers with beginner competency.

**Recommendation 9:** That developed resources be paired with an open community, providing opportunities for peer- and expert-assistance, and facilitating the development, evaluation and sharing of new resource components.

**Recommendation 10:** That resources associated the use of innovative physical technologies (e.g. hardware and electronics), supporting Digital Technologies, Design and Technologies, and their integration be supported.

**Recommendation 11:** That a consistent and appropriate licensing agreement style be adopted for all resources to be developed, and that this licensing agreement style be promoted as that to be used for external, third parties when developing resources to be used within the Education sector.

**Recommendation 12:** That resources accommodate the diverse population of F-12 learners and contexts (e.g. learners with disabilities, contexts with limited Internet access, female engagement), taking advantage of personalised and inclusive learning design, alternative implementation approaches or supporting resources, where appropriate.

**Recommendation 13:** That an introductory guiding document be developed to introduce teachers to the Digital Technologies subject, the key terminology, the context and how to get started, within an Australian context.

# CONCLUSIONS

The Australian Curriculum: Technologies presents a significant opportunity for the next generation of Australian children to develop skill and mastery in the design and creation of technology, both digital and otherwise. This provides the necessary awareness and background knowledge for expanding innovation, and ensuring that technology, in all of its forms, can be utilised to its full extent within our future society.

This report, prepared as part of the Australian Government's Coding Across the Curriculum initiative, aims to support curriculum implementation and resource development of the Australian Curriculum: Technologies. This report involved the review and analysis of currently available Computer Science resources for teachers and students, suitable for the Australian Curriculum: Technologies. The analysis explored two aspects: one part being the type of available resources and coverage of resources according to the Technologies content descriptors and the second involved analysis of the educational quality of the resources

As part of this review, the University of Adelaide Computer Science Education Research (CSER) Group, in consultation with the Department of Education, prepared a quality assessment framework, designed to assess and categorise available Computer Science resources according to their appropriateness for the educational context, their integrity with the Computer Science discipline, their alignment with curriculum goals, and their sustainability as a resource. Further, we reviewed, categorised and mapped each resource according to the content descriptors of the Australian Curriculum: Technologies.

In our review, we found a predominance of quality resources that were appropriately contextualised and maintained high integrity and overall accuracy in their support for the Computer Science discipline. We found that certain categories of content descriptors, such as the development of algorithms and programming content knowledge, and support for safe collaboration in online environments, were well supported. For many of these existing resources, supplementary materials may be developed to align the resources with the Australian Curriculum and support classroom implementation in Australian F-12 contexts.

However, we identified gaps in the support available for Computer Science resources within the Australian Curriculum: Technologies, in which we recommend the development of additional resources that complement those that are currently available. The set of recommendations in this report may provide guidance and points to consider for those developing resources for the Australian Curriculum: Technologies.

These recommendations, if supported, will provide valuable and needed support for teachers in their work towards addressing the challenge of this new Australian Curriculum: Technologies.

# PROJECT APPENDICES

# Appendix 1 - Resource Mapping Attributes

The following is a list of attributes that were used as a basis to develop the Survey Tool and to 'map' the curated Computer Science resources to the Australian Curriculum: Technologies.

| Australian Curriculum Links | |
|---|---|
| Subject area | Digital Technologies<br>Design and Technologies |
| Audience | Teacher<br>Student |
| Year Level Band | F-2<br>Years 3-4<br>Years 5-6<br>Years 7-8<br>Years 9-10<br>Years 11-12 |
| Strands | Knowledge & Understanding<br>Processes & Production Skills |
| Content descriptors | Relevant content descriptors by curriculum codes, e.g. ACTDIP005*. |
| Location for use | Offline/ Online |
| License | Free/ Creative Commons License/ Purchase |
| Type of resource (short text description) | e.g. professional development, lesson plan, video, website, booklet, ebook, game, etc. |
| Does it require any (at cost) technology? | List: (e.g. Little Bits, Makey Makey…) |
| Topic (short text description) | e.g. binary counting, algorithms, game creation, data visualisation. |
| Location of resource development | Identify the location of where the resource was developed (e.g. Australia, UK, New Zealand). |
| Assessment | Assessment guide present? (none, some, detailed). |
| Competence required (student/teacher) | Beginner, intermediate, advanced (or flexible (constructivist approach)). |
| Resource Quality Assessment Framework | See Appendix 2 |
| * As Year 11 & 12 curriculum is yet to be reviewed, we will map as being suitable for these years and all attributes except the Strands and content descriptors. | |

# Appendix 2 - Resource Quality Assessment Framework

Quality assessment framework including a blend of both ESA's[11] and Lloyd and Cochrane's[12] frameworks.

| Principle | Element | Description |
|---|---|---|
| **Context** | Relevant | Is the resource based on authentic learning or is the resource explicitly relevant to the intended audience? Is the resource purposeful, meaningful and local/real? |
| | Immediate | Does the resource support 'just-in-time' learning? (e.g. Can it be used immediately? Is it practical?) |
| | Inclusive | Is the resource adaptable and/or inclusive of age, race, culture, disability, gender or location? (e.g. adaptable, other languages). Is it able to be used with screen readers or able to be downloaded? |
| | Personalised | Does the resource support prior learning, future learning and personal growth? (e.g. is it flexible? Does it allow for varying levels of complexity or different learning styles? Does it provide scaffolding? Concept explanations?) |
| **Integrity** | Disciplinary | Does the resource include and use CS disciplinary language/symbols and content? |
| | Accurate | Is the information accurate and/or from a reputable source? |
| | Current | Is the content current and up-to-date? |
| | Open | Is the resource open and accessible? Can it be freely used? |
| **Alignment** | Evaluative | Does the resource include assessment guidelines or an element of evaluation? |
| | Interdisciplinary | Does the resource draw connections with other learning areas and/or skill development? |
| | Embedded | Is the resource mapped to curriculum or Teacher Professional Standards? |
| | Explicit | Is the resource explicit about the purpose, process and intended outcomes required of learning/ development? Are these clearly defined and labelled? |
| **Sustainability** | Supportive | Does the resource feature 'help' resources and guided learning development? |
| | Ongoing | Does the resource support ongoing learning and development? (e.g. self-directed learning opportunities, suggestions for extended learning, additional resources, open-ended tasks, reflection) |
| | Community | Does the resource have an element of community - or link to one? (e.g. where learners can seek help or access a wider network and resources?) |
| | Collaborative | Does the resource allow for collaborative opportunities and sharing? |

[11] Education Services Australia. (2012). Educational value standard for digital resources, version 1.01, Available at: http://www.ndlrn.edu.au/verve/_resources/Educational_value_standard_for_digital_resources.pdf

[12] Lloyd, M. & Cochrane, J. (2006). Celtic knots: Interweaving the elements of effective teacher professional development in ICT. *Australian Educational Computing* 21(2), pp. 16-19.

# Appendix 3 – Project Timeline

| Date | Milestone | Description |
|---|---|---|
| 28th January 2015 | **Project commences.** | |
| 2nd February 2015 | **Department briefing session.** | Phone conference meeting with Jane Aitken, A/Prof Katrina Falkner, Dr Rebecca Vivian. |
| | | Prepare project plan for submission to the Department, in accordance with the tender application and the phone discussion with Jane Aitken. |
| 9th February 2015 | **Project plan submitted for consultation.** | Project plan submitted to the Department for consultation and approval.<br>A list of attributes that will be used for resource mapping will be submitted, with the project plan, to the Department for review approval. Once approved, resource mapping will commence. |
| | **Commence development of the resource quality assessment framework.** | A/Prof Katrina Falkner and Dr Rebecca Vivian will immediately commence the development of the quality assessment framework drawing on ESA's 'Education value standard for digital resources' and Lloyd & Cochrane's (2006) paper on effective online professional development. |
| 13th February 2015 | **Resource quality assessment framework submitted for Department approval.** | The resource quality assessment framework will be submitted by email to Jane Aitken at the Department for review and approval.<br>Any recommendations or required changes requested by the Department will be made to the resource quality assessment framework. |
| | **Commence review of** Computer Science **resources.** | A/Prof Katrina Falkner and Dr Rebecca Vivian will commence the review and mapping of available online and offline Computer Science resources, as outlined in the project overview. Each resource will be evaluated using the resource quality assessment framework. |
| | **Analysis of available resources.** | A/Prof Katrina Falkner and Dr Rebecca Vivian will analyse the available resources that have been curated and identify gaps for future resource development. |
| | **Report writing.** | A/Prof Katrina Falkner and Dr Rebecca Vivian will maintain the report as an ongoing process of development throughout the project. The research findings from the resource mapping will be added to the report after the analysis of available resources. Prior to the 6th of March the report will be finalised and ready for submission to the Department. |
| 6th March 2015 | **Draft report submitted for consultation.** | Draft report to be emailed to the Department for consultation and feedback.<br>Draft report amended to meet the Department feedback/recommendations. The final polished version of the report will be prepared for submission. |

| | | |
|---|---|---|
| 13th March 2015 | **Final report submitted.** | The final polished report will be submitted by email to the Department. |
| 30th March 2015 | **Project concludes.** | |

# Appendix 4 – Curated Resources

*DT= Digital Technologies, D&T = Design and Technologies

| Resource | Resource URL/ Location | Subject | Competence Required | Technology Required? | Main Topic(s) | Audience | Band |
|---|---|---|---|---|---|---|---|
| **Websites** | | | | | | | |
| CS Field Guide | http://www.csfieldguide.org.nz | DT | Flexible | None | Data & Visualisation, Digital Data (incl. binary), Algorithms, Programming | Student (Teacher materials available on request) | 5-6, 7-8, 9-10 |
| CS Unplugged | http://csunplugged.org | DT | Beginner | General purpose programming, Internet | Digital Data (incl. binary), Hardware & Software (incl. interface design), Algorithms, Compression and Encryption | Teacher (lesson/unit plans) | 3-4, 5-6, 7-8 |
| Google Computational Thinking | https://www.google.com/edu/resources/programs/exploring-computational-thinking/ | DT | Varied. 9-12 level assume competence and ability in Python. | Visual Programming, General purpose programming, Robotics, Electronics , Internet | Data & Visualisation, Algorithms, Programming, Computational Thinking | Teacher (lesson/unit plans) | F-2, 3-4, 5-6, 7-8, 9-10, 11-12 |
| BBC | http://www.bbc.co.uk/schools/0/computing/ | DT | Flexible | None, Internet only required for the interactive tests | Data & Visualisation, Digital Data (incl. binary), Algorithms, Programming, Computational Thinking | Teacher (instructions/lesson plan) & Student (interaction with tutorial) | F-2, 3-4, 5-6, 7-8, 9-10 |
| Common Sense Media | https://www.commonsensemedia.org/educators/lesson | DT | Flexible | Internet | Creating and Interacting Online | Teacher (instructions/lesson plan) & Student (interaction with tutorial) | F-2, 3-4, 5-6, 7-8, 9-10, 11-12 |
| Cyber Smart | Cybersmart.gov.au | DT | Beginner | Visual Programming | Creating and Interacting Online | Teacher (instructions/lesson plan) & Student (interaction with tutorial) | F-2, 3-4, 5-6, 7-8, 9-10, 11-12 |
| Think You Know? | http://www.thinkuknow.co.uk | DT | Beginner | Internet | Creating and Interacting Online | Teacher (instructions/lesson plan) & Student (interaction | F-2, 3-4, 5-6, 7-8, 9-10, 11-12 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | with tutorial) | |
| Web Maker | https://webmaker.org/en-US/make-your-own | DT | Intermediate | Visual Programming, Internet | Programming, General resources built using web-based technology as examples | Teacher (lesson/unit plans) | 7-8, 9-10, 11-12 |
| ABC Splash | http://splash.abc.net.au/home#!/home | Both | Beginner | Visual Programming, Internet | Programming | | F-2, 3-4, 5-6, 7-8, 9-10 |
| Science Buddies | http://www.sciencebuddies.org/ | Both | Intermediate | Visual Programming | Programming | Student | 7-8, 9-10, 11-12 |
| Try Computing | http://www.trycomputing.org/inspire | DT | Intermediate | Internet | Digital Data (incl. binary), Networks, Algorithms | Teacher (lesson/unit plans) | 5-6, 7-8, 9-10, 11-12 |
| Thinking Yourself | http://games.thinkingmyself.com/ | DT | Beginner | General purpose programming, Internet | Computational Thinking | Student | F-2, 3-4 |
| EduTopia - Coding | http://www.edutopia.org/topic/coding-classroom | DT | Intermediate | None, Visual Programming, Internet | Algorithms, Programming | Teacher (professional learning) | F-2, 3-4, 5-6 |
| Australian Curriculum Lessons - Digital Technologies | http://www.australiancurriculumlessons.com.au/category/technology-lesson-plans/digital-technologies-lesson-plans/ | DT | Beginner | None | Data & Visualisation, Algorithms, Programming | Teacher (lesson/unit plans) | F-2, 3-4, 5-6 |
| Teaching Ideas - Computing | http://www.teachingideas.co.uk/ict/contents.htm | DT | Intermediate | General purpose programming, Internet | Hardware & Software (incl. interface design), Creating and Interacting Online, ICT Literacy | Teacher (lesson/unit plans) | F-2, 3-4, 5-6 |
| Greenfoot | http://www.greenfoot.org/doc | DT | Intermediate | Visual Programming, Electronics | Algorithms, Programming | Teacher (professional learning) | 9-10, 11-12 |
| Educade - Makey Makey | http://educade.org/lesson_plans/use-makey-makey-to-design-a-videogame-controller | Both | Intermediate | General purpose programming, Internet | Hardware & Software (incl. interface design) | Teacher (lesson/unit plans) | 3-4, 5-6 |
| ScratchED | http://scratched.gse.harvard.edu | DT | Beginner | General purpose programming, | Data & Visualisation, Algorithms, | Teacher (instructions/lesson plan) | 3-4, 5-6, 7-8 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | Internet | Programming | & Student (interaction with tutorial) | |
| Coding Bat | http://codingbat.com | DT | Intermediate | General purpose programming | Programming | Student | 9-10, 11-12 |
| **Outreach/ Organisations** | | | | | | | |
| Code Club | http://projects.codeclubworld.org/en-GB/ | DT | Beginner | Presentation Software (& any other showcase tech) | Programming | Student | 3-4, 5-6, 7-8, 9-10, 11-12 |
| CS FIRST | http://www.cs-first.com/materials | DT | Beginner | Internet | Programming | Teacher (instructions/lesson plan) & Student (interaction with tutorial) | 3-4, 5-6, 7-8 |
| Mobilizing CS | http://www.mobilizingcs.org/for-teachers | DT | Beginner | Visual Programming, Internet | Data & Visualisation | Teacher (lesson/unit plans) | 7-8, 9-10, 11-12 |
| CSTA - Computational Thinking | http://csta.acm.org/Curriculum/sub/CompThinking.html | DT | Beginner | Visual Programming, General purpose programming, Electronics , Hardware (e.g. Arduino), Internet | Computational Thinking | Teacher (professional learning) | F-2, 3-4, 5-6, 7-8, 9-10, 11-12 |
| CSIRO CREST | http://www.csiro.au/Portals/Education/Teachers/Classroom-activities/CREST/Overview/How-it-works.aspx | D&T | Beginner | Internet | Depends on projects - science & technology | Teacher (instructions/lesson plan) & Student (interaction with tutorial) | F-2, 3-4, 5-6, 7-8, 9-10, 11-12 |
| Bebras | http://www.bebras.edu.au/ | DT | Beginner | Internet | Computational Thinking | Student | 3-4, 5-6, 7-8, 9-10, 11-12 |
| NCWiT - IT Outreach | http://www.ncwit.org/resources/outreach-box-discovering-it | Both | Beginner | Visual Programming, General purpose programming, Robotics, Electronics , Hardware (e.g. Arduino) | Overview of CS & Careers | Outreach Materials | F-2, 3-4, 5-6, 7-8, 9-10, 11-12 |
| CS4FN | http://www.cs4fn.org/teachers/ | DT | Intermediate | Visual Programming, | Programming, Computational | Teacher (lesson/un | 11-12 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | General purpose programming, Internet | Thinking | it plans) | |
| CSTA - Curriculum Resources | http://csta.acm.org/Curriculum/sub/CurrResources.html | DT | Beginner | | Digital Data (incl. binary), Networks, Algorithms, Programming, Project Management | Teacher (lesson/unit plans) | 3-4, 5-6, 7-8, 9-10, 11-12 |
| DLTV | https://dltv.vic.edu.au/search?query=Digital%20Technologies | DT | Beginner | Internet | Algorithms, Computational Thinking, Digital Technologies learning area overview. | Teacher (professional learning) | F-2, 3-4, 5-6, 7-8, 9-10, 11-12 |
| NCWiT - Agent Cubes | http://www.ncwit.org/resources/agentcubes-box-introduce-computing-through-game-design | Both | Beginner | Visual Programming | Programming | Teacher (lesson/unit plans) | 3-4, 5-6 |
| Made With Code | https://www.madewithcode.com/resources | Both | Beginner | Internet | Hardware & Software (incl. interface design), Programming | Student | 3-4, 5-6, 7-8, 9-10, 11-12 |
| Exploring CS (Google) | https://sites.google.com/site/tasteofcs/<br>http://www.exploringcs.org/about/mission<br>(First site leads to second) | DT | Intermediate | General purpose programming, Internet | Digital Data (incl. binary), Programming, Computational Thinking | Teacher (lesson/unit plans) | 11-12 |
| **Courses** | | | | | | | |
| Coursera | https://www.coursera.org/course/ictinprimary | Both | Beginner | Depends on projects | Hardware & Software (incl. interface design), Programming | Teacher (professional learning) | F-2, 3-4, 5-6, 7-8 |
| Tickering Fundamentals (Coursera) | https://www.coursera.org/course/tinkering | Both | Beginner | Visual Programming | Hardware & Software (incl. interface design) | Teacher (professional learning) | F-2, 3-4, 5-6, 7-8, 9-10, 11-12 |
| Code Yourself! An Introduction to Programming (Coursera) | https://www.coursera.org/course/codeyourself | DT | Beginner | Visual Programming | Algorithms, Programming | Anyone | 3-4, 5-6, 7-8, 9-10, 11-12 |
| Introduction to Programming with | https://uni-cs4hs-scratch.appspot.com/preview | DT | Beginner | Internet | Programming | Teacher (professional learning) | 3-4, 5-6, 7-8, 9-10, 11-12 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Scratch in Education MOOC | | | | | | | |
| Google Making Sense of Data MOOC | https://datasense.withgoogle.com/course | DT | Beginner | General purpose programming, Hardware (e.g. Arduino), Internet | Data & Visualisation | Anyone | 3-4, 5-6, 7-8, 9-10, 11-12 |
| Teach Mobile CS | https://ram8647.appspot.com/teach_mobileCSP/preview | DT | Beginner | Internet | Hardware & Software (incl. interface design), Programming | Teacher (professional learning) | 7-8, 9-10, 11-12 |
| Google Internet 101 Course | https://educourses.withgoogle.com/101 | DT | Beginner | Internet | Creating and Interacting Online, Internet | Student | 3-4, 5-6, 7-8, 9-10, 11-12 |
| IV MOOC | http://ivmooc.cns.iu.edu/ | DT | Intermediate | Internet | Data & Visualisation | Anyone | 9-10, 11-12 |
| Teach IT MOOC | https://css-cs4hs.appspot.com | D&T | Beginner | Internet | Internet | Teacher (professional learning) | 3-4, 5-6, 7-8, 9-10, 11-12 |
| CSER F-6 Digital Technologies MOOC | https://csdigitaltech.appspot.com/cser_foundations6/preview | DT | Beginner | Visual Programming, Internet | Data & Visualisation, Digital Data (incl. binary), Hardware & Software (incl. interface design), Algorithms, Creating and Interacting Online, Computational Thinking | Teacher (professional learning) | F-2, 3-4, 5-6 |
| Creative Computing MOOC | https://creative-computing.appspot.com | DT | Beginner | None, Visual Programming, Internet | Algorithms, Programming, Computational Thinking | Teacher (professional learning) | 3-4, 5-6, 7-8, 9-10, 11-12 |
| **Technology Sites** | | | | | | | |
| Scratch MIT - Makey Makey (Games) | http://scratch.mit.edu/studios/223260/ | DT | Intermediate | Visual Programming, Internet | Hardware & Software (incl. interface design), Algorithms, Programming | Teacher (professional learning) | 3-4, 5-6 |
| Scratch - Makey Makey (Music) | http://scratch.mit.edu/studios/223257/ | DT | Intermediate | Visual Programming, | Algorithms, Programming | Teacher (professional learning) | 3-4, 5-6 |
| Wonder Workshop - Dash and Dot | https://www.makewonder.com/play/ideas/ | DT | Intermediate | Visual Programming, General purpose programming, | Hardware & Software (incl. interface design), Algorithms, | Teacher (professional learning) | 3-4, 5-6 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | Electronics, Dash and Dot | Programming | | |
| Little Bits | http://littlebits.cc/browse-lessons | Both | Intermediate | Electronics , Internet | Hardware & Software (incl. interface design), Algorithms, Programming | Teacher (lesson/unit plans) | 5-6, 7-8 |
| Kids Ruby | http://kidsruby.com/ | DT | Intermediate | General purpose programming, Internet | Algorithms, Programming | Teacher (instructions/lesson plan) & Student (interaction with tutorial) | 7-8 |
| Ruby for Kids | http://ruby4kids.com/ruby4kids | DT | Intermediate | Visual Programming, Tablet | Algorithms, Programming | Anyone | 7-8 |
| ScratchJR | http://www.scratchjr.org/ | DT | Beginner | Visual Programming | Algorithms, Programming | Teacher (instructions/lesson plan) & Student (interaction with tutorial) | F-2 |
| Scratch MIT | http://scratch.mit.edu/ | DT | Beginner | Visual Programming, Internet | Data & Visualisation, Algorithms, Programming | Student | 3-4, 5-6 |
| Hackety Hack | http://www.hackety.com/lessons | DT | Beginner | Visual Programming, Hardware (e.g. Arduino) | Programming | Anyone | 7-8, 9-10, 11-12 |
| MIT App Inventor | http://appinventor.mit.edu/ | Both | Beginner | General purpose programming, Internet | Hardware & Software (incl. interface design), Algorithms, Programming | Teacher (instructions/lesson plan) & Student (interaction with tutorial) | 7-8, 9-10, 11-12 |
| Kodu | http://www.kodugamelab.com/resources/ | DT | Beginner | Visual Programming, Internet | Programming | Teacher (instructions/lesson plan) & Student (interaction with tutorial) | 3-4, 5-6, 7-8 |
| Blockly Games | https://blockly-games.appspot.com | DT | Beginner | Internet, Web based | Algorithms, Programming | Student | 3-4, 5-6, 7-8, |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | programming | | | 9-10, 11-12 |
| Bootstrap | http://www.bootstrapworld.org/ | DT | Beginner | General purpose programming, Internet | Algorithms, Programming | Teacher (instructions/lesson plan) & Student (interaction with tutorial) | 7-8, 9-10, 11-12 |
| Alice 3D | http://www.alice.org/index.php# | Both | Beginner/Varied | Internet | Programming, Creating and Interacting Online | Teacher (instructions/lesson plan) & Student (interaction with tutorial) | 7-8, 9-10, 11-12 |
| Code Academy | http://www.codecademy.com/learn | DT | Can be beginner, but recommended some understanding of Algorithms, CS. | Visual Programming | Hardware & Software (incl. interface design), Programming | Student | 7-8, 9-10, 11-12 |
| Code Academy | http://www.codecademy.com/schools/curriculum | DT | Intermediate | Internet, Visual Programming | Digital Data (incl. binary), Algorithms, Programming | Teacher (professional learning), Student | 7-8, 9-10 |
| Khan Academy - Computer Science | https://www.khanacademy.org/computing/computer-science | DT | Beginner | General purpose programming, Internet | Digital Data (incl. binary), Algorithms, Programming | Student | 3-4, 5-6, 7-8, 9-10, 11-12 |
| Khan Academy - Computer Programming | https://www.khanacademy.org/computing/computer-programming | DT | Beginner | Internet | Programming | Student | 3-4, 5-6, 7-8, 9-10, 11-12 |
| Khan Academy - Computing Professional | https://www.khanacademy.org/computing/computer-programming/meet-the-computing-professional | Both | Beginner | Internet | Careers and applications of CS | Student | 3-4, 5-6, 7-8, 9-10, 11-12 |
| Code.org | code.org | DT | Beginner | Visual Programming, General purpose programming, Internet | Digital Data (incl. binary), Algorithms, Programming, Computational Thinking | Teacher (instructions/lesson plan) & Student (interactio | F-2, 3-4, 5-6. 7-8, 9-10 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | n with tutorial) | |