Asynchronous critiques via video to enable studio collaboration for employability skills in distance education

Final report 2021

Lead institution: University of New England

Project leader: William Billingsley

Team members: Huy Phan, Bing Ngu, Paul Kwan,
Nicolas Gromik

Additional contributors: Rosemary Torbay, Peter R. Fletcher,
Dambaru Subedi

www.une.edu.au

Support for the production of this report has been provided by the Australian Government Department of Education, Skills and Employment. The views expressed in this report do not necessarily reflect the views of the Australian Government Department of Education, Skills and Employment.

Higher Education Program Management
Governance Quality and Access Branch
Higher Education Group
Australian Government Department of Education
GPO Box 9880
Location code C50MA7
CANBERRA ACT 2601

<learningandteaching@dese.gov.au>

2021

ISBN 978-1-76114-102-7 [PDF]
ISBN 978-1-76114-113-3 [DOCX]
ISBN 978-1-76114-110-2 [PRINT]

Acknowledgements

We would like to thank the former Australian Government Office for Learning and Teaching for funding the project and for their collaborative engagement with us since.

We would also like to thank the students of the courses we have been examining, who undertook their work and critiques with enthusiasm and care. We would also like to thank past colleagues in the development and running of studio courses that inspired the project, including Jörn Guy Süß, Jim R. H. Steel, Richard N. Thomas, and Simon Kaplan.

The project lead would also like to thank Peter R. Fletcher, Rosemary Torbay, and Dambaru Subedi, who joined the project after it began and have contributed greatly to studies we have conducted. I look forward to continuing our collaborations into the future.

List of acronyms used

ACM	Association for Computing Machinery
ToCE	*ACM Transactions on Computing Education*
ASCILITE	Australasian Society for Computers in Learning in Tertiary Education
ICT	information and communication technology
IxD	interaction design
LTI	Learning Tools Interoperability standard
MOOC	massively open online course
OLT	Australian Government Office for Learning and Teaching
UNE	University of New England
UX	user experience

Executive summary

Studio courses have become a key way in which professional skills, especially those involving collaboration and design, are taught in several fields. Studios typically involve students working on a design problem (individually or in groups), periodically presenting their work, and critiquing the work of others. They support 'productive inquiry', teamwork, communication, and reflection. They also enable students to apply their knowledge to more realistic, multifaceted problems, and to learn from each other's project experiences, not just their own. However, this culture of collaboration and critique of work in progress is typically offered in on-campus modes and can be difficult to achieve for online and distributed classes. This project examined the dynamics of using an asynchronous video-based approach to critiques, in classes that predominantly comprise distance education students. In this approach, students are asked to submit video presentations of their work in progress, and then to record video critiques of each other's work.

Approach

We extended our open-source critique tool, Assessory, to schedule and manage a critique process based on posting and responding to videos. Assessory allows students to post their videos via cloud video services (including YouTube, easing upload for smartphone-recorded video) and uses a just-in-time, 'least critiqued' allocation strategy to help ensure that late-submitted videos will receive some critiques. We have explored the use of video critique using this tool as well as approaches that do not use a specialised tool.

First round

A pilot study was conducted in early 2016, in which different critique tasks were run in three teaching units in two disciplines: an Interaction Design class in computer science, and Learning Theory and Mathematics Pedagogy classes in education. The Interaction Design class used Assessory, whereas the education classes used alternative upload mechanisms. Analysis in this round was exploratory. Video proved to be an extremely flexible medium for delivering critique, and students took a broad variety of approaches. These included talking to camera, but also sketching alternative designs, using physical items to demonstrate a point, or screen recording the playback of the presentation video while providing a voice commentary and scrubbing to points of interest. These results were presented at the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE) in 2016 (Billingsley, Ngu, Phan, Gromik, & Kwan, 2016).

Second round

A second round of analysis focused on a distributed software engineering studio class in late 2016. This is a 'supercollaborative' unit in which groups of students work on different features of a class-wide project. This round used content analysis and past research on peer feedback from the literature to develop a model for classifying and analysing the approaches taken in critique videos. The results in this class gave support for the social value of critique in online studios – students predominantly adopted a social approach to deliver their message, using encouragement and suggestion, and focus on higher-level aspects of development work such as the 'mechanic' behind the group's feature. More detailed results are included in a paper published in an *ACM Transactions on Computing Education* (ToCE) special issue on global software engineering education (Billingsley et al., 2019).

Third round

A third round of data collection took place in 2017 in the Interaction Design unit and a unit on information and communication technology (ICT) in education. In this round, we extended the content analysis to understand the varying 'personas' that students adopt in presentation and critique. We developed a model describing how the flow of choices (e.g. in persona, organisational logic, and visuals) relate to each other in the presentation video and critique video, as well as how one video influences the other. This model informs how presentation and critique skills can be taught rather than only practised. A paper on these aspects is being written.

We have also found that in studios for topics where professional tools are available, there is an opportunity to make 'authentic analytics' available, thereby bringing another aspect of professional practice into the studio. An accepted book chapter (Billingsley & Fletcher, in press) explores this approach, and in future work Assessory will be adapted to connect to professional collaboration and communication tools, to surface analytics, and to allow conversational critiques.

Conclusion

In computer science, the introduction of aspects of studio pedagogies is a feature of UNE's newly redesigned undergraduate computer science degree. Software Development Studio 2 (formerly Software Engineering Studio) is the first Australian 'supercollaborative' software engineering studio designed predominantly for off-campus students. Supporting studio styles of teaching off campus is therefore a problem we needed to solve for that discipline. However, as studio pedagogies originated outside of computer science – being drawn from design, planning, and architecture – it is also useful to examine how the techniques we use to support studio pedagogies can also be reapplied in non-computing disciplines, such as education.

We also believe asynchronous video critiques would be useful to other institutions that have significant numbers of off-campus students and wish to implement studio pedagogies. As computer science has been a particularly strong adopter of collaborative studio pedagogies, we believe our results would be useful to computer science educators in both off-campus and on-campus formats.

Table of contents

Tables and figures

Tables

Figures

Chapter 1: Background

Studio courses have become a key way in which professional skills, especially those involving collaboration and design, are taught in several fields (Bull, Whittle, & Cruickshank, 2013; Kuhn, 2001; Levy, 1980; Long, 2012; Schön, 1987). Studios typically involve students working on a realistic design problem (individually or in groups), periodically presenting their work for critique, and critiquing the work of other groups. They support 'productive inquiry' – seeking knowledge when it is needed for a task – as well as teamwork, communication, and reflection. They also enable students to apply their knowledge to more realistic, multifaceted problems – to leave the highlands of theoretically solvable problems and enter the 'swampy lowlands' wherein lie the messy problems that are of greatest human concern (Schön, 1983).

Studio pedagogies are social and constructivist. They are designed to support productive inquiry, drawing on the theories of Dewey (1938). This was a particular influence on Schön when developing the concept of the reflective practitioner (1983) and describing studio as a means of applying this concept to education (1987). These pedagogies are also supported by theories of experiential learning (Kolb, 1984), and, more recently, researchers have sought to refine the theories behind reflection (Bleakley, 1999; Leitch & Day, 2000; van Manen, 1995) that usually accompanies studio practice.

Studio teaching approaches, and the pedagogical theories behind them, are particularly important for employability skills. John Seely Brown, who co-founded the Institute of Research on Learning in the 1980s, advocates productive inquiry and social learning when describing how education needs to change to produce graduates who are adaptable to modern, changing careers (Seely Brown & Adler, 2008). The Core Skills for Work Developmental Framework (Ithaca Group, 2013), referenced in the Australian Government Office for Learning and Teaching's (OLT) commissioned projects for graduate employability, lists interacting with others and getting the work done as two of its three skill clusters. The B20 Human Capital Taskforce Policy Summary (B20 Australia, 2014) includes problem-solving, critical thinking, collaboration, and interpersonal skills in recommendation HC3.5.

Among OLT's commissioned work, Oliver (2010) finds reflective learning to be a key facet of employability, and in her framework advocates work-integrated learning. She notes, from Patrick, Peach, and Pocknee (2008) and Little (2006), that this is not only work placements but also the integration of work-related 'authentic tasks' into the curriculum, and from Moreland (2006) that 'there is the implication that higher education programs must ... gradually confront [students] with complex, in-the-world activities that encourage risk assessment and reflection in their studies' (p9). The Curtin University project *Building course team capacity to enhance graduate employability* (Oliver & Whelan, 2011) developed Graduate Employability Indicators, which particularly included solving complex real-world problems, working effectively with others, and thinking critically and analytically. The current OLT project *Developing graduate employability through partnerships with industry and professional associations* (Jollands et al., 2015) has used Dacre Pool and Sewell's (2007) framework for graduate employability, which centrally features reflection and evaluation.

In computer science, studio courses are helpful for aligning curricula with workforce and industry needs. Professional skills, teamwork, communication, and project work with clients are Australian Computer Society accreditation requirements for degrees (ACS Accreditation Committee, 2014). Software development is also a field full of messy problems with no perfect technical solution (Brooks, 1987), and in the early 1990s, the first software studio courses began to be introduced (Tomayko, 1991; Tomayko, 1996). These have proliferated and there are now many different approaches to on-campus software studios (Bull et al., 2013; Docherty, Sutton, Brereton, & Kaplan, 2001; Hundhausen, Narayanan, & Crosby, 2008; Nurkkala & Brandle, 2011).

A number of technical skills in modern software production, such as distributed version control and continuous integration, require larger-scale collaborative practice to teach effectively (Süß & Billingsley, 2012). Industry advisory committees, including our own at the University of New England (UNE), often request collaborative reflective practice to be embedded throughout degrees. Some companies have also attempted to fill the gap in graduate employability using studio techniques. ThoughtWorks's *LevelUp* program, a volunteer-led studio for recent graduates, was developed in Australia in response to the discovery that some Australian universities do not teach modern collaborative software production practices. Part of its intent is to encourage universities to adopt studio teaching of modern professional practices so that the burden does not fall on industry (Leonor Salazar, ThoughWorks, private communication, July 2014) .

Studio collaboration is, however, difficult to achieve for off-campus classes. At UNE, most of the computer science students are off campus, and many would not be able to attend a synchronous virtual class. Teaching staff on-campus courses with large cohorts have also reported difficulty scheduling enough time in the class for every group to present its work for critique (Matthews, 2013). There is therefore a need for asynchronous techniques that can enable remote students to fully participate in studio teaching. In particular, we need asynchronous techniques to support the studio critique process that binds the class together.

Until now, there have been limited attempts to support asynchronous critiques in studio courses. These include two approaches at The University of Queensland (UQ), both of which ask for text critiques of in-person or video presentations (Billingsley & Steel, 2014; Matthews, 2013). Although both were designed with online teaching in mind, they were implemented in on-campus courses only. The situation of a student in a large on-campus cohort, who meets many students but cannot engage with all students, is significantly different to that of an off-campus student, who does not physically meet any other student in the course. With online learners, the massively open online course (MOOC) provider

NovoEd (Ronaghi, Saberi, & Trumbore, 2015) uses asynchronous critique-style feedback in some courses, but again only using text critiques, and outside of formal higher education. Saghafi, Franz, and Crowther (2012) experimented using text methods such as Wikis and Facebook comments for critiques in a virtual studio, but found that some students felt isolated using this method.

In this project, the merits of students critiquing other groups' work asynchronously via video to enhance studio collaboration among distance education learners are explored. Using video as a medium for critique allows students to present critiques more richly (e.g. including demonstrations of an issue, document-camera interactions, and using deixis). As the critiques, not just the group presentations, can be mediated via video, it also means that each student's voice can be heard by other students in the course. We hypothesised that this would improve the sense of studio as a community of practice.

We pilot asynchronous video critiques in computer science and education units at UNE. UNE is Australia's longest-serving continuous provider of distance education and has many more off- than on-campus students. Our intent for the project is that it should enable us to support studio pedagogies for our predominantly off-campus cohorts, and increase engagement and the sense of class cohesion in remote group work.

In computer science, the introduction of aspects of studio pedagogies is a feature of UNE's newly redesigned undergraduate computer science degree. Software Development Studio 2 (formerly Software Engineering Studio) is the first Australian 'supercollaborative' software engineering studio designed predominantly for off-campus students. Supporting studio styles of teaching off campus is therefore a problem we needed to solve for that discipline. However, as studio pedagogies originated outside of computer science – being drawn from design, planning, and architecture – it is also useful to examine how the techniques we use to support studio pedagogies can also be reapplied in non-computing disciplines, such as education.

We also believe asynchronous video critiques would be useful to other institutions that have significant numbers of off-campus students and wish to implement studio pedagogies. As computer science has been a particularly strong adopter of collaborative studio pedagogies, we believe our results would be useful to computer science educators in both off-campus and on-campus formats.

Sector readiness

For on-campus cohorts, studio teaching has seen an uptake in many disciplines (Levy, 1980; Long, 2012; Schön, 1987). Computer science has been a particularly enthusiastic adopter of studio pedagogies – computing was quick to recognise the importance of collaborative design and reflective practice to the discipline (Brooks, 1987), and from the 1990s sought to adopt studio teaching from architecture (Docherty et al., 2001; Hazzan 2002; Kuhn, 2001; Tomayko, 1996). As on-campus studio courses in computing proliferated, academics adapted the pedagogies to fit the needs of the field, and there are now many variations (Billingsley & Steel, 2013; Bull & Whittle, 2014; Carter & Hundhausen, 2011; Hendrix, Myneni, Narayanan, & Ross, 2010; Hundhausen et al., 2008; Nurkkala & Brandle, 2011; Reardon & Tangney, 2015). Australia has been at the forefront of this, and studio courses and collaboration are now embedded in many Australian universities' on-campus computing and design degrees. Although the education discipline has not traditionally used studio approaches, Jordan (2012) has proposed using video for recording feedback exchanged among learners in professional development for teachers.

There have been recent endeavours to develop asynchronous studios (Billingsley & Steel, 2014; Matthews, 2013; Ronaghi et al., 2015; Saghafi et al., 2012) and, as discussed previously, OLT projects on employability have recognised the importance of reflection and evaluation (Jollands et al., 2015) and working on complex real-world problems (Oliver, 2010; Oliver & Whelan, 2011).

The use of video as a way that students in authentic learning tasks can present their work for assessment and reflection has also been a topic of a number of OLT projects. The PHENC Project (Hands et al., 2009) asked students in physical education, education, nursing, and counselling degrees to videorecord themselves in practical tasks and analyse their performance using the Dartfish video analysis tool. The eCAPS project at UQ (Engstrom, Hay, Macdonald, Brukner, & Khan, 2011) asked clinical students to record themselves demonstrating various clinical practical skills, and to take part in synchronous video assessment of their practical skills over Skype. This suggested a two-dimensional scale for learner independence, depending on the interactivity of the learner and the interactivity of the content.

Chapter 2: Technology Testbed

Although this project is not primarily focused on the creation of new technology, a technology testbed was needed in order to allow us to experiment with how critiques should work and how they relate to the wider assessment and tasks in the courses.

In previous work, the project lead had co-developed a 'supercollaborative' software studio course at UQ (Billingsley & Steel, 2013; Süß & Billingsley, 2012) in which teams of students worked on different features of a common class-wide project. From the 2013 iteration, this course used a custom open-source system, Assessory, he developed for scheduling critiques and allowing students to review their critiques (Billingsley & Steel, 2014). In this on-campus course, students needed to know in advance of the presentations which groups they would be asked to critique, ensuring a fair distribution of critiques and also ensuring that students were only asked to critique other groups who were in the same tutorial session as themselves. Authentication was managed by OAuth using students' GitHub accounts, as this was the system students were already using for software collaboration.

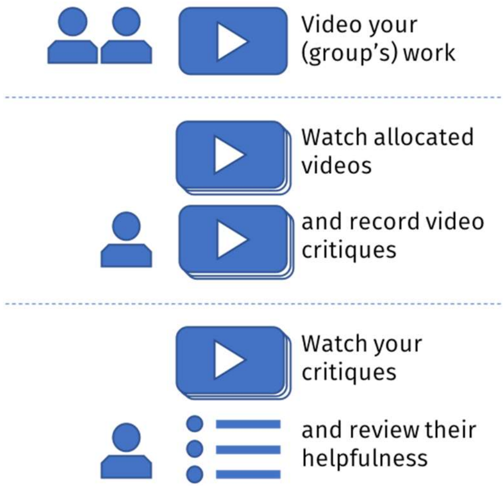

Figure 1: Three-stage critique process in Assessory

At the start of this project, we adapted this technology to suit distributed classes and the submission of presentations and critiques via video. This introduced a three-step critique process, as shown in **Error! Reference source not found.**.

First, students would record their presentation as video; unlike the on-campus studio, they would not be able to present in person. They would upload this video to a cloud media provider, such as YouTube (or, later, Kaltura). This allows us to take advantage, for example, of most students' smartphones, which already contain the functionality to compress and upload video to YouTube.

Second, students would access a task that would allocate them presentations to watch and critique. The allocation mechanism here was adjusted to be dynamic. Whereas on-campus students needed to know in advance which presentations to watch, as the presentations all occurred in a scheduled session, in this case the allocation needed to take into account that videos would be submitted progressively. By default, a 'least critiqued' algorithm is used, whereby every time a student accesses the task, they are allocated the videos that have so far been allocated to the fewest critics. However, we have found this is still susceptible to the problem where a student accesses the task, receiving an allocation, but then leaves the course causing the system to count critiques that might never be completed. As a potential mitigation to this, we allow the task to be restricted on whether the previous task has been completed, with the view that a student is less likely to drop out of a course in the short space of time between having produced and submitted their own video and critiquing others'.

Third, students access a task that allows them to review the critiques their (or their groups') work has received. This typically requires students to complete a short form, often a four-question survey, on whether they found the critique constructive, specific, actionable, and useful.

We also adapted Assessory to accept logins and automatic registrations via the Learning Tools Interoperability (LTI) standard, so that the tasks are accessed seamlessly from the assessment block in Moodle, as shown in Figure 2.

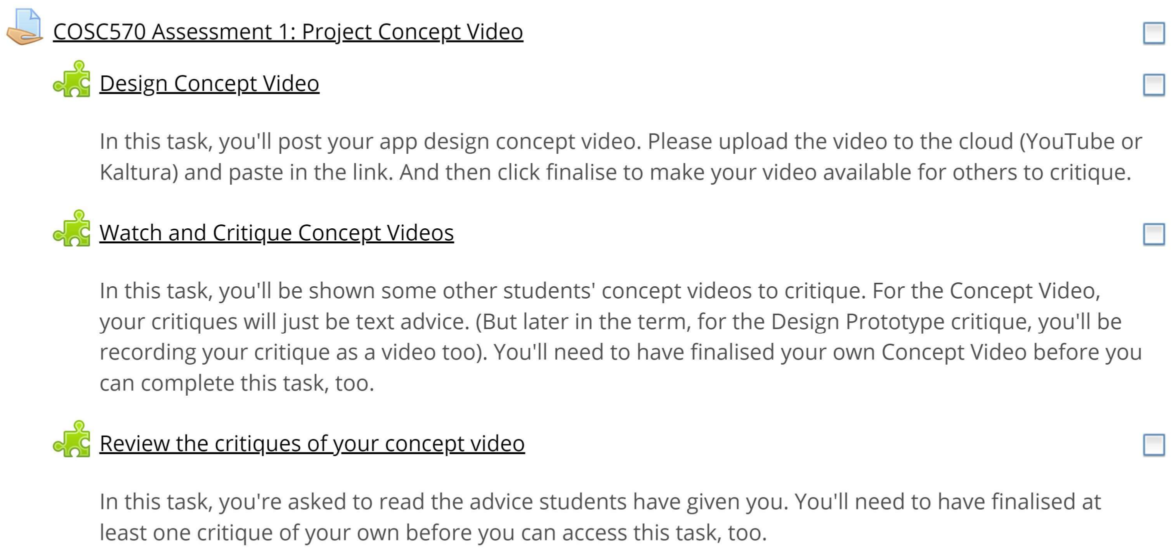

Figure 2: Each critique stage operates as an LTI link from Moodle, automatically registering students as necessary.

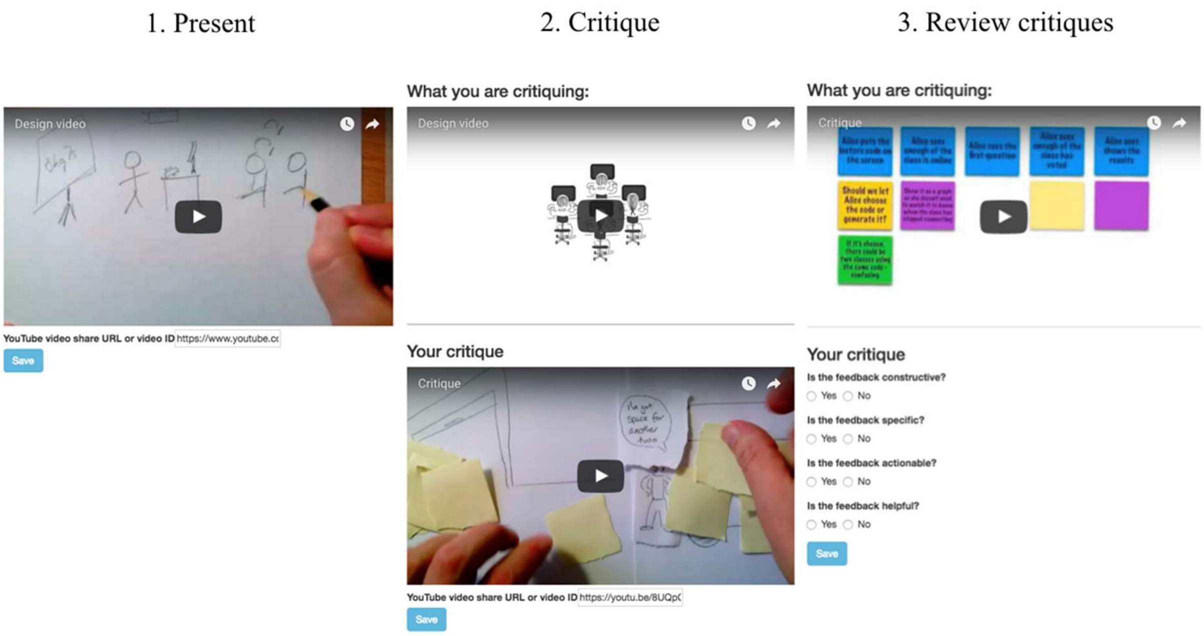

Figure 3: The three-stage critique process, as it initially appeared to students. Figure shown in Billingsley et al. (2016). For privacy reasons, the videos in these screenshots show staff-generated videos rather than genuine presentations and critiques.

Figure 3 depicts this process as it appeared to students in 2016, the figure taken from our 2016 ASCILITE presentation (Billingsley et al., 2016). The system continues to be developed, and in more recent courses, the tasks may look a little different depending on what questions have been set.

Behind the scenes, each of these tasks operates as a questionnaire. As students upload their videos to YouTube or to the university's Kaltura video repository (dubbed 'My Media'), the video submission acts as a question type that can recognise video URLs or embed codes and replace them with an appropriate video player. The primary aspect that Assessory handles is the need for one task output to be a targeted response to another task output. For example, in the critique stage, completed design concept video questionnaires are selected for students to critique using a critique questionnaire that can include video questions. In future work, this also gives us flexibility to introduce question types that integrate other external systems, such as integrating authentic analytics from a system like GitLab.

We are also able to mark certain questions as 'hidden in critique'. This allows us to ask participants research questions alongside their assessment work while keeping their research responses confidential from both the students allocated to critique the work and the marker of the work.

As a technology testbed, Assessory currently lacks an administration user interface for creating the units. Rather, they are set up via a programmatic script. This ties into a separate work program where we intend to allow courses and assessments to be defined by text files written in a human-readable text notation, thereby allowing course definitions to be committed to version control, and allowing courses to be managed using techniques that have proven successful for managing software and other digital product releases. An example of the current script is shown below in Figure 4, although this is not intended to be the eventually supported format as it requires a Scala compiler.

```
TaskDetails(
    name = Some("Critique Concept Videos"),
    description = Some(
        """
        | When you click into this task, three other students' work will be picked for you
        | to critique. (You'll need to click the '1', '2', '3', etc, to see it)
        | You are asked to watch their video and offer some feedback that should be
        | constructive, objective, specific, actionable, and useful.
        """.
        stripMargin),
    restrictions = Seq(MustHaveFinished(dv.id)),
    individual = true
    ),
    body = CritiqueTask(
    strategy = AllocateStrategy(what = TTOutputs(dv.id), number=3),
    task = QuestionnaireTask(Seq(
        VideoQuestion(
        id = TaskDAO.allocateId.asId,
        prompt = "Please upload your video to YouTube or MyMedia and post the link (or MyMedia embed code)"
        ),
        ShortTextQuestion(
        id = TaskDAO.allocateId.asId,
        prompt = "If you would prefer to reply via text, you can type in text here", hideInCrit = false
        ),
        ShortTextQuestion(
        id = TaskDAO.allocateId.asId,
        prompt = "What did you find easy about providing feedback? (Your answer is only visible to staff)",
        hideInCrit = true
        ),
        ShortTextQuestion(
        id = TaskDAO.allocateId.asId,
        prompt = "What did you find difficult about providing feedback? (Your answer is only visible to staff)",
        hideInCrit = true
        )
    ))
)
```

Figure 4: Assessory tasks are currently set up using Scala scripts. This is not intended to be the future format, but is a testbed for the principle of keeping course definitions in text formats under version control. Additional support of other options, including human-readable JavaScript Object Notation (HJSON), is intended for future versions.

Chapter 3: First Pilot Study

In Trimester 1, 2016, we ran a small pilot study of a video-based critique process across three units of teaching: User Experience and Interaction Design (UX and IxD) in computer science, and in Learning Theory and Mathematics Pedagogy subjects in the School of Education. This pilot study is primarily reported in Billingsley et al. (2016), so there is some overlap between that paper and the following description.

This pilot study was run earlier than originally intended owing to some team members becoming unavailable during 2016. This caused the design of the pilot study to be concurrent with the adaptation of the technology. Accordingly, this limitation was used as an opportunity first to gather coarse-grained data on the practical issues that teachers and students could expect with the introduction of a video-based critique process. Rather than run all three units using the same technology and task design, each unit coordinator (a team member) was asked to design the task in their unit. We considered this appropriate as in practice teachers can always be expected to tailor pedagogies to their classes, and this would let us explore even within the team how tasks can be expected to be adapted.

The unit coordinators in education were cautious of using a YouTube upload mechanism, especially as the trials would now be occurring so close to the technology development. Consequently, these units requested not to use Assessory. Because the university, at the beginning of 2016, did not yet support student uploads to its video management service, a mechanism based simply on form-upload of a video file was used temporarily. The coordinator of the Learning Theory unit also preferred students not to see each other's work, so in this unit students were provided with a staff-generated video to critique. This led to a qualitative pilot study in which three units ran in three very different modes – with Assessory and the full three-stage critique process used in the UX and IxD unit. These are summarised in Table 1.

Table 1: Pilot study unit summary from Billingsley et al. (2016)

Unit	Topic	Presentations	Critiques	Other
A	Interaction design	14 videos	39 videos	11 demo videos
B	Learning theory	Lecturer-provided videos	15 videos, 4 audio	
C	Mathematics pedagogy	12 videos	25 text	

The pilot study – though early, rudimentary, and exploratory – revealed useful information for refining the later stages of the study, logistically, socially, and in terms of confirming hypotheses about the role that video critique would take within units. As this was a pilot study, the results are presented here as observations.

Variety in use

Part of the motivation for video as a presentation medium is its flexibility, for example, in supporting deixis (such as pointing to items on paper and saying 'this' while elaborating on the reasons for a design).

Especially in the UX and IxD unit, we found a great variety in how students produce their presentations and in how they produce their critiques. Design videos included animations, recorded digital presentations (e.g. Keynote or PowerPoint), recorded on-paper presentations (using coloured notes in place of slides), screen recordings explaining documents, and voice overs talking through sketches and design mock-ups. A similar variety was seen in the critique videos, including recording one phone from another while swiping through design mock-ups, physically manipulating alternative paper prototypes, and use of physical items to represent controls in a suggested alternative design.

One particularly efficient and popular critique mechanism was to screen record the playback of the design video. This allowed the critic to scrub to points of interest and give a running commentary while using the presenter's own imagery in their explanations. This has since been taken up as a suggestion for students in some busier and more technical units where the depth of critiques is not expected to be high but there is perceived educational value in asking students to perform (rather than receive) critique and learn from experiences other than just their own. For example, in the Advanced Web Programming unit, critique has been introduced as a way to allow students to choose one technology for their own work (e.g. Vue.js) but also observe others' development journeys with competing technologies (e.g. React or Angular).

Privacy concerns over appearing in video

In Unit B, a small number of students expressed concern about whether they would need to physically appear in the video. Five students did not upload video (four uploaded audio only and one uploaded text).

An additional privacy factor we have since observed in later units is that this can also arise in the context of transgender students concerned that by appearing in a video they will reveal their gender identity to online peers who might previously have been unaware. In practice, this concern can be alleviated. In the majority of submissions, students frame their videos to show the work being discussed and the presenter does not appear on screen. So a concerned student can adopt this format and retain their confidentiality.

Generally, we find it useful ahead of the task to demonstrate video presentation styles that do not involve being on screen as well as those that do, to normalise some variety in presentation styles and encourage students to consider the context of what they are presenting in how they construct their video. We have since observed a number of other workarounds for students with concerns about presenting, such as the use of stand-ins for students with a severe physical disability, as well as the use of captioning or automated text to speech.

Logistical aspects of Assessory

Comparatively few issues were identified with the use of Assessory. A small number of videos were accidentally marked private in YouTube; however, this was typically quickly resolved over email, having been flagged by the student allocated to watch the video. In future, the YouTube API may allow us to build an automated check into Assessory to alert to videos that are set as ‘private’ at the time of posting.

The form-based upload that the education units used instead of YouTube in the pilot study proved to be problematic. Many of the videos were produced using students’ smartphones with high-resolution cameras, resulting in exceedingly large files. Without the advantage of their phone’s ready-provided compression and upload facility for YouTube, they were faced with the difficulty of transferring their video. In some cases, students were finding it difficult to transfer their video to their computer before they could access programs to downscale the video. In others, students attempted to upload full-scale video and underestimated how long it would take, believing that there must be a connection issue for the upload to take that long.

In the time since this pilot study was conducted, the university has introduced its own video cloud for students, which has been dubbed ‘My Media’ and based on Kaltura. Assessory has been altered to support both platforms, and we have observed an approximate 2:1 ratio within computer science units of students preferring YouTube over Kaltura for their video upload. This may be because YouTube upload is already integrated into many platforms and automatically handles downscaling the video to a reasonable size.

Kaltura is, however, successful in education units and has the advantage of being officially supported by the university. As it is now practical to mediate a simple post-and-review process using Moodle and Kaltura, Assessory is typically only used in the computer science units where a full three-stage critique process is used. As discussed later in the report, there is an opportunity for Assessory’s unique advantage to move to its integration with other external platforms, for example, automatically capturing analytics from professional collaboration tools and platforms for use in critique.

Notification and preservation

Assessory currently lacks a notification mechanism for informing students when a new critique of their work is available for viewing. This can require students to tediously check back to see if there are new critiques available, although in practice few complaints about this have been received.

Where students do submit via YouTube, there is the question of how long the video will be available for. In practice, the youtube-dl open-source package can be used to download students' videos for archival purposes, and this is scriptable through Assessory to download all student videos for a presentation or critique task. However, this package is not officially supported by YouTube.

Chapter 4: Distributed Software Engineering Studio

After the initial pilot demonstrated that the video-based critique process was indeed feasible, we conducted a content analysis of student critique in our second-year software engineering studio unit.

Further details on the course and the results presented in this chapter are included in a journal article published in a special issue of ToCE on global software engineering education (Billingsley et al., 2019).

Unit background

COSC220, now titled Software Development Studio 2, is a ‘supercollaborative’ course in which the entire class operates as a distributed software development team. The class is given a common project to develop – typically a networked multiplayer game – and groups of students form around different features that they will develop. We call this *supercollaborative* because it involves two layers of collaboration: students collaborate with their group mates on their feature, but the groups in turn collaborate with other groups on the project as a whole.

The unit design stemmed from previous work developing a supercollaborative software studio on campus at UQ (Süß & Billingsley, 2012). In the 2012 on-campus offering, only 18 per cent of ‘commits’ (shared changes to the common code base) were observed to have occurred during scheduled class time, as shown in Figure 5, and that it could be argued that ‘many on-campus students are effectively taking the course virtually rather than physically’ (Billingsley & Steel, 2013, p5). This suggested that it should be viable to employ a similar design for online students and we had also in the past modelled how the course design could operate as a MOOC (Billingsley & Steel, 2014).

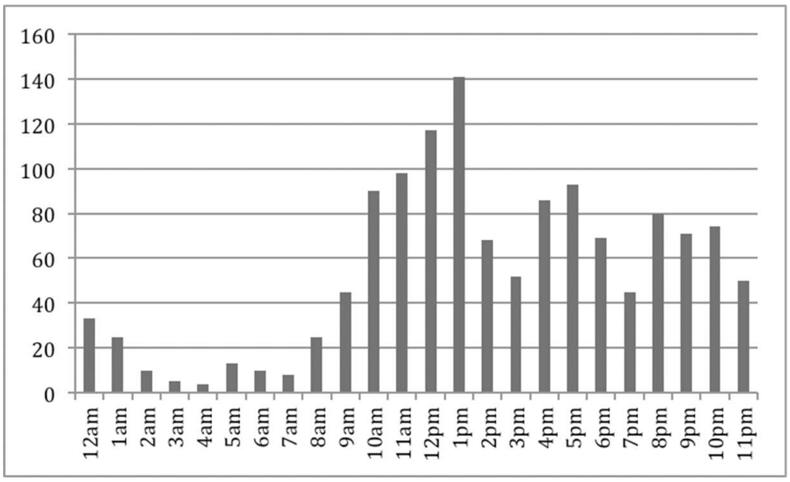

Figure 5: Timing of student commits to the common code base in the on-campus offering of a supercollaborative software studio course at UQ in 2012. Figure excerpted from Billingsley and Steel (2013).

Critique content analysis

Forty-eight videos from the 2016 iteration of the unit were selected for analysis, comprising 12 presentation videos and 36 critiques. These were analysed using an inductive approach to video research (Erickson, 2006) that involved repeated viewings with progressively deeper analysis.

In advance of the content analysis, a literature review was conducted on peer and tutor feedback to identify and refine appropriate frameworks for the analysis. The full literature review is not reproduced here (for further detail, see Billingsley et al., 2019), but the developed model drew in particular upon Tseng and Tsai's (2007) notion of *feedback type*, Brown and Glover's (2006) categories of *motivational praise*, and the *suggestions* and *examples* categories from Gielen and De Wever's (2012) scoring rubric. The framework was revised iteratively among the project team while examining the categories and codes that emerged from content analysis of the collected videos. This resulted in the addition of several themes that were not present in the literature; for example, whether the presentation videos focused on structural, visual, code, or 'mechanic' aspects of their development work. This led to an analysis framework based on the following themes:

- The focus of the team presentation video under critique – whether it was on the technical detail of the program code or higher-level aspects such as the structure of the design, mechanics of the feature, or aesthetics of the visual design
- The corresponding focus of the critique video
- The mode of the critique video – particularly the choice of what to place on screen (e.g. face to camera, camera to paper, text on screen, computer screen, supporting diagrams, or images)
- The feedback type in the critique video – reinforcing, suggestive, didactic, or corrective
- The quality of feedback in the critique video – including the organisational logic, presence of evaluative or descriptive motivational praise, and presence of abstract or concrete suggestions.

None of the categories in the analysis were considered mutually exclusive; for example, a critique video may begin focused on the structure of a team's feature before progressing to discuss its mechanics. Likewise, a video might be recorded focused on the student's computer screen, which may be used to show combinations of text on screen and images on screen.

Selected results

Regardless of whether the presentation video being critiqued focused on the detailed program code of the work or higher-level aspects, we found that critique videos predominantly focus on the mechanics and visual aspects of the feature being developed.

This is summarised in Table 2. Although approximately one-third of the critique videos referred to code, it was with low focus.

Table 2: Summarised results of the analysis of critique videos against the 'reviewer focus' theme

Categories	Prevalence (*n* = 36)	With low focus	With high focus
Visual	58%	42%	17%
Mechanic	88%	31%	58%
Coding	36%	28%	8%
Structure	33%	3%	31%

Previous to the study, we had realised the presentation videos would in turn have a shaping effect on the critiques (it is difficult to critique aspects that are not in the presentation). This, however, provided the additional confirmation that the medium and context also have a shaping effect. Students have access to each other's code in the project version control repository, and during the course of the project can be expected on occasion to interact with and alter the code of other groups whose features they need to collaborate with. In the critique task, however, students are presented only with the team's video and asked to respond with a video of advice. It therefore becomes appropriate for students' critique videos to focus on higher-level aspects of the feature design, whereas discussions on code alterations can take place directly in the code collaboration tools.

We found the students' critique focus encouraging in terms of the aims of studio critique in the unit. The studio is not primarily a programming language skills unit, but a unit on distributed software development, and studio critiques are intended as an avenue for reflective practice and higher-level thinking.

In this class, there was greater variety in the presentation mode (what was shown on screen) for the critique videos than for the presentation videos. The presentation videos all used a slide-based presentation with spoken narration to document the team's design work – this was not the case for all years and may have been a factor of the particular class project that year having a looser coupling between the teams' features. The critique videos, however, predominantly showed the student's computer screen (78 per cent), followed by face to camera (15 per cent) and camera to paper (seven per cent). Text on screen was the major prop, present in all critique videos, with supporting diagrams (39 per cent) and images (22 per cent) also widely used.

Qualitatively, these results are somewhat different from those in the pilot study, where, for example, tangible interactions with physical items also appeared. This is presumably due to the contextual difference of the project being studied – tangible interactions can be expected to be more common in interaction design, where the construction of paper prototypes is a teaching topic, than in software engineering. However, they do confirm that video provides a richer medium for critique that allows more variation in approach, with a significant number of critique videos providing supporting diagrams, images, or paper that would not have been possible with text-only critiques.

The discovery that only 15 per cent of critique videos showed the student's face also helps address some of the privacy concerns raised in the pilot study (and by individual students occasionally since). As the majority of students do not physically appear in their video, students concerned that appearing in the video would reveal an aspect of themselves to their online peers they had wished to keep private can choose not to appear in the video without this seeming unusual.

We also found that students primarily take a social approach to providing feedback. Ninety-five per cent of critique videos included *evaluative* motivational praise, with 49 per cent also providing *descriptive* motivational praise giving greater detail within a specific context. Eighty-three per cent of critiques offered suggestions for improvement, with 37 per cent providing only *concrete* suggestions for immediate improvements, 6 per cent providing only *abstract* higher-level suggestions, and 44 per cent offering an approximately even balance between concrete and abstract suggestion. The organisational logic of critique videos was considered clear and reasonably well structured in 72 per cent of cases. Most videos, when classified against feedback type, predominantly used reinforcing or suggestive feedback, as shown in Table 3.

Table 3: Feedback type in critique videos

Feedback type	Prevalence
Reinforcing	47%
Suggestion	33%
Didactic	14%
Corrective	6%

The results around the feedback in the critique videos were also encouraging for how video-based critique can enhance the social aspects of the course. As the UNE student cohort predominantly comprises online students, they rarely see or hear their fellow students. While the critique videos tend not to show their faces, they almost invariably include their voices. That the task should cause them to hear those voices, giving them predominantly well-structured and constructive feedback, including motivational praise, seems inherently helpful for the social fabric of the course.

Chapter 5: Third Round Analysis and Site

A third round of content analysis was conducted across units from Trimester 1, 2017. In this case, the units selected were a School of Education unit ICT Across the Curriculum (herein abbreviated as ICTinEd) and the computer science unit User Experience and Interaction Design (herein abbreviated as UX&IxD). This round of analysis was intended to help create a stronger theory of how aspects of presentation and critique relate to each other, as well as to explore the dynamics of critique in a wider variety of units.

An article describing the results in this chapter is in preparation for future submission to a relevant journal. As the model derived from this analysis is still being refined for publication, the results described in this chapter are kept brief and high level.

Units and tasks

In the ICTinEd unit, students were asked to design a 'webquest' task for school students to undertake. The presentation video focuses on the design of this webquest, with students then asked to critique the webquest of one other student. This was mediated over Assessory, with a simplified questionnaire for the 'Review my critiques' stage that asked open response questions on what students found useful, how clear and appropriate the critique was, and their overall impressions of giving and receiving feedback. Forty-five webquest videos and 45 critique videos from this unit were selected to be studied.

In the UX&IxD unit, students have a trimester-long project to design a mobile app using a process based on design thinking. Students initially pitch an app concept, including the intended target users and how they will research them. In the first half of the trimester, students focus on user research, defining the problem, and ideation, leading to a low-fidelity prototype that is tested with users. In the second half of the trimester, students then engage in a technology spike, developing some part of the app into a working prototype that is once again user tested. Students are asked to present videos at the concept and low-fidelity prototype stages of the course, each time also critiquing three other students' videos using Assessory's three-stage critique process. Twenty-two presentation videos and 51 critique videos were selected from this unit for study.

Methodology and goals

The methodology for the analysis in this study built on the content analysis from the collaborative software studio course. We had identified a small set of themes and characteristics, partially from literature and partly emerging from the coding of the data. We now sought to elaborate this against more videos in multiple units in order to develop a model of how the different aspects of presentation and critique relate to each other.

We had, for example, earlier identified that when conducting critique, the content of the stimulus and the context and medium of critique are likely to impact on the content of the critique. It is difficult for a critic to comment on aspects that were not present in the video under critique; for instance, in the software engineering unit, the context had (correctly) led most students to comment on higher aspects of the design rather than lower-level code. However, we wished to bring more shape to this analysis and develop a model that could be used to teach and improve critique skills.

In this phase, a three-phase methodology was used:

1. Drawing on the context and stimulus for each case to establish the emerging themes
2. Matching these themes against the codes identified from the literature and previous study (additional literature was brought in at this stage) in order to further develop the analysis framework
3. Coding the videos from the cases against the enhanced framework in order to compare and contrast the emerging patterns.

This developed a detailed coding scheme for the presentation and critique videos, versions of which are shown in Appendices B and C.

Emerging concepts

As well as elaborating the coding model, we added the concepts of 'genre' and 'persona type' in how a student conducts a presentation or critique. For example, in genre, the student might incorporate the following elements into their presentation or critique:

- Narrative – for example, storyboard walk-throughs chronologising the 'what' and 'how' of design
- Factualisation – providing a user guide instructing in procedure and the 'who', 'what', 'when', and 'where' of the output
- Engagement – persuasive messages made dramatic, in role or in context, with an intent to be 'punchy' and journalistic.

Persona types are inspired from the presenter types introduced in corporate advice as a means of teaching presentation skills (Decastro, 2014; Feloni, 2014; McMahon, 2014) combined with Cooper's concept of personas from the field of interaction design (Cooper, 1998; Cooper, Reimann, Cronin, & Noessel, 2014). In interaction design, personas form a practical means by which a designer can consider the variation among their users (for which they may have a bewildering array of data) by hypothesising them as a smaller number of imaginary people, complete with names, goals and backgrounds, representing particular kinds of users.

It is hoped that by identifying persona types this would give students a similarly practical tool to think about the organisational logic of their presentation depending on what they are trying to convey. Some of the draft persona types used (some of which are taken from McMahon, 2014, but some of which are derived from the data) include:

- Analyst – data, analytics, facts; quick transfer of organised and focused information, economical and to the point
- Coach – moves quickly and energetically through the presentation; conversational and talking in the first or second person
- Counsellor – structured, with information confidently and logically presented in an easy-to-follow manner, though can sometimes appear clinical
- Drone – stays mostly with what is written on the slides
- Inventor – connector of ideas, builds logical sequences to convey the details of design in a comfortable, sometimes impromptu style
- Storyteller – depth and detail conveyed through anecdotes, stories, and experiences
- Teacher – ideas in logical steps, well structured in organisation and delivery, and driven to present ideas diverting the focus from the audience
- Producer – well-structured and unfolding presentations that outline and build key points without deviation.

We had previously observed (anecdotally) that students tend to adopt the same mode of delivery across all their critiques; for example, if they screen record the playback of the presentation for one video, they will probably also do so for the other two. We were interested in understanding also whether people would adopt a preferred persona for all videos they conduct, or whether they adjust their manner of presentation contextually depending on the subject under critique.

Our analysis here is still ongoing; however, we do see many instances where students switch persona from video to video, so students are engaged in contextual decision-making on their video strategy. There are some persona types that can on occasion be consistent; for example, students adopting the drone persona in all three critiques, which may be indicative of a need to improve communication skills, or the inventor or analyst personas.

An emerging model

From the data analysis and emerging concepts, we have begun to produce a working model of critique planning that may be useful in teaching presentation and critique skills. This comprises two phases corresponding to the presentation and the critique; however, naturally, the critique takes the presentation as one of its inputs. This model is presented in Figure 6, although it is a work in progress. It shows, however, how the critique side of the flow has the additional complexity that, as well as considering the product and audience, the critic must consider their own particular area and level of expertise in how they conduct their critique. This in turn influences the persona they can adopt, as it regulates the level of detail they are able to go into and what logical structures they will be able to use in their critique.

As well as providing a means of articulating what students should think about when developing critique, it also suggests potential new communication skills exercises; for example, requiring students to adopt a particular persona type in order to gain experience in delivering a different style of feedback.

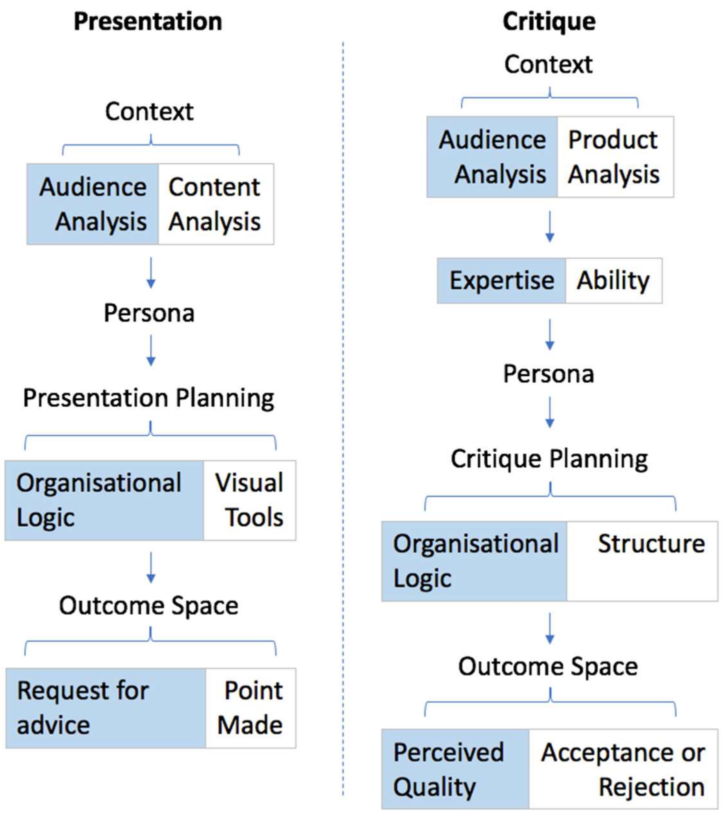

Figure 6: Working model of the presentation and critique planning process for use in teaching critique skills

Site

To improve the ways in which critique can be used and taught in the future, we have developed a site (currently internal to UNE) gathering resources and examples. Currently, this includes explanations, literature, and examples for persona types, social signals, logical organisation, and peer feedback.

Over time, this will be augmented to include example exercises as well as teaching materials and reusable exercises based upon the working model of the presentation and critique planning process.

Chapter 6: Outputs and Dissemination

Over the course of the project, asynchronous critiques using video have been implemented experimentally at some level of detail across seven teaching units at UNE in the disciplines of computer science and education. They have become regular practice in four computer science units. As well as the units already described, asynchronous critiques have been adopted into practice in additional computer science units for a variety of reasons:

- **Software Development Studio 1:** We identified that although we had significantly improved the teaching of group work and professional collaboration in the degree, there were shortcomings in how that group work was scaffolded. In 2017, the university introduced a first-year software development studio in order to introduce collaborative group work to online students at the beginning of their degree. A critique process using Assessory is used in the third assignment, where groups develop a small game (such as Asteroids) using Processing.

- **Advanced Web Programming:** Advanced Web Programming teaches modern web development, which is (technologically) a fragmented and fast-moving space. There are multiple front-end frameworks, such as Vue.js, React, and Angular, that students might wish to use. Likewise, there are multiple compile-to-JavaScript languages, such as TypeScript or ECMAScript 2015, and multiple server-side frameworks and languages that can be used. Accordingly, from 2017, Advanced Web Programming adopted a studio-style approach including critiques using Assessory so that students could choose one technology stack for their own development, but also observe others' experiences in the class using alternative technologies.

Outputs

The following outputs have been produced as part of the project:

- **Assessory** has been augmented to support questionnaire tasks including video questions and an asynchronous video-based critique process. This tool is in regular use across four units in computer science at UNE, as well as having previously supported UQ's implementation of the studio course. However, the intention of this open-source platform is primarily to be a testbed for further development of social assessment tasks. For example, as a by-product of this work, we have identified it as a means to bring authentic analytics (analytics of the students' work produced by the professional tools themselves) more fully into the critique process and to act as a hub between the teaching tools in classes and authentic collaboration tools.

- The **distributed 'supercollaborative' software engineering unit**, now renamed Software Development Studio 2, is in operation. This forms a central component in the university's Bachelor of Computer Science degree, with growing enrolments. At the time of writing, its fourth cohort is undertaking the unit, with 75 students collaboratively developing a two-dimensional sandbox game with integrated modding and scripting – a game to inspire children to learn to code. This is the third iteration to use asynchronous critiques via video. The unit is, to our knowledge, the world's only online supercollaborative software development studio unit, and is the subject of a paper accepted to a special issue of ToCE (Billingsley et al., 2019)

- Three rounds of analysis of students' video-based critiques have been conducted, across five different units in computer science and education, at successive levels of depth. This encompasses more than 170 critique videos that were selected for analysis across the three phases. This comprises more rounds of analysis, more units, and more videos than were originally planned in the project.

- This has enabled us to develop a **resource site** at UNE. This site is intended to assist students in developing skills for planning rich design task and critique presentations. It also provides resources to assist lecturers in planning formative peer-critique tasks for students and a repository of literature related to feedback and peer critique.

- We have begun to develop a **draft model** of the dynamics of critique that will allow critique skills to be taught more explicitly, including practical tools such as persona types that can suggest communication skills exercises.

Dissemination

Through the course of the project, the following papers and presentations have been published, accepted, or are in preparation:

- A conference presentation and publication at ASCILITE in 2016 (Billingsley et al., 2016). This primarily presented the pilot study work described in Chapter 3 and Assessory as a tool for facilitating and coordinating asynchronous video critiques.

- A journal paper (Billingsley et al., 2019) has been published in a special issue of ToCE on global software engineering education. This paper focuses particularly on the 'supercollaborative' software engineering studio, including the content analysis of video critiques described in Chapter 4.

- Under an agreement between the ACM ToCE journal and ACM SIGCSE conference, by which a subset of accepted ACM ToCE journals are presented at the following year's ACM SIGCSE conference, our ToCE paper was also presented at the ACM SIGCSE conference in Minneapolis, USA on 1 March 2019.

- A lightning talk and accompanying paper (Billingsley, 2019 February) was presented at the SPLICE Spring 2019 Workshop in Minneapolis, USA. The SPLICE workshop considers interoperability between tools for computer science education. The lightning talk covered Assessory, and in particular, future plans to incorporate techniques from earlier projects on intelligent teaching materials and course self-publishing formats.

- A by-product of the work has been an accepted book chapter (Billingsley & Fletcher, in press) exploring how more authentic analytics can be used to augment authentic tasks such as studio development projects. This developed from our experiences incorporating Assessory with the supercollaborative software engineering studio. Analytics from the professional development tools are already used in the assessment of students’ work and some are already available to students; however, they are not always well surfaced. Assessory is a tool that already integrates with third-party systems (including support for OAuth integration with GitHub IDs as well as using video cloud services). It is a small step to consider bringing in visualisations of students’ work (e.g. their commit history) into Assessory in the critique. The book chapter continues this line of thinking further, for example, exploring how these authentic analytics from professional tools can also be integrated with smart advice systems in order to give students rich automated feedback on professional work practices.

- A conference paper (Billingsley, 2019 May) presented at the 2019 International Conference on Software Engineering (ICSE) in Montreal, Canada discusses the 'fragmentation' of student cohorts as the flexibility of study causes classes to comprise of students studying in different modes from different locations. Assessory, video critiques, and techniques that were introduced for data collection in this project are discussed as ways of converting that fragmentation into an advantage rather than an isolating experience.

An ongoing site, initiated by a project member before the project, maintains links and information to supercollaborative projects, studio pedagogies, and video critiques. This includes resources and links from this project. https://www.supercollaborative.org

References

ACS Accreditation Committee. (2014). *Application guidelines – Professional level courses.* Sydney, Australia: Australian Computer Society Professional Standards Board.

B20 Australia. (2014). *B20 Human Capital Taskforce Policy Summary*. Sydney, Australia: B20 Australia.

Billingsley, W. (2019, February). *A technologist's agenda for scriptable, smart, social, and republishable courses*. Paper presented at the SPLICE Spring 2019 Workshop in Minneapolis, MN. Retrieved from https://cssplice.github.io/SIGCSE19/Billingsley.pdf

Billingsley, W. (2019, May). The case of the fragmented classroom. In ICSE-SEET '19: *Proceedings of the 41st International Conference on Software Engineering: Software Engineering Education and Training* (pp. 74-83). Piscataway, NJ: IEEE Press.

Billingsley, W., & Fletcher, P. R. (in press). Employing authentic analytics for more authentic tasks. In T. Prodromou (Ed.), *Big data in education: Pedagogy and research*.

Billingsley, W., Ngu, B., Phan, H., Gromik, N., & Kwan, P. (2016). Using a video-based critique process to support studio pedagogies in distance education – A tool and pilot study. In S. Barker, S. Dawson, A. Pardo, & C. Colvin (Eds.), *Show Me The Learning: Proceedings ASCILITE 2016 Adelaide* (pp. 43–48). Tugun, Australia: ASCILITE

Billingsley, W., & Steel, J. (2013). A comparison of two iterations of a software studio course based on continuous integration. In ITiCSE '13: *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education* (pp. 213–218). New York, NY: ACM. https://doi.org/10.1145/2462476.2465592

Billingsley, W., & Steel, J. R. (2014). Towards a supercollaborative software engineering MOOC. In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 283–286). New York, NY: ACM. https://doi.org/10.1145/2591062.2591157

Billingsley, W., Torbay, R., Fletcher, P. R, Thomas, R. N., Steel, J. R. H., & Süß, J. G. (2019). Taking a studio course in distributed software engineering from a large local cohort to a small global cohort. *ACM Transactions on Computing Education*, *19*(2), 1–27. https://doi.org/10.1145/3218284

Bleakley, A. (1999). From reflective practice to holistic reflexivity. *Studies in Higher Education*, *24*, 315–330. https://doi.org/10.1080/03075079912331379925

Brooks, F. P., Jr. (1987). No silver bullet: Essence and accidents of software engineering. *IEEE Computer*, 20(4), 10–19. https://doi.org/10.1109/MC.1987.1663532

Brown, E., & Glover, C. (2006). Evaluating written feedback. In C. Bryan & K. Clegg (Eds.), *Innovative assessment in higher education* (pp. 81–91). Abingdon, UK: Routledge.

Bull, C. N., Whittle, J., & Cruickshank, L. (2013). Studios in software engineering education: Towards an evaluable model. In *Proceedings of the 35th International Conference on Software Engineering* (pp. 1063–1072). Piscataway, NJ: IEEE Press.

Bull, C. N., & Whittle, J. (2014). Supporting reflective practice in software engineering education through a studio-based approach. *IEEE Software*, *31*(4), 44–50. https://doi.org/10.1109/MS.2014.52

Carter, A. S., & Hundhausen, C. D. (2011). A review of studio-based learning in computer science. *Journal of Computing Sciences in Colleges*, *27*(1), 105–111.

Cooper, A. (1998). *The inmates are running the asylum: Why high tech products drive us crazy and how to restore the sanity*. Indianapolis, IN: Macmillan.

Cooper, A., Reimann, R., Cronin, & Noessel, C. (2014). *About face: The essentials of interaction design* (4th ed.). New York, NY: Wiley.

Dacre Pool, L., & Sewell, P. (2007). The key to employability: Developing a practical model of graduate employability. *Education + Training*, *49*, 277–289. https://doi.org/10.1108/00400910710754435

Decastro, M. (2014, 8 December). What type of public speaker or presenter are you really? Retrieved from https://mindfulpresenter.com/mindful-blog/public-speaker-presenter/

Dewey, J. (1938). *Experience and education*. New York, NY: Simon and Schuster.

Docherty, M., Sutton, P., Brereton, M., & Kaplan, S. (2001). An innovative design and studio-based CS degree. *ACM SIGCSE Bulletin*, *33*(1), 233–237. https://doi.org/10.1145/366413.364591

Engstrom, C., Hay, P., Macdonald, D., Brukner, P., & Khan, K. (2011). *An integrated system for online clinical assessment of practical skills (eCAPS) for web-based courses* (PP8-893 Final Report). Sydney, Australia: Office for Learning and Teaching.

Erickson, F. (2006). Definition and analysis of data from videotape: Some research procedures and their rationales. In J. L. Green, G. Camilli, & P. B. Elmore (Eds.), *Handbook of complementary methods in education research* (pp. 177–192). Mahwah, NJ: Erlbaum.

Feloni, R. (2014, 5 December). How to be a better public speaker based on your personality type. Retrieved from https://www.businessinsider.com.au/public-speaking-tips-by-personality-type-2014-12

Gielen, M., & De Wever, B. (2012). Peer assessment in a wiki: Product improvement, students' learning and perception regarding peer feedback. *Procedia - Social and Behavioral Sciences*, *69*(2012), 585–594. https://doi.org/10.1016/j.sbspro.2012.11.450

Hands, B., Coffey, A., Clark-burg, K., Das, A., Gerrard, P., Hackett, C., Jenkins, S., MacNish, J., Miller, D., Rycroft, P., van Staden, C., Gilomen, K., & Flyn, B. (2009). *Interactive video analysis to develop learning and assessment of university students' practical and communication skills* (CG7-385 Final Report), Strawberry Hills, Australia: Australian Learning and Teaching Council.

Hazzan, O. (2002). The reflective practitioner perspective in software engineering education. *Journal of Systems and Software, 63*(3), 161–171. https://doi.org/10.1016/S0164-1212(02)00012-2

Hendrix, D., Myneni, L., Narayanan, H., & Ross, M. (2010). Implementing studio-based learning in CS2. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 505–509). New York, NY: ACM. https://doi.org/10.1145/1734263.1734433

Hundhausen, C. D., Narayanan, N. H., & Crosby, M. E. (2008). Exploring studio-based instructional models for computing education. *ACM SIGCSE Bulletin*, *40*(1), 392–396. https://doi.org/10.1145/1352322.1352271

Ithaca Group (2013). *Core Skills for Work Developmental Framework*. Canberra, Australia: Departments of Industry, Innovation, Climate Change, Science, Research and Tertiary Education; and Education, Employment and Workplace Relations.

Jollands, M., Clarke, B., Grando, D., Hamilton, M., Smith, J., Xenos, S., Brodie, M., & Pocknee, C. (2015). *Developing graduate employability through partnerships with industry and professional associations* (SP13-3256 Final Report). Sydney, Australia: Office for Learning and Teaching.

Jordan, L. (2012). Bringing video into the mainstream: Recommendations for enhancing peer feedback and reflection. *Research In Learning Technology*, *20*(Suppl. 1). https://doi.org/10.3402/rlt.v20i0.19192

Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice-Hall.

Kuhn, S. (2001). Learning from the architecture studio: Implications for project-based pedagogy. *International Journal of Engineering Education*, *17*, 349–352.

Leitch, R., & Day, C. (2000). Action research and reflective practice: Towards a holistic view, *Educational Action Research*, *8*, 179–193. https://doi.org/10.1080/09650790000200108

Levy, A. (1980). Total studio. *Journal of Architectural Education*, *34*(2), 29–32. https://doi.org/10.1080/10464883.1980.10758247

Little, B. (2006). *Learning and employability series 1: Employability and work-based learning*. Heslington, UK: The Higher Education Academy.

Long, J. G. (2012). State of the studio: Revisiting the potential of studio pedagogy in U.S.-based planning programs. *Journal of Planning Education and Research*, *32*, 431–448. https://doi.org/10.1177/0739456X12457685

Matthews, B. (2013). *Evaluating vicarious critiques as a tool for open online studio-based design education* (Unpublished grant application). The University of Queensland, Australia.

McMahon, G. (2012, 9 March). The 6 presenter types: Which one are you? Retrieved from https://makeapowerfulpoint.wordpress.com/2012/03/09/the-6-presenter-types-which-one-are-you-2/

Moreland, N. (2006). *Learning and employability series 1: Entrepreneurship and higher education: An employability perspective*. Heslington, UK: The Higher Education Academy.

Nurkkala, T., & Brandle, S. (2011). Software studio: Teaching professional software engineering. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 153–158). New York, NY: ACM. https://doi.org/10.1145/1953163.1953209

Oliver, B. (2010). *Benchmarking partnerships for graduate employability* (Teaching fellowship final report). Perth, Australia: Curtin University. Retrieved from https://espace.curtin.edu.au/handle/20.500.11937/31598.

Oliver, B., & Whelan, B. (2011). *Building course team capacity to enhance graduate employability* (CG8-735 Final Report). Sydney, Australia: Office for Learning and Teaching.

Patrick, C.-j., Peach, D., & Pocknee, C. (with Webb, F., Fletcher, M., & Pretto, G.) (2008). *The WIL [work-integrated learning] report: A national scoping study* (Australian Learning and Teaching Council final report). Brisbane, Australia: Queensland University of Technology.

Reardon, S., & Tangney, B. (2015). Smartphones, studio-based learning, and scaffolding: Helping novices learn to program. *ACM Transactions on Computing Education*, *14*(4), 1–15.

Ronaghi, F., Saberi, A., & Trumbore, A. (2015). NovoEd, a social learning environment. In P. Kim (Ed.), *Massive open online courses: The MOOC revolution* (pp. 96–105). New York, NY: Routledge.

Saghafi, M. R., Franz, J., & Crowther, P. (2012). A holistic blended design studio model: A basis for exploring and expanding learning opportunities. In *Society for Information Technology and Teacher Education International Conference (SITE 2012)*. Austin, TX: EdITLib.

Schön, D. A. (1983). *The reflective practitioner: How professionals think in action*. New York, NY: Basic Books.

Schön, D. A. (1987). *Educating the reflective practitioner: Toward a new design for teaching and learning in the professions*. San Francisco, CA: Jossey-Bass.

Seely Brown, J., & Adler, R. P. (2008). Minds on fire: Open education, the long tail, and learning 2.0. *EDUCAUSE Review*, *43*(1), 16–32.

Süß, J. G., & Billingsley, W. (2012). Using continuous integration of code and content to teach software engineering with limited resources. In *Software Engineering (ICSE), 2012 34th International Conference on* (pp. 1175–1184). Piscataway, NJ: IEEE Press. https://doi.org/10.1109/ICSE.2012.6227025

Tomayko, J. E. (1991). Teaching software development in a studio environment. In *Proceedings of the twenty-second SIGCSE technical symposium on Computer science education* (pp. 300–303). New York, NY: ACM. https://doi.org/10.1145/107005.107070

Tomayko, J. E. (1996). Carnegie Mellon's software development studio: A five year retrospective. In *Proceedings of 9th Conference on Software Engineering Education* (pp. 119–129). Piscataway, NJ: IEEE Press. https://doi.org/10.1109/CSEE.1996.491367

Tseng, S.-C., & Tsai, C.-C. (2007). On-line peer assessment and the role of the peer feedback: A study of high school computer course. *Computers & Education*, *49*, 1161–1174. https://doi.org/10.1016/j.compedu.2006.01.007

van Manen, M. (1995). On the epistemology of reflective practice. *Teachers and Teaching: Theory and Practice*, *1*, 33–50. https://doi.org/10.1080/1354060950010104

Appendix A

Certification by Deputy Vice-Chancellor (or equivalent)

I certify that all parts of the final report for this OLT grant provide an accurate representation of the implementation, impact and findings of the project, and that the report is of publishable quality.

Name:  Date: 31 July 2018

Appendix B Coding Scheme for Presentation Videos

SE GROUP DIVISION OF LABOUR
Single presenter (0–1)
Multiple presenters – successive (0–1)
Multiple presenters – simultaneous (0–1)
ROLE
Persona
Type
Genre
Narrate (P, S, T)
Factualise (P, S, T)
Engage (P, S, T)
SE FOCUS ELEMENTS (‘The Bricks’)
Visual (L1–H3)
Mechanics (L1–H3)
Code (L1–H3)
User Story (L1–H3)
ED FOCUS ELEMENTS (‘The Bricks’)
Goal (L1–H3)
Scaffold (L1–H3)
Mechanics (L1–H3)

Utility (L1–H3)
ORGANISATIONAL LOGIC ('The Mortar')
Overall Content
Signalling (L1 cue-less – H3 cues added)
Segmenting (L1 continuous – H3 bite size units)
Multimodal Elements: AUDIO, GRAPHICS, WORDS
Coherence (L1 non-essential – H3 essential)
Spatial contiguity (L1 far – H3 near)
Temporal contiguity (L1 successive – H3 simultaneous)
Redundancy (L1 duplicative – H3 supplemental)
PRESENTATION STYLE ('The Tools')
Social Signal
Gesture and posture (0–2)
Face and eye behaviour (0–2)
Vocal behaviour (0–2)
Media Used
Camera (0–1)
Screencast
slide-based (0–1)
screen recording (0–1)
Audience Looking at
PC Screen (0–1)

Mobile (0–1)
Paper (0–1)
Poster (0–1)
Wall/Whiteboard (0–1)
Person (0–1)
Person in context (0–1)
Context alone (0–1)
Props Used
Person (0–1)
Whiteboard/Screen (0–1)
Code (0–1)
Text (0–1)
Diagram/s (0–1)
Image/s (0–1)
Object/s (0–1)
Pen (0–1)
Video (0–1)
Prop Interactivity
Talk (0–1)
Physical Point (0–1)
Paper Shuffle (0–1)
Swipe – Slide – Scroll with Finger (0–1)
Draw (0–1)
Mouse Scroll (0–1)

Mouse Point (0–1)
Zoom/Pan/Appear (0–1)
Slides Progress (0–1)
Slide/s Static (0–1)
Simulation (0–1)
Video playing on screen (0–1)
APPEAL FOR FEEDBACK (0–1)
Structure
L1 vague – H3 explicit
Undefined (0–1)
Type (P, S, T)
Correctness: Is it right/wrong?
Confirmatory: 'Am I on the right track?
Instruction seeking: 'How to …? Better way …?
Incompleteness: 'suggestion/s to further develop …?
Undefined (0–1)

Appendix C Coding Scheme for Critique Videos

ROLE
Persona
Type
Genre

Narrate (P, S, T)
Factualise (P, S, T)
Engage (P, S, T)
SE FOCUS ELEMENTS ('The Bricks')
Visual (L1–H3)
Mechanics (L1–H3)
Code (L1–H3)
User Story (L1–H3)
Presentation (L1–H3)
ED FOCUS ELEMENTS ('The Bricks')
Goal (L1–H3)
Scaffold (L1–H3)
Mechanics (L1–H3)
Utility (L1–H3)
Presentation (L1–H3)
ORGANISATIONAL LOGIC ('The Mortar')
Overall Content
Signalling (L1 cue-less – H3 cues added)
Segmenting (L1 continuous – H3 bite size units)
Multimodal Elements: audio, graphs, words
Coherence (L1 non-essential – H3 essential)
Spatial contiguity (L1 far – H3 near)
Temporal contiguity (L1 successive – H3 simultaneous)

Redundancy (L1 duplicative – H3 supplemental)
PRESENTATION STYLE ('The Tools')
Social Signal
Gesture and posture (0–2)
Face and eye behaviour (0–2)
Vocal behaviour (0–2)
Media Used
Camera (0–1)
Screencast
slide-based (0–1)
screen recording (0–1)
Audience Looking at
PC Screen (0–1)
Mobile (0–1)
Paper (0–1)
Wall/Whiteboard (0–1)
Person (0–1)
Person in context (0–1)
Context alone (0–1)
Props Used
Person (0–1)
Whiteboard/Screen (0–1)
Code (0–1)

Text (0–1)
Diagram/s (0–1)
Image/s (0–1)
Object/s (0–1)
Pen (0–1)
Video (0–1)
Prop Interactivity
Talk (0–1)
Physical Point (0–1)
Paper Shuffle (0–1)
Swipe – Slide – Scroll with Finger (0–1)
Draw (0–1)
Mouse Scroll (0–1)
Mouse Point (0–1)
Zoom/Pan/Appear (0–1)
Slides Progress (0–1)
Slide/s Static (0–1)
Simulation (0–1)
Video playing on screen (0–1)
RESPONSE TO FEEDBACK REQUEST (0–1)
Structure
L1 vague – H3 specific
Undefined (0–1)

FEEDBACK STYLE
Corrective Feedback
Predominance (P-S-T)
Strength (L1 muted – H3 salient)
Reinforcing Feedback
Predominance (P-S-T)
Strength (L1 muted – H3 salient)
Didactic Feedback
Predominance (P-S-T)
Strength (L1 muted – H3 salient)
Suggestive Feedback
Predominance (P-S-T)
Strength (L1 muted – H3 salient)
QUALITY OF PEER FEEDBACK
Tone Equilibrium
Negative (1) – Positive (3)
Motivational (0–1)
– Evaluative judgement (0–1)
– Descriptive (0 = No) Specificity (L1–H3)
De-Motivational (0–1)

Suggestions: No (0) – Yes (1)
Concrete (Useful L1 – L3)
Abstract (Useful L1 – L3)