A thesis submitted for the degree of Doctor of Philosophy

Studios in Software Engineering Education

Christopher Neil Bull, *B.Sc, M.Res*

December 2015

HighWire CDT
School of Computing and Communications
Lancaster University

"An important problem with the conventional teaching method is that it favors problem solving over conceptual understanding. As a result, many students memorize 'problem solving strategies' [...] what good is it to teach just the mechanical manipulation of equations without achieving understanding?"

– Eric Mazur

"I hear and I forget.
I see and I remember.
I do and I understand."

– Confucius

Abstract

Computing has its roots in mathematics, where lectures are the dominant mode of education. Software engineering (SE) education, born from computer science, is also traditionally taught using lectures, but has grown beyond its mathematical roots; as the name implies, it is an engineering discipline. It is arguably necessary for SE to rethink its approach to education.

Studio education is one alternative being explored. Studios originated from architecture and design, and are complex spaces used by collocated students to collaboratively and individually work on projects; they emphasise a physical "home" for students, problem-based and peer-based learning, and mentoring by academic staff rather than formal lectures. There are inherent similarities between SE and the original studio disciplines: e.g. we often use the architecture of buildings as metaphors when designing and describing software. This suggests that studios in SE should be further explored, despite its apparent lack of uptake across institutions worldwide.

This thesis aims to provide useful information for anyone considering utilizing a studio-based approach. Initially, with no widely accepted definition for studio education available, a series of interviews with design/architecture studio educators was conducted, culminating in an understanding in the form of the 'studio framework'. This is followed by further interviews, with SE studio educators, to determine their perspective of studio education, and exploring the SE specific elements to studio education. Finally, experiences and observations are shared of Lancaster University's recent SE studio, comparing it to the studio framework.

Publications

Some of the work presented in this thesis has previously been published:

▷ Christopher N. Bull, Jon Whittle and Leon Cruickshank (2013). 'Studios in Software Engineering Education: Towards an Evaluable Model'. In: *Proceedings of the 35th International Conference on Software Engineering (ICSE '13)*. San Francisco, CA, USA: IEEE Press, pp. 1063–1072

▷ Christopher N. Bull and Jon Whittle (2014a). 'Observations of a software engineering studio: Reflecting with the studio framework'. In: *Proceedings of the 27th Conference on Software Engineering Education and Training (CSEE&T'14)*. Klagenfurt, Austria: IEEE, pp. 74–83. DOI: 10.1109/CSEET.2014.6816784. *[Best Paper Award]*

▷ Christopher N. Bull and Jon Whittle (2014b). 'Supporting Reflective Practice in Software Engineering Education through a Studio-Based Approach'. In: *IEEE Software* 31.4, pp. 44–50. DOI: 10.1109/MS.2014.52

▷ Jon Whittle, Christopher N. Bull, Jaejoon Lee and Gerald Kotonya (2014). 'Teaching in a Software Design Studio: Implications for Modeling Education'. In: *Proceedings of the 10th Educators' Symposium (EduSymp '14) at MODELS 2014*. Valencia, Spain: CEUR-WS

▷ Jaejoon Lee, Gerald Kotonya, Jon Whittle and Christopher Bull (2015). 'Software Design Studio: A Practical Example'. In: *Proceedings of the 37th International Conference on Software Engineering (ICSE '15) [in press]*. Florence, Italy: IEEE

Acknowledgements

I would like to thank my supervisors, Prof. Jon Whittle and Prof. Leon Cruickshank, for their assistance and dedicated involvement throughout this process. Thank you for your guidance, support, and understanding over the years, but also for showing me what it is to be an academic. Also, thanks to HighWire CDT and its Directors for providing me with the opportunity to do a PhD in the first place, especially Prof. Gordon Blair, who has been a support to us all throughout.

I would also like to show gratitude to all of the anonymous interview participants and Lancaster University's software engineering students for participating in this research. Thank you for taking time out of your busy schedules to have conversations with me – I intend to take this thread of research further, alongside others I have experienced along the way, because if nothing else, your passions for changing how students learn is infectious.

I would like to acknowledge all of the staff in the School of Computing and Communications, my fellow HighWire PhD students, and my family, for the countless discussions and critiques. This process of support and critique is something that I have seen embodied in studio education as well as the PhD process during this research, and I feel it is essential to our success as researchers. I particularly want to mention a few key people, though I can't name you all, who have helped in precisely this process: Stephen Forshaw, John Hardy, Graham Dean, Marcia Smith, Barney Craggs, Kiel Long, John Vidler, Carl Ellis, and Stephen Wattam.

. . . also coffee. The perpetual trade-off between productivity and sleep.

Finally, and most importantly, I would like to thank my wife Jennifer for her love and constant support. I have found out that a PhD, on the contrary to what we were told at the start, is not a solo pursuit – indeed, by making the decision to take on a PhD, you have implicitly (whether you realise or not) decided to take those around you on that process as well.

Jen: You have shared in my ups and downs. Thank you for those times you had to listen to me ramble on, for proofreading after an all-nighter, at times nodding along even though I am filling the air with buzzwords, and frequently making me coffees. You don't have to do these things, but you do. For that, I thank you, and I love you.

This work was undertaken as part of the post-disciplinary HighWire programme at Lancaster University. HighWire is funded by the Digital Economy Programme: a Research Councils UK cross council initiative led by EPSRC and contributed to by AHRC, ESRC and MRC.

Declaration

I declare that this thesis is my own work and that it has not been submitted in substantially the same form for the award of a higher degree elsewhere.

Christopher N. Bull

28th December 2015

Contents

List of Figures

List of Tables

1 Introduction

There are many approaches to learning that can be utilised in software engineering, ranging from traditional structured approaches (such as lectures and homework) to more agile ones (such as interactive seminars and project-based learning). A traditional lecture is an effective method for presenting information to numerous students at once (Bligh, 1998, p.3). However, in itself it does not necessarily promote understanding of the subject content, because it is argued that the *"traditional approach to teaching reduces education to a transfer of information"* (Mazur, 2009, p.50). This follows a long held idea by some that lectures are not necessarily an optimal approach to education (Gibbs, 1981). Project-based courses are one alternative that focuses on *"teaching by engaging students in investigation"* (Blumenfeld et al., 1991, p.371); these courses increase motivation and can better support a learning-by-doing approach, especially by allowing students to *"apply system-level thinking"* (Jazayeri, 2004, p.xxv). Traditional lecture-based education is prominent in the computing discipline, and sometimes supported with project-based work, which is echoed by others, stating that *"Instruction continues to be conducted in the traditions established when computer science emerged from mathematics decades ago"* (Hundhausen et al., 2008a, p.392).

An approach that looks to develop complex interrelationships could possibly better prepare students for real-world systems development – especially reflecting on non-technical skills such as communication and the ability to work in a team (Jazayeri, 2004). Brooks (1995, p.4) recalled memories where *"two programmers in a remodelled garage have built an important program that surpasses the best*

efforts of large teams" and then posed the question *"why then have not all industrial programming teams been replaced by dedicated garage duos?"* Here, two teams achieve different results, not too dissimilar to what could be expected of the differing learning approaches. What is it that this duo has, that a large company therefore does not? Students do not generally, if at all, work on large complex systems during their education, so institutions could get away with using traditional teaching methods (if students did not find them boring). Software engineering is an inherently creative process (Brooks, 1995; Glass and DeMarco, 2006) and it is very beneficial to learn with other students (Falchikov, 2001). To support this, we should ask: are the environments that we learn in suited to promoting a collaborative culture that is sensitive to creative processes?

1.1 Studio Education

Looking towards the design disciplines (design, art and architecture etc.) that face similar challenges with learning and team-based work, there are aspects of their commonly used working environments which, from the outset, promote a range of advantages (Kuhn, 1998), some of which include collaboration and creativity (Fällman, 2007). These environments are called studios, an alternative to lecture-oriented courses, studios are open or shared working environments where students do visually-centred work. Studios are a very prominent approach in design and arts education (Lynch et al., 2002). A more in-depth look at studios informs us that they promote several human-centric aspects of learning including: 'reflective practice' (Schön, 1987) (a process of observing and refining practice in a continuous cycle); problem-based learning (Morkel, 2011); coaching, peer-learning, and mentoring by academic staff rather than formal lectures (Schön, 1987; Hokanson, 2012); and an emphasis on a physical *"home"* for the students (Carlhian, 1979, p.8). Some examples of studios are shown in Figures 1.1-1.4.

This thesis takes an inclusive approach to researching studio education with the intention of exploring the core concepts of studio education. Studios can vary

Figure 1.1: An example professional architecture studio – Alvar Aalto's studio: 'Studio Aalto' © Pepechibiryu, CC BY-SA 3.0

Figure 1.2: An example architecture studio – A student's workspace at the University of Oregon: 'Architecture Studio - My new home' © Erik Bishoff, CC BY-NC-ND 2.0

Figure 1.3: An example architecture studio: 'Desks of architecture students in the Yale Art and Architecture Building' © Ragesoss, CC BY-SA 3.0

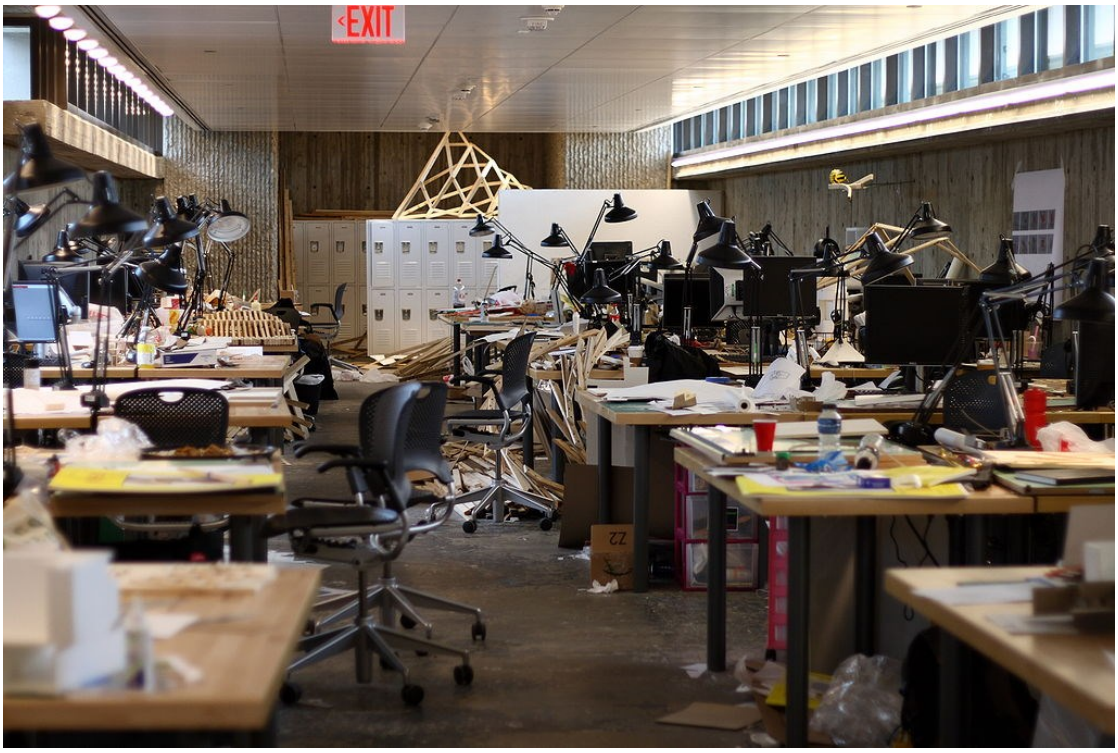


Figure 1.4: An example service design studio at the Royal College of Art: 'RCA Service Design studio' © Ben Terrett, CC BY-NC-ND 2.0

in implementation, with some studio courses being entirely studio based, whilst others may be a studio module within a course. Studios courses may even still utilise lectures, but they are not the primary approach to learning. They may even be shared across cohorts of students, or even dedicated to one group of students. Sometimes this may be forced through the institutions that provide the space for the courses.

Previous research has linked the type of activities and problems faced by architecture students (building, not software) with those of software engineering students (Hazzan, 2002). There have been some attempts to transfer studio-based teaching to software engineering education (software studios), with numerous reporting successes; software studios are courses that utilise a studio approach to teaching software engineering and development – these include software engineering, computer science, and information technology. In many ways, this is natural as software engineering has significant practical elements. However, attempts at software studios have usually ignored experiences and theory from arts and design studio teaching. There is therefore a lack of understanding of what "studio" really means, and how well the concepts transfer to software engineering. It is this gap in knowledge that this thesis will address.

Recent research continues to recognise the need to improve our learning environments (Herrick, 2011) – regardless of whether new environments are studio-based or not, the trend is shifting towards more flexible open spaces which promote group work. The evolving spaces that are required blend *"instruction, course work and social interactions"* (Herrick, 2011, p.65).

1.2 Background Summary

Interpretations of studio education for software disciplines (computer science and software engineering etc.) have been explored for nearly 25 years, with the earliest known software studio implemented at Carnegie Mellon University (CMU) in 1990 (Tomayko, 1991); this institution offers a 'Master of Software Engineering' program

with a studio experience. Software studios are reviewed in detail in chapter 2, Literature Review.

Despite continued interest in studio-based learning (Carter and Hundhausen, 2011), there appears still to be very few implementations of software studios, yet potential benefits of studio education are numerous and understood (Hazzan, 2002; Hundhausen et al., 2008a).

If publications are taken as example implementations of studio-based education, then besides the limited number in existence, they also significantly vary in execution and guiding principles; this in itself is not necessarily a negative point, as each will have their own successful elements. However, this makes it difficult to compare or benefit from shared implementation details. Despite the limited numbers of studios, there is still continued interest; for example Carter and Hundhausen (2011) recently presented a brief overview of five studios in computer science. The studios they analysed did not replace lectures, but were instead used to reinforce content taught in lectures – this is not necessarily the case in all studios in design or architecture, with some rarely using lectures. The success or failure of software studios are also *"usually assessed through end-of-term questionnaires"* (Carter and Hundhausen, 2011, p.110), which may provide limited information on their own.

1.3 Motivation

Despite the apparent success of the limited number of documented studios, it is also apparent that none of them follow a shared definition or understanding of studio education, possibly due to the possible variability in studio education – making comparisons difficult. Some of the documented studio implementations point towards literature that only gives broad descriptions or a vague understanding, and a small number relied on the tacit knowledge of someone who originated from the disciplines that gave us studios – a consultant of sorts. Environments that were claimed to be studios may have been effective, but were they truly studios? Do they offer what studios in other disciplines offer? It is difficult to succinctly define

studio practice due to the inherently complex nature of studio education.

Some people argue that lectures are no longer the desired medium for education, with Mazur (2009, p.50) succinctly summarising this viewpoint: *"The traditional approach to teaching reduces education to a transfer of information [...] education is so much more than information transfer"*. Carpenter et al. (2013) state that the fluency of an instructor in a lecture only increases perceptions of learning, and not the actual learning. There are other benefits to a fluent instructor, for example enjoyment level, but this study suggests that the perception of learning is decoupled to actual learning – suggesting that exploring different teaching approaches are a better avenue to follow.

Despite the lack of a concrete understanding of studio education, software studio implementations often report varying levels of success; it has been indicated that, in certain studios, the student's content mastery is *"as much or more than students in the same courses taught in the traditional way"* and also that *"students in studio courses have shown higher levels of motivation and engagement than students in traditional courses"* (Narayanan et al., 2012, p.165).

Studio-based learning for software engineering is a well-received concept, despite its apparent lack of uptake across institutions worldwide. Apart from the benefits a studio may provide, one of the reasons to explore software studios further is because students enjoy them (Carter and Hundhausen, 2011).

1.4 The Research

The research objectives of this thesis are to investigate studio-based education in software engineering, with the intention of making studios more accessible to others who wish to implement one. Upon embarking on research into studio-based education for software engineering, it became apparent, as mentioned earlier, that a shared, clear, and agreed upon definition of what a studio is did not exist – it also became apparent that an understanding of their complex nature resides in tacit knowledge. The lack of a definition is explored in detail in the Literature

Review chapter, section 2.2.1. This lack of understanding is the foundation for the following research questions explored throughout this thesis:

RQ1) What is a studio?

RQ2) What is the current state of studio education in software engineering?

RQ3) Does studio education transfer to software engineering education?

Firstly an understanding of studio education is required to be formulated. This is explored by RQ1 in chapter 3 by interviewing studio educators from the disciplines where studio education originated from, and subsequently performing an analysis of the interview transcripts to discover what people native to studio education think it consists of. This culminated in the creation of the studio framework – providing an understanding of studio education.

To complement RQ1, RQ2 is explored in chapter 2 (Literature Review) and chapter 4 (State of Software Studios in Universities). This thread of research is conducted to get a sense of the software studios that are in existence, and then perform interviews with software engineering educators using a similar methodology as chapter 3. By following a similar methodology, but with participants from software engineering as opposed to the design disciplines, allows for the results of the analyses to be comparable. This is discussed further in chapter 4's 'Methodology' section (4.1).

Finally, RQ3 is explored in chapter 6 (Analysis of Lancaster University's Studio). Observations are presented of a new software engineering studio course at Lancaster University, sharing practical examples of studio activities. These observations are complemented by a focus group with some of the students from the studio course, and then correlated with the studio framework to determine if Lancaster's software engineering studio is indeed a studio.

1.5 Summary of Results and Contributions

Whilst exploring RQ1 in chapter 3, the 'studio framework' was created from the interview transcripts analysis. This framework provides a definition and understanding of studio education.

Further to the studio framework, chapter 4 led to the creation of a studio framework variant, the 'software studio framework'. As the interviews and analysis (with software engineering educators) in chapter 4 followed a similar methodology as chapter 3 (with design, architecture and art educators), the results were comparable. The analysis of the software engineering educators interviews resulted in a framework with mostly the same set of categories, except for one, and the parameters within each category had variable overlap between the two studies; some were more strongly linked than others. This means that the software studio framework was somewhat similar to the original studio framework. The differences between the two frameworks is more nuanced, and exist inside the categories, e.g. the parameters.

The final contribution to this thesis is the analysis of an observational study performed on Lancaster University's own software engineering studio. A summary of Lancaster's software engineering studio is given in chapter 5, including a description of the physical space, curriculum, motivation for its creation, and guiding principles provided to teaching staff. This observational study, chapter 6, analysed observations and correlated its findings with the studio framework, to determine how well Lancaster's software engineering studio conformed to the framework and see if in this example whether studio education can be transferred to software engineering. It is concluded that Lancaster's software engineering studio is a studio, as it supports each of the studio framework's categories; this therefore answers RQ3 as it informs us that studio education is indeed capable of transferring to software engineering education.

2 Literature Review

The objective of this chapter is to provide an understanding of the current state of studio education, supplying an adequate foundation for the research questions in this thesis to be answered. The first aim is to collate information about the origins of studio education in design and art disciplines, followed by any prominent definitions or features that are widely expressed in studio literature. The second aim is to explore literature on current implementations of software studios, and to gather information on what is considered to form these studios. Software studios are courses that utilise a studio approach to teaching software engineering and development; these include software engineering, computer science, and information technology.

2.1 Roots Of Studio Education

The roots of the design studio can be traced back as far as Medieval Times where guilds would take on apprentices (Schön, 1985). Most literature that discusses the origins of the design studio often describes two main schools of studio education: École des Beaux-Arts and the Bauhaus (Broadfoot and Bennett, 2003; Salama, 1995). As such, this section briefly summarises them and their main contributions to design studios, followed by a section on more recent literature regarding studio education.

2.1.1 École des Beaux-Arts

École des Beaux-Arts (France, 1819-1914), or 'School of Fine Arts', was a French school of art which taught subjects such as drawing, painting, sculpture, engraving and architecture. The environments used were known as ateliers, French for 'workshop', where students worked under the guidance of a patron (tutor) – these are widely regarded as the first knowns environments that resembled a studio, as more recent studios are said to *"have inherited the historical tradition of the École des Beaux-Arts and its atelier model"* (Oh et al., 2013, p.303). The purpose of École des Beaux-Arts was to allow working and learning to occur simultaneously. Whilst working under their patron, the students would also be studying a theory of design; each atelier had a different *"acquired reputation"* that would lure different students depending on their taste or intended direction (Carlhian, 1979, p.7). One of the most important factors of the atelier was that the *"design problem and learning by doing superseded the lecture as the primary method of teaching"* (Broadfoot and Bennett, 2003, p.1). Many traditions in studios today evolved from the École which *"provided the basis of a pedagogical method that is still the core of design and architectural education"* (Broadfoot and Bennett, 2003, p.1). Some aspects include:

- Esquisse (initial sketch to a problem that would be further developed)
- Teaching of design by practising professionals
- The final evaluation of student work by a jury

Another aspect that occurred at the École, although not essential to the studio environment, was the tradition where 'anciens' (older students) helped younger students. There were also monthly architectural competitions (*'Concours Mensuels d'Emulation'*), which the students were expected to enter regularly. The reason the École closed in 1914 was because its *"buildings were reserved as a military hospital"* (Salama, 1995, p.49).

2.1.2 Bauhaus

The Staatliches Bauhaus, or School of Building (Germany, 1919-1933), was founded by Walter Gropius. Bauhaus is well known for its fusion of theory and practice, but more importantly, art and craft; it taught these new approaches in a workshop environment that formed the basis of today's studios. Some of the topics taught at this school were art, architecture, graphic design, interior design, industrial design, and typography. An important way that Bauhaus differs from École des Beaux-Arts is that Bauhaus' founder, Walter Gropius, placed an emphasis on production and technology: *"design was neither an intellectual nor a material affair but simply an integral part of modern concepts of mass production and modern technology"* (Lackney, 1999, p.2).

Although this school was *"a challenge to the Beaux Arts tradition of education, the basic form of studio-based learning model remained unchanged"* (Lackney, 1999, p.3). Instruction was *"of a practical nature, providing actual work with materials in the shops and on buildings under construction"* (Lackney, 1999, p.2). One of the major aims was *"the unification of art and modern technology to create architecture and design for the modern free man and woman"* (Ehn, 1998, p.208), with another aim being to reinstate control to the designer over architectural decisions (Salama, 1995). What made Bauhaus particularly unique was that *"apprentices were to be instructed, not only by 'masters' of each particular craft, but also by fine artists"* (Salama, 1995, p.50). One final, yet important, aspect was that the teaching program aimed to develop the personalities of the students (Broadfoot and Bennett, 2003), as this was considered key to the future design experiences.

The effects and influences of Bauhaus were widespread, becoming *"an abbreviation for the radical modernization of life and its positive and negative side-effects"* (Droste, 2002, p.6); furthermore, it became more than just a school, it became a concept which also had an associated catchphrase: 'Bauhaus-style'. It is often claimed that *"architectural education as we have it today owes more to the Bauhaus than to any other single institution"* (Dearstyne, 1962, p.13).

2.2 Studios

A studio is an alternative to lecture-oriented courses, which are a dominant teaching method at universities. It is often a room that supports and promotes several human-centric aspects of learning, and is also sometimes called an 'atelier' (which is the term used at the École des Beaux-Arts mentioned earlier). Studio-based education often includes a dedicated physical space for students where learning comes from tackling complex, partially specified problems, and teaching staff are seen more as mentors and coaches rather than disseminators of information. Learning can often be collaborative, is very practical, often project-based, employs coaching, formal and informal critique, and the studio is commonly said to act as the student's second home: *"Students do not so much attend these events as live in them"* (Schön, 1987, p.311). Conceptually, a studio *"is a process of learning by doing"* (Lawson, 2012, p.7). Students also frequently play roles to their peers (peer-learning or peer-coaching) in addition to what the staff would also normally perform, e.g. coaching and critiquing. A key difference to traditional lecture-based teaching is that lectures are rarely used, if at all, and when they are the students are in their studio before and after the lectures – which is similar in approach to the École des Beaux-Arts (see section 2.1.1, above), which had some lectures, but were not the primary method of teaching.

Studios can be quite variable, as they could be a studio module within a larger degree, or the studio may be their studio space for everything in a degree. Studio spaces can sometimes be dedicated to cohorts of students, and others may share a studio space with everyone on a course. Indeed, some studios may share their space across courses. Due to this variability, this thesis takes an inclusive look at studio education. Schön (1987, p.37) describes studios as a reflective practicum:

> *"A practicum is a setting designed for the task of learning a practice. In a context that approximates a practice world, students learn by doing, although their doing usually falls short of real-world work."*

Schön is a popular name in studio literature, with many referring to his work on reflective practice, particularly in studio settings. He also points out the similarities between the relationship between studio teachers ("masters") and their students, which is akin to a master and apprentice's relationship.

This section reviews studios from the perspectives of architecture, design and art, allowing for this thesis to position itself and its contributions with respect to studio-based literature. Two activities that are frequently described as occurring often within a studio setting are reflection and critique.

> *"Studios are typically organized around manageable projects of design, individually or collectively undertaken, more or less closely patterned on projects drawn from actual practice. They have evolved their own rituals, such as master demonstrations, design reviews, desk crits, and design juries, all attached to a core process of learning by doing. And because studio instructors must try to make their approaches to design understandable to their students, the studio offers privileged access to designers' reflections on designing. It is at once a living and traditional example of a reflective practicum."* (Schön, 1987, p. 43).

Koch et al. (2002, p.3) succinctly summarise experiences that those who have studied in an architectural studio may have experienced:

> *"Late nights, exciting projects, extreme dedication, lasting friendships, long hours, punishing critiques, unpredictable events, a sense of community, and personal sacrifice. [...] Those aspects are not usually written into the curriculum or even the design assignments, but they are likely the most memorable and influential".*

2.2.1 Lack of a definition

Despite studio education being popular in architecture, design and art disciplines, and now surfacing within software disciplines, there is a distinct struggle with an

agreed upon studio definition. By no means is it suggested that no understanding of studios exist, far from it; however, descriptions and definitions of studios appear vague, and it is clear that an understanding of their complex nature resides in tacit knowledge (Löwgren and Stolterman, 2004) – literature cited in this section refers to studio education in arts and design. A studio could be the physical space, the teaching style or even the attitudes of the students. There are numerous understandings and implementations of studio education, with many nuances that are not covered in a succinct definition of a studio, especially as each varies depending on its context, as *"evidenced in the tremendous diversity of content and methods in studio teaching in different schools and even within one department"* (Ledewitz, 1985, p.2). Not much is understood still about studio learning and what goes on within the studios (Kvan and Jia, 2005), as they are often *"characterized by multiple and sometimes contradictory goals, implicit theories and inherent conditions of 'inexpressibility, vagueness, and ambiguity'"* (Ledewitz, 1985, p.2). This problem is addressed directly, later in chapter 3, through interviews with experienced studio educators in architecture, design and art.

For a long time it was considered that there was a *"lack of clarity over the purpose and effectiveness of the design studio"* (Ledewitz, 1985, p.2), even though at the same time it *"has been held up as a highly sophisticated means to teaching creative problem-solving"* (Ledewitz, 1985, p.2); dealing with this complexity and vagueness is part of the paradox of learning how to design (Schön, 1987), which is said to mean that you learn through doing it, but cannot do it until you have learnt it. The purpose of the studio seems well documented, teaching the paradox of design education, but a framework does not appear to exist to reflect on when implementing your own studio – it is likely more of a culture passed down through generations of designers based on their implicit experiences, as there are many intangibles that make up a studio. Though it is not succinctly defined, there are many descriptions of what a studio is and what goes on within it. This has led to attempts to utilise studios in other disciplines, trying to replicate the benefits

that studios afford Design disciplines. However, if you cannot concretely define a studio, perhaps you can model it instead to assess its effective implementation?

Despite the fact that there appears to be no agreed definition, there are many thinkers that have provided important observations and definitions of elements within studio education. The most widely cited thinker, Donald Schön, distils a core concept of studio education: 'reflective practice' (Schön, 1983). This takes place in a *'reflective practicum'*, and the students *"learn by doing, with the help of coaching"* (Schön, 1987, p.xii); this is described in greater detail in section 2.2.2. Another important facet of studio education is the use of critiques (Schön, 1987), and they are widely considered an *"indispensable tool"* (Uluoğlu, 2000, p.54); critiques come in various forms ranging from formal to informal, and individual or group – used as a mechanism to communicate ideas and evaluate designs. These are described in more detail in section 2.2.3. Several aspects of Schön's books (Schön, 1983; Schön, 1985; Schön, 1987; Schön, 1991) discuss elements of education and practice that can occur independently of studios, but they are discussed in the context of studio education as they are prominent in those spaces. Another observation made by Schön is that studios work with real world projects. Real world projects often deal with complex issues that can be categorised as "wicked problems" (Buchanan, 1992). Wicked problems are difficult to frame and have no "right" answer. Despite these observations, there is still a lack of a concrete agreed upon definition. However, there are three main elements of studio education that appear in the literature frequently: Reflective practice, critique and coaching. These are covered in further detail below.

2.2.2 Reflective Practice

This is a form of reflective learning by doing, with the help of coaching (Schön, 1987, p.xii). The notion was developed through observations of students and their professors in architectural studios, and is essentially about observing and refining practice in a continuous cycle. It is also considered to be the essence of

the studio concept (Schön, 1983; Tomayko, 1991). Some examples of the activities that support reflective practice include critiques (section 2.2.3) or writing reflective reports. Furthermore, the first software studio, by Tomayko (1991), was primarily based on the notion of reflective practice (achieved through *"constant questioning"* (Tomayko, 1991, p.301), see section 2.3.1); this studio has been used and referred to as a foundation for other software studios since then.

Reflective practice comes in two main forms: *reflection-in-action* and *reflection-on-action* (Schön, 1983), which are described in further detail below. McCall (2013, p.24) also partially describes reflective practice as *"repeated alternation between knowing-in-action and reflection-in-action"*; knowing-in-action is a form of embodied tacit knowledge. In many professions, these reflective activities are assumed to improve *"the proficiency and performance"* of practitioners (Hazzan, 2002, p.161); those professions that Hazzan refers to share several similarities with software engineering – indeed, one in particular, the architecture of buildings, is often used as a metaphor for software development (Kuhn, 1998).

Lecture-based delivery can suppress reflection due to its didactic style (Brockbank and McGill, 2007). Project-based courses are better in this regard because they give students the opportunity to rework solutions over time. However, such courses do not guarantee that they will "build in" reflection, primarily because reflection is often only considered implicitly in learning goals. Reflective practice is important here because practicums attempt to help *"students learn to become proficient at a kind of reflection-in-action"* (Schön, 1987, p.40).

Reflection-in-action

When someone approaches their work, there are times when they gain new insight by approaching a problem from a different perspective. This is what Schön refers to as reflection-in-action, or reflection during the problem-solving process (Schön, 1983), for example *"bringing past experience to bear on a unique situation"* (Schön, 1987, p.66). Broadfoot and Bennett (2003, p.3) further summarise reflection-in-

action as *"the questioning and challenging associated with problematic situations in practice – a reflective dialogue with the designer's own knowing in action"*. Others also provide their own succinct description, describing it as students learning *"not by assimilation but by trial-and-error practice"* (Wang, 2010, p.175).

Reflection-on-action

After problem solving has occurred, a practitioner may consider what they could have done differently or would do differently next time. This is what Schön calls reflection-on-action, or reflecting back on past experience. Schön (1987, p.26) states that it can occur *"after the fact, in tranquility, or we may pause in the midst of action"*, but it *"has no direct connection to present action"*.

One approach to reflection-on-action is highlighted by Schön: *"We reflect on action, thinking back on what we have done in order to discover how our knowing-in-action may have contributed to an unexpected outcome"* (Schön, 1987, p.26).

2.2.3 Critique

Critique aids reflective practice, and is widely used and talked about as a discrete mechanism. People giving critique is an *"important role in design both in the growth of human knowledge and in terms of error elimination"* (Fischer et al., 1998, p.540). It is also seen as a tradition in studio education, and it is therefore very common, but beyond the traditions it is a widely held belief that critiques are an *"indispensable tool"* (Uluoğlu, 2000, p.54) and a *"critical and fundamental means with which to teach design"* (Oh et al., 2013, p.321). There are many flavours of critique, each with their own unique properties (e.g. online crits, peer crits, group crits), but the main styles of critique are a 'Desk Critique', 'Design Review' and 'Design Jury' (Schön, 1987) – although each goes by various names or nicknames.

Desk Critique

The desk critique is central to the studio experience (Schön, 1983; Shaffer, 1997). This type of critique, more commonly referred to as 'Desk Crit', and sometimes simply just 'Crit', is usually a one-on-one activity between a coach and the student. The name is derived from the simple fact that these critiques generally happen with each student, during an on-going project, whilst they are sat at their desk. These activities are small, often informal, and cover elements such as the design and the thinking process.

A desk crit is supportive of spontaneous or scheduled interactions. A succinct description of the desk crit process is given by Shaffer (1997, p.201):

> *"a student describes his or her work to the professor, including areas of particular interest or concern in the design. The professor probes the design, asking for clarification where needed, and then isolating potential problem areas. As students present possible solutions, the professor explores the implications of various design choices, suggesting alternative possibilities, or offering ways for the student to proceed in his or her exploration of the problem".*

A desk crit can also take the form of peer critique (Hokanson, 2012). A peer crit is very similar to a desk critique, but with a peer rather than an instructor, and will likely be constructed around different social boundaries, for example, a desk crit may draw a small crowd of *"incidental learners"* that do not participate due to the 'expert' nature of the instructor (Hokanson, 2012, p.78). Peer crits on the other hand are less formal, and are in fact the *"least formal of the critique formats"* (Hokanson, 2012, p.79). Peer crits also provide the students with the opportunity to *"extend their own skill in critique"* (Hokanson, 2012, p.79).

Design Review

Sometimes referred to as a 'Group Critique', these critique sessions are usually performed in an *"informally structured environment"* (Hokanson, 2012, p.77), and

occur quite frequently – ideally multiple times during a project, to aid the learning-by-doing process. They can take the form of presentations, demonstrations or posters, and they have a flat audience hierarchy (that is to say, the non-presenting students taking equal place alongside members of staff in the audience, as a potential critic). These critiques are seen as *"more generative and educational"* than the final design jury counterpart, as they focus more on the *"improvement and development of the design project"* (Hokanson, 2012, p.77).

Design Jury

Sometimes referred to as a 'jury critique', 'pin-up' session, 'design crit', or often just as a 'jury'. Juries are very similar to design reviews, though are more formal and usually only occur at the end of a project – forming part of the final assessment. This particular critique gets its name from the attendance of *"a group of outside critics (the 'jury')"* (Schön, 1983, p.80), which is also a significant point that differentiates itself from design reviews, creating a very different social dynamic and set of interactions. Although critiques are considered very important to design education, in a cruel twist of irony regarding its name, design juries can be *"punishing"* and difficult for the students (Koch et al., 2002, p.3).

In a study by Dannels and Martin (2008, p.143-147), they identified a topology of feedback given during design juries through looking at various levels of design studio that range from novice to expert. These studios were for different levels of students in their educational career and included freshmen, junior, senior and a graduate studio. The study identified nine types of feedback:

- **Judgement** (25.4%) – Judgemental feedback that occurred when the critics react and *"rendered some assessment of its quality"*.
- **Process oriented** (20.8%) – Statements or questions about a student's design approach or process. Can be suggestive or observation-oriented.
- **Brainstorming** (18.3%) – Questions or statements about *"future imagined*

possibilities"; often taking the form of 'what if?' questions. The tone of the questioning *"clearly indicated that the questions were rhetorical"*.

- **Interpretation** (12.4%) – Trying to make sense of the concept or product.
- **Direct recommendation** (9%) – When specific advice is given about a design. This feedback is *"focused, purposeful, and usually specific"*.
- **Investigation** (5.1%) – Feedback that is geared towards requesting information, typically by questioning. This feedback *"necessitated a response"*.
- **Free association** (3.7%) – When critics make reactive (spontaneous) and associative statements about a design. Statements are *"often started with 'It reminds me of' or 'This looks like' "*.
- **Comparison** (2.8%) – When the feedback compares the design with something else. This differs to 'free-association' which is *"reactive (spontaneous) whereas comparison feedback was strategic (focused, intentional)"*.
- **Identity invoking** (2.5%) – Questions and statements aimed at making the students *"consider the larger picture of themselves as designers"* and *"within the larger context of the design profession"*. The tone of the questioning *"clearly indicated that the questions were rhetorical"*.

Dannels and Martin (2008, p.151) summarises their study as follows:

> *"Students in novice studios were exposed to feedback that was less collaborative and more consultative and directive. The judgment, direct-recommendation, and free-association categories (all more frequent in the freshman studio) all indicate a relationship in which the student essentially depends on the evaluation, direction, or reaction of the critic. Conversely, students in expert studios were exposed to feedback that was more collaborative, less directive, and less hierarchically driven. The interpretative, brainstorming, investigative, and process-oriented*

> *categories (more frequent in the three upper-level studios) all indicate a more collaborative relationship between the critic and the student– with the student as a more active agent and independent thinker and designer."*

They did however end by suggesting that what they observed was feedback that was *"more reflective of academic developmental stages or idealized workplace contexts than of actual professional settings"* (Dannels and Martin, 2008, p.135).

2.2.4 Coaching

There is a significant difference between coaching and lecturing. Typical definitions of lecturing refer to the lecturer giving or reading to an audience by formal discourse. It is amusing that the term 'lecturing' is also a synonym for reprimand or used to describe someone giving a lengthy scolding to someone else.

Coaching on the other hand is about providing instruction or advice. In a studio, Schön (1987, p. 38) states that teaching staff act as a coach, and their main functions in a practicum are:

- Demonstrating
- Advising
- Questioning
- Criticizing

Coaching can also come in the form of peer-coaching, whereby *"fellow students may, in various ways, play the coach's role"* (Schön, 1987, p.170). Schön goes on further to emphasise the importance of this type of coaching from peers by stating *"groups of students [...] are often as important to one another as the coach"* (Schön, 1987, p.38).

The importance of coaching over lecturing also resonates within some psychology literature, where Vygotskiĭ (1978, p.86) champions the notion of the 'Zone of Proximal Development' which effectively suggests that students learn better under

supervision or in collaboration with others. He describes it as *"the distance between the [student's] actual development level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers"*.

2.2.5 Arguments Against Studios

Although the design studio is currently widely used, and presents a solid argument for its continued use, there are also some arguments against it. The following are a couple of reasons why the studio is under threat, as outlined by Cruickshank (2006):

- *Space is expensive* – The first issue is that permanent space is an expensive premium. When an environment is to be used as a permanent home for the occupants, as well as their work, nobody else can use that space.

- *Less contact time* – Lecturers may have less contact time with their students. There is always a drive for increased participation and interaction, and studios provide this with coaching. However, without a corresponding increase in funding, lecturers are pushed to teach larger groups of students (or other non-teaching responsibilities), which also makes it more difficult to build trust with individual students. Whilst this argument depends on how an individual studio is run, it highlights the potential for contradictory objectives set by universities.

Another argument against the use of studios could be the potential impact on the student's health and work habits. Students working on projects individually in studios can potentially feel isolated with their work, as each student can work on widely different projects and work long hours in the studio. This resonates with Schön (1987, p.311) when he states that *"students do not so much attend these events as live in them"*.

One other argument is the potential "punishing critiques" (Koch et al., 2002). This relates more specifically to the practice of critiquing than to the studio itself, and is described earlier in section 2.2.3.

A report by Koch et al. outlines numerous myths that studio education perpetuates. Koch et al. (2002, p.5) say that these *"influence the mentality of students and promote certain behaviors and patterns"*. Following is the list of myths:

- *Architectural education should require personal and physical sacrifice*
- *The creation of architecture should be a solo, artistic struggle*
- *The best students are those who spend the most hours in studio*
- *Design studio courses are more important than other architecture or liberal arts courses*
- *Success in architecture school is only attained by investing all of your energy in studio*
- *It is impossible to be a successful architect unless you excel in the design studio*
- *Students should not have a life outside of architecture school*
- *The best design ideas only come in the middle of the night*
- *Creative energy only comes from the pressure of deadlines*
- *Students must devote themselves to studio in order to belong to the architecture community*
- *Collaboration with other students means giving up the best ideas*
- *It is more important to finish a few extra drawings than sleep or mentally prepare for the design review*
- *It is possible to learn about complex social and cultural issues while spending the majority of time sitting at a studio desk*
- *Students do not have the power to make changes within architecture programs or the design studio*

Although these myths are said to prevail *"within many design studios"*, that does not indicate that these scenarios are necessary or even occur (Koch et al., 2002, p.6).

2.3 Software Studios

There are several software studio implementations worldwide, where *"students in studio courses have shown higher levels of motivation and engagement than students in traditional courses"* (Narayanan et al., 2012, p.165). Although studio-based pedagogy is increasing in software disciplines, not many have been implemented (Hundhausen et al., 2008b). However, despite the apparent success of these few studios, it is also apparent that none of them follow a shared definition or understanding of studio education – making comparisons non-trivial. Because of this, previous studio attempts are collated below and detailed to provide the necessary context to this thesis.

The idea of software studios has been around for a couple of decades, with some of the first attempts at implementing one going as far back as 1991 (Tomayko, 1991). A software studio is an attempt to utilise similar environments to that of a design studio but for software related disciplines; many names have been given to these studios which include 'software studio', 'software development studio', 'software design studio' etc.

Although there are many instances where a variation of a design studio has been used to specifically teach design topics to computing students (Docherty et al., 2001; Greenberg, 2009), these are not the focus here. The focus in this literature review is not to explore teaching design through the medium of a design studio (e.g. visual thinking or interaction design outside of a software development context), but to see how aspects of a design studio could improve software design and development in disciplines such as computer science and software engineering.

There have been several attempts to integrate a studio-based learning environment into computing degrees to reasonable levels of success. There also appear to be studios that have been implemented with limited or no acknowledgement in the wider literature and just appear in course descriptions online (Wallingford, 2011; University of Nebraska-Lincoln, 2011).

Lastly, what was of most interest throughout this review was that most of the

software studios focused on utilising the design studio for the design aspects of software projects, but less so for the implementation, especially in HCI and other media-based courses. Implementation tasks at some institutions still followed traditional approaches and were relatively unaffected by any studio processes.

In this section, documented software studios (those that appear in publications) are identified and summarised, with various descriptions of implementation, curriculum, and intended audience – features that are explicitly stated about the nature of these studios. Table 2.1 collates and summarises the list of studios and their explicitly stated studio features and activities; all have been mentioned in publications from each of their respective institutions, with each studio described in more detail below. The only drawback is that some activities or features may indeed exist in a particular studio, but if it is not mentioned in their respective publications it cannot be listed in the table; therefore Table 2.1 is a reflection on what they describe as core studio elements, not necessarily what they think a studio is. The following subsections, in chronological order (according to their respective publication dates), look at implementations of these design studios within software-based degree schemes. Summarising these efforts, all of the attempts did successfully utilise certain features found in typical design studios. However, in those that only implemented a subset of features from a studio, the software environments effectively became another type of environment, but not a design studio – often these were environments that facilitated problem-based learning and had a minimal culture of critique.

2.3.1 Carnegie Mellon University (CMU)

CMU started offering a Master of Software Engineering with a significant studio experience in 1991 (Tomayko, 1991). This degree, including the crucial studio elements, is still being used today (Carnegie Mellon University, 2013).

This implementation of the studio has reflective practice at its core, enforced by the nature of "constant questioning" and by assigning faculty members to students

Table 2.1: Existing software studios' activities

Institution	Reflective Practice	Constant Questioning	Coaching	Open-ended Problems	Critiques	Rapid Design Iterations	Co-located Students	Frequent Presentations or Demonstrations
Carnegie Mellon University	✓	✓	✓	✓	✓		✓	
Massachusetts Institute of Technology	✓			✓	✓	✓		✓
University of Queensland	✓			✓	✓		✓	✓
Monash University	✓		✓		✓		✓	
Washington State University	✓				✓		✓	
Auburn University	✓				✓		✓	✓
University of Hawaii								
University of California – Irvine	✓			✓	✓	✓	✓	
University of Victoria	✓				✓	✓	✓	✓
Poznan University of Technology			✓	✓			✓	
Open University	✓		✓	✓	✓			

as coaches (Tomayko, 1991). The studio is also integrated across the curriculum, allowing many courses to have aspects developed into the studio projects. The way that this studio implementation had open-ended and complex problems to solve was by having a pre-selected problem from a real company; in the first year the course was run, the problem came from Boeing.

The studio was only used for group work; there were no individual projects to work on. The groups in the studios often suffered from bad group dynamics (Tomayko, 1996), but this was eventually smoothed out by moving students amongst the groups. Furthermore, the studio environment was only used for one day a week during the first and second semester, essentially using the room as a place to meet as a group for project management or reflection. This is out of step with a normal design studio where the students would be in there most of the time. This was, however, followed by a summer of heavy studio use, but this time it was used as a base of operations during the implementation stage of the project. In this regard, the studio does not benefit the implementation of projects, instead just supporting the design – during the summer it would be akin to any other environment where a group is collocated and implementing a project solution.

CMU appears to have the main principles set out by Schön (1987) as their guide: Reflection and coaching through the use of critiques and constant questioning. Indeed, when encouraging reflective practice, the students are expected to *"complete the real-world project, but also to reflect on the rationale, motivation, and consequences of their decisions and actions"* (Garlan et al., 1997, p.62).

2.3.2 Massachusetts Institute of Technology (MIT)

In 1995, a software design studio course was created at MIT (Kuhn, 1998) to explore what a design studio's practice and culture had to offer to software designers. The course was available for undergraduates and postgraduates, and was taught by Mitchell Kapor and Dean Mitchell, who between them had both architecture education and software design experience, with Ronald MacNeil and Sarah Kuhn

in support roles.

The course set out, similarly to standard design studios, with intentions to begin with a complex open-ended question, which was followed by rapid iteration of design solutions. There was also an intent to facilitate a culture of critique, which occurred mainly by faculty except during the weekly meetings where both faculty and peers gave critique to the presentations given. The down-side is that these opportunities for critique were only really available during the weekly meetings as *"students were not physically co-located as they worked on their projects"* (Kuhn, 1998, p.66); this is at odds with many studio definitions (section 2.2). The meetings were usually only for three hours a week and often consisted of presentations and demonstrations, only allowing for jury critiques. Contact time between the students will have been significantly lower than that of a traditional design studio – thus continual informal discussion and critique would have been a rare occurrence.

2.3.3 University of Queensland (UQ)

In 1999, UQ started a studio-based course called 'Bachelor of Information Environments'. This course consists courses such as Interaction Design, Multimedia, the Web, Computer Supported Cooperative Work and Information Appliances (Docherty et al., 2001), but based on a computer science foundation.

One aspect that this course taught was design skills (or skills in a design context) to computing students, for example, generative thinking, observation, production and integration. The course had open-ended real-world design projects to work on, involved teamwork, collaborative learning, interactive problem solving, presentations and peer review – so there was scope for reflective practice and critique.

The course integrated design skills and thinking into a computer science degree for the purposes of designing information environments. The studio was used as a place where skills learnt from other courses were applied.

In 2011, a further software engineering studio was developed at the University of Queensland (Süß and Billingsley, 2012). The core approach used in this studio

was continuous integration practices, where a large student group all worked on a legacy code base. There is a focus on just-in-time teaching, where lecture content is *"delivered just-in-time to address issues the class should be facing in their project at a given stage"* (Billingsley and Steel, 2013, p.213).

In this course a monitoring tool is used to observe the code base (e.g. whether tests are failing and commits etc.). This same tool also allows *"the teaching staff to monitor the class's collective progress reducing the need to meet regularly with each small group"* (Billingsley and Steel, 2013, p.213). This has a practical benefit for teaching staff that are stretched for time, but also reduces potential benefits that the students may receive through frequent contact with teaching staff. However, in a revision of their studio, the students benefit from increased peer-engagement because the teaching staff use the *"marking criteria to promote inter-group co-operation and earlier development"* (Billingsley and Steel, 2013, p.214).

This studio course also used to provide student-delivered content, which followed a *"teach-to-learn pedagogy"*, but it was removed from the course after a review. This type of content was reported by students in their course reviews as a distracting task (Billingsley and Steel, 2013, p.216).

In recent observations of this second studio at UQ they state that *"allowing the students to choose their own project topics was a success"* (Billingsley and Steel, 2013, p.218). Interestingly, they are also considering to attempt opening *"parts of the course up to online delivery"*, they expect that it *"will also present a challenge"* (Billingsley and Steel, 2013, p.218).

2.3.4 Monash University

Monash University has two studios that have been published: 'Bachelor of Information Management and Systems (BIMS)', and 'MUSE Studio Lab'.

Bachelor of Information Management and Systems (BIMS)

> *"In 2000, the School of Information Management and Systems, at Monash University, was awarded a Strategic Innovation Fund grant ($70,000) to institute an innovative teaching and learning model based on a studio approach in a core subject of the Bachelor of Information Management and Systems (BIMS)."* (Carbone and Sheard, 2002, p.213)

BIMS has a *"strong practical focus"* (Lynch et al., 2002, p.75) and is based on the Bauhaus school of design. They state that this model is used for *"the development of collaborative learning, integrated curriculum and problem-based learning"*(Lynch et al., 2002, p.75). Central to the BIMS program is a compulsory *"year long studio subject in each year of the program"*, with other core and elective subjects running alongside (Lynch et al., 2002, p.75).

During their implementation of their studio, Lynch et al. (2002, p.75) identified that approaching a studio required a radical rethink of their:

- *"layout and design of the physical teaching space"*
- *"teaching and learning approach used"*
- *"IT infrastructure"*

The studio space is a transformed classroom space that was re-architected to include *"the studios themselves, an Internet café, a meeting room, and an area for technical support staff"* (Lynch et al., 2002, p.75). The studios are where the main interactions and collaborations occur. The café was hypothesized to improve student performance by allowing a break away space, to escape and get *"relief from intensive studio activity"* (Lynch et al., 2002, p.76).

They claim that they adopted a *"mentor and apprenticeship relationship"*, following one of the Bauhaus philosophies; however they do not provide details of how this relationship was implemented or reinforced (Lynch et al., 2002, p.77).

The course identified the need to alter the assessment of the course, stating that *"typically in IT courses students are required to submit specific solutions to a question"* (Lynch et al., 2002, p.77), but opting to include portfolio assessments (similar to a design studio). To navigate the change in format, they combined assessment techniques of both traditional IT and design studio assessments: the main difference is the assessment of the student's work through the *"presentation of a portfolio of work or a product"* (Lynch et al., 2002, p.77).

Lynch et al. (2002, p.78) performed a survey and found the responses generally positive, with the studio giving them *"the opportunity to chat with peers in an informal, flexible and collaborative way, but also 'forced' them to communicate, collaborate and learn"*, which also provided the students with the *"opportunity to further develop their communication and analytical skills"* (Lynch et al., 2002, p.79).

Most students felt *"having a number of lecturers has been beneficial"* (Lynch et al., 2002, p.78). However, *"some students found having a number of lecturers a little confusing and they found it difficult to develop a rapport with any one of them"* (Lynch et al., 2002, p.79).

MUSE Studio Lab

This studio is part of Monash's Bachelor of Software Engineering program, and is only used by its final year students. In this course, the studio is used in their team-based Capstone SE project, where *"teams have worked in groups of four on real client projects over two semesters"* (Ramakrishnan, 2003, p.22), with the other units (modules) following a *"more traditional teaching approach"* (Ramakrishnan, 2003, p.21).

Ramakrishnan (2003, p.21) briefly describes studios as a setting where *"students learn by doing, working in teams and by seeking assistance from mentors when needed"*; these mentors guide the students *"in such a way as to promote reflective thinking and the abstraction of knowledge"* (Ramakrishnan, 2003, p.24). They

also acknowledge virtual environments play a role in studios, *"Studio is often a physical/virtual environment"* (Ramakrishnan, 2003, p.21), though it is not clear if they mean a mix of physical and virtual, or potentially just virtual – however, their implementation is a physical environment that is supported by a virtual one.

Ramakrishnan (2003, p.21) goes on to describe how their studio implementation differs from other units in the degree program :

- *"the physical layout"* of the studio facility.
- *"the hardware and software resources required"*
- *"the teaching approach used in terms of content and method"*
- *"the development of strategies for integrating the formal and pragmatic aspects of SE, by getting students to reflect (muse over) and see links between the formal aspects of SE, critical thinking and research methods in the thesis component"*
- *"the Studio project and the software tools used to support the SE life cycle."*

Their studio is *"aimed at supporting agile SE practices"* (Ramakrishnan, 2003, p.21). However, to avoid the possible deterioration of light-weight agile methods into *"adhoc development"*, the MUSE studio used a *"tailored middle-weight process which requires students to devote enough effort in requirements and design models and move more to the agile method during implementation"* (Ramakrishnan, 2003, p.22) .

Assessment in this studio is continual, with the teams building up a portfolio for their project which also includes reports and log books. Throughout the year there are two instances of peer-assessment, each team gives two presentations and perform two walk-throughs. The portfolios are also presented at the end of the project in a 'Design Jury'.

2.3.5 Washington State University

Hundhausen and Brown (2005, p.45) explored the pedagogical value of studio experiences by implementing and studying them within their *"introductory CS1 unit on algorithmic problem-solving"*. They state that the reason for their continued exploration of studios is because of the *"increasing importance of communication, teamwork, and critical thinking skills"* (Hundhausen et al., 2010, p.500). The studio course at Washington is supported by a series of lectures, but they only took place *"during the regularly-scheduled two hour and fifty minute lab periods in the second and fourth weeks of each course"* (Hundhausen and Brown, 2005, p.47). In this unit, the students were given algorithm design problems and tasked with *"constructing algorithmic solutions and accompanying visualizations"*, and *"presenting their visualizations for feedback and discussion in a session modeled after an architectural 'design crit'"* (Hundhausen and Brown, 2005, p.45). The type of design crit that they refer to is not explicitly stated in their paper, although they do describe it: the students *"presented their visualizations for feedback and discussion during presentation sessions scheduled after the assignment due date"* (Hundhausen and Brown, 2005, p.46), which would imply that these were likely jury critiques (see section 2.2.3). For their studies, Hundhausen and Brown utilised 'participant observation', 'videotaping', 'artifact collection', 'interviewing', and 'questionnaires'.

A key observation from their studies was that *"students benefited from constructing and presenting their own visualizations not only because this exercise increased their motivation and level of interest in algorithms, but also because it stimulated meaningful discussions about algorithms"* (Hundhausen and Brown, 2005, p.46).

Another important observation was related to the types of technology the students used to perform the tasks, whether "high tech" software (called Samba), or "low tech" materials (pens, paper and scissors), was that it *"had a significant impact both on the focus of students' activities during the construction phase of the assignment, and on the focus of the conversations during the presentation sessions"*

(Hundhausen and Brown, 2005, p.46). When the students used the "high tech" approach they *"tended to spend inordinate amounts of time steeped in low-level implementation details"* which was then followed by sharing *" 'programming war stories' when they ultimately discussed their visualizations with the class"* (Hundhausen and Brown, 2005, p.46). However, with the "low tech" approach the students *"tended to focus their attention squarely on the algorithm itself both in the construction phase, and in the conversations"* (Hundhausen and Brown, 2005, p.46). They later present data in a graph showing that the "low tech" art supply approach spent a greater percentage of time communicating, but in other data, the semantic error rates of the solutions were significantly higher in the students using the "low tech" approach – they postulated that this was most likely because of *"the text editors they used for algorithm development made it difficult for them to determine if their coding efforts were on track"* (Hundhausen and Brown, 2005, p.51).

Hundhausen and Brown (2005, p.55) conclude in their paper that *"studio-based pedagogy is a step in the right direction, but will require more refinement"*, and also provide a set of recommendations for educators interested in exploring studios:

- *"Make sure that novice students have consistent access to knowledgeable experts during the coding process"*
- *"Make sure to give students enough time to develop their algorithmic solutions"*
- *"Make sure to have skilled facilitators for the presentation sessions"*

Finally, Washington is part of a *"multi-institutional research project"* (Hundhausen et al., 2008b, p.393) in which they aim to *"refine, empirically evaluate, and build a community of educators"* around the studio-based approach to teaching (Hundhausen et al., 2008a, p.393). The institutions included in this particular relationship are Washington State University, Auburn University and University of

Hawaii at Manoa. Updates related to this venture are also posted online: *studiobasedlearning.org*.

This relationship has continued, and they now state that they have involved *"26 computing courses at 15 institutions in seven states"* with studio-based learning (Narayanan et al., 2012, p.165). However, information about these particular 26 courses has not been found. Based on how Narayanan et al. describe and define studio education, it can be concluded that their definition is limited. Their main focus is on a few set activities (e.g. the "design crit"), and the students have limited exposure to studio sessions compared to other studios described within the art and design disciplines. This reinforces the need to explore or provide more descriptive definitions and understandings of studio education.

Pedagogical Code Review

This activity developed at Washington State University is an *"adaptation of the formal code inspection"* (Hundhausen et al., 2009, p.291) employed within Washington State's studio-based course that adapts the studio approach for *"lower-division computing courses, which traditionally revolve around a series of individual programming assignments"* (Hundhausen et al., 2010, p.500). This technique is essentially a process where *"developers first review a piece of code individually, and then come together as a team to log issues[. . .] that they encountered, and to identify additional issues as a team"* (Hundhausen et al., 2009, p.291). The authors also stated that this process enabled students to *"exercise a variety of skills that are increasingly important for computing careers, including (a) computer programming, (b) critical review of their own and others' code, and (c) communication and teamwork"* (Hundhausen et al., 2009, p.291).

In an evaluation of this technique within a studio-based course and an identical course without this technique, they found *"no learning outcome differences between the two courses"*; however, they did find that the use of the pedagogical code review in a studio-based course had two interesting trends (Hundhausen et al.,

2010, p.500):

- *"self-efficacy decreased more in the traditional course than in the studio-based course; and"*
- *"peer learning decreased in the traditional course, but increased in the studio-based course."*

OSBLE

The 'Online Studio-Based Learning Environment'[1] is an online environment that *"supports both asynchronous and face-to-face [Pedagogical Code Reviews]"* (Hundhausen et al., 2013, p.14:6), developed at Washington State University. It is used for constructing and maintaining teaching and learning communities based on studio-based learning. The OSBLE website describes it as enabling course *"instructors to set up and maintain online presences for their courses"*. One of the claimed unique points of OSBLE is *"its powerful support for studio-based assignments"*.

In a video preview of OSBLE 2.0, OSBLE is described as *"a learning management system that enables studio-based learning activities to take place asynchronously and online"* (Hundhausen, 2011, 1:06).

2.3.6 Auburn University

Hendrix et al. (2010, p.505) state that new approaches to teaching computing education are needed because it is *"critically important to increase student enjoyment in problem solving and raise student motivation levels and their interest in computer science"*. For this reason they explored studio-based learning, but acknowledge that studio education's *"implementation in computing education can vary widely"* (Hendrix et al., 2010, p.505). Therefore they ground their understanding of studios on two core features identified by Hundhausen et al. (2008b): Students construct their own artefacts and design critiques.

[1] Online Studio-Based Learning Environment: osble.org

Hendrix et al. adapted studio-based learning to their CS2 'data structures' course. One of the activities they employed was the use of anonymous peer-reviews, whereby work would be anonymously redistributed back to the students after a submission. The students were expected to read and review each other's work and *"make meaningful comments regarding its correctness, efficiency, creativity, etc."* (Hendrix et al., 2010, p.506). This hit a hurdle when the teaching staff realised that *"unless there were significant point values associated [with] the design, peer reviews, and oral response, students wouldn't put as much effort into those components as we would like"* (Hendrix et al., 2010, p.506). Later, Hendrix et al. (2010, p.507) had distilled the following five activities as being *"fundamental to our implementation of the studio-based learning model"*:

1) *"Develop multiple solution strategies."*

2) *"Justify the choice of one solution strategy."*

3) *"Present work-in-progress to peers and respond to comments."*

4) *"Submit final work to peers for review."*

5) *"Review the work of others."*

One of the purposes of their research was to gather data to allow them *"to compare the difference in both student learning and student motivation and attitudes"* (Hendrix et al., 2010, p.508) between the studio-based course and traditional course. However, publications sharing results about student motivation at Auburn University have not been found. The data capture techniques used were: *"Pre/Post course content tests, Pre/Post attitude and sense of community surveys, video recording of studio sessions, all graded items from the course, and exit interviews of selected students from the top, middle, and bottom of the class"* (Hendrix et al., 2010, p.508). After examining the graded work of the courses, they *"saw significantly higher performance from students who took the studio offering"* (Hendrix et al., 2010, p.509).

Lastly, Hendrix et al. (2010, p.509) summarise the elements that they think the students gain experience in:

- *"individually and collaboratively solving computational problems"*
- *"evaluating and selecting among competing solutions based on engineering design issues"*
- *"explaining their solutions to others in writing and through oral presentations"*
- *"critically analyzing each other's solutions in peer reviews"*
- *"reflecting on and learning from these studio sessions over the course of a semester"*

2.3.7 University of Hawaii

Publications about this university's studio implementations are limited. However, it was working in conjunction with Washington State University and Auburn University as part of a *"multi-institutional research project"* (Hundhausen et al., 2008b, p.393). The University of Hawaii has claimed that it has studios implemented into its pre-CS1, CS1, CS2 courses (Hundhausen et al., 2008b, p.395), but no specific publications about them could be found.

2.3.8 University of California – Irvine

This university (UC Irvine) offers two software design courses within their Informatics major, started in 2004. The first course is offered at the end of the sophomore year, and the second course is at the start of the junior year. The placement of these courses *"sets the stage for an advanced software architecture course"* in the senior year (Baker and Hoek, 2009, p.519). These studio-based courses focus more on the design process, rather than the design artefacts, and acknowledge that *"fre-*

quent and open critique is a key technique in honing students' design skills" (Baker and Hoek, 2009, p.519).

Baker and Hoek (2009, p.519-520) state that in an attempt to *"provide a broader perspective on design"* the studio courses at UC Irvine follow four key principles:

- Provide a General Design Perspective – The authors state that they *"have spent time previously in studying design in general"*, and have designed their courses using their perspective of those courses.
- Encourage Design-Minded Thinking – The courses would *"explicitly discuss and promote these 'designerly' behaviors"*.
- Separate System Design and Implementation Design – Separating them into the first and second studio respectively, which is claimed to allow students to *"more easily relate their activities to non-software design fields"*.
- Balance Theory and Practice – This is to help students *"assimilate their experiences in class"* and forming the *"basis for long-term learning"*.

In some initial lectures there is a focus on *"non-software products"*, in an attempt to support *"non-software design activities"* (Baker and Hoek, 2009, p.520). The teaching staff also take pictures and movies of the students designing during these activities, which are used for future reflection on their designing processes, with the intention of getting the students to compare it to how they design software. These courses were reported as successful.

2.3.9 University of Victoria

At this university, Estey et al. are exploring studio-based learning in a 'game design' course. Their motivation was to move towards a focus on *"communication and teamwork skills in addition to programming skills"*, to enable a smoother transition into industry (Estey et al., 2010, p.64).

In their course they introduced a set of activities that focused on *"cooperative learning, group orientation, and peer review, instead of programming skills"* (Estey et al., 2010, p.64), restructuring existing peer review activities on their course. Peer review activities were integrated *"into every step of the game development process, allowing students to refine their ideas while progressing through a game project"* (Estey et al., 2010, p.64). In their approach to designing peer review activities, they considered issues that are apparent in peer review: *"students feeling uncomfortable with peer review activities, feeling unqualified to give effective feedback, and participating in conspiracy activities"* (Estey et al., 2010, p.66). Others have disagreed with avoiding these issues, for example Schön (1987) says this is part of the process to becoming a competent designer.

The analysis of the course, through questionnaires, *"indicated increased student interest in pursuing a computer science degree"*, which was one of their goals for exploring studio education, however, *"the same pre- and post-surveys suggested that our collaborative activities may have resulted in a decrease in student interest regarding course work and in pursuing studies in game design"* (Estey et al., 2010, p.64). Yet the peer review activities did have positive influences on the students (Estey et al., 2010, p.64):

- *"Sense of community and competition"*
- *"Motivation and excitement towards class performance"*
- *"Perceived value and applicability of peer feedback"*

The peer review activities were *"successful in increasing student presentation confidence, sense of community, and excitement towards their course projects"* (Estey et al., 2010, p.64), and therefore also succeeding in improving their other 'motivation' goal. But beyond motivation, their exit survey also *"revealed that the peer review activities had a positive effect on student engagement, and that students felt the peer review activities were effective in helping them improve their game projects"* (Estey et al., 2010, p.70).

2.3.10 Poznan University of Technology

Kopczyńska et al. (2012, p.13) feel, like others, that studio education is suited to applying theory garnered from other courses or teaching approaches within the same course, for example they state that *"of course, [it] requires that other courses included in the programme of studies provide sufficient theoretical background, and that they are well integrated with the software studio"*. Strangely, the authors state that a *"software development studio is a frequently used method"* (Kopczyńska et al., 2012, p.13), which is contrary to most other authors in this review.

The aim of this studio is to be similar to a software studio observed in industry *"that enables students to practice all the roles in near-real-life conditions"* (Kopczyńska et al., 2012, p.14), which has been running in their Master degree (M.Sc) since 1998, and also occurs in their Bachelors (B.Sc). In this studio the projects are developed by teams *"of two or three M.Sc. students and four B.Sc. students. [...] the Master students serve the roles of project manager, architect, and analyst, while the Bachelor students are software developers"* (Kopczyńska et al., 2012, p.14). The studio projects try to replicate real industry projects by introducing *"real customers and budgets"* (Kopczyńska et al., 2012, p.16).

Kopczyńska et al. (2012, p.16) conclude by stating that *"Treating the whole studio as an organization, with its own culture, processes, and portfolio of projects, helps students to understand how real organizations operate"*.

2.3.11 Open University

The Open Design Studio, developed at the Open University, is an attempt to provide a design studio approach which is feasible in distance education (Petre, 2013). They state that they have demonstrated that their online studio is both an effective online shared space, and that it scales (something physical studios struggle with). This studio was developed by the author's colleagues in design, however, *"it provides both a model and an infrastructure that are readily applicable in CS"* (Petre, 2013, p.45); with its origins in design, it emphasises that the online

design studio is *"not just about the product but also about the dialogues: between student and teacher, between design and criticism, between design and its intended audience"* (Petre, 2013, p.44). This should prove promising because, being from the design discipline, the Online Design Studio will likely have been built using valuable tacit knowledge.

The focal point of their Online Design Studio is a website *"which is structured around learning communities and features students' work as it develops"*, and also *"hosts the online material"* (Petre, 2013, p.44). Also noteworthy is their approach to support greater communication online, by embedding the online studio "in a virtual learning environment (Moodle), which provides a social networking platform and uses Elluminate, a web conferencing system, for synchronous discussions". The website also has a 'pinboard area' which allows students to post content that is relevant to their work, for example *"images, text, source material, and video"* (Petre, 2013, p.45).

Students are expected to interact with other students, emphasising process over product, and to *"explore and comment on other students' work"* (Petre, 2013, p.45). In a description, the *"vivid, visual nature"* of the studio is also supposed to make the virtual space *"easy to 'wander around' [...] and spot things of interest"* (Petre, 2013, p.45). To help serendipity and promote exploration in this virtual studio, *"the studio has an enhanced search facility that combines retrieval on keywords with a random element"* (Petre, 2013, p.45).

To support the emphasis of process over product that studios normally afford, this online studio also integrates a tool called Compendium *"a visualization tool for mapping and management of ideas, to articulate their design process"* (Petre, 2013, p.45). This then helps inform the assessment of the students' work because *"it is this overview of and reflection on the design process that is assessed, rather than the design product"* (Petre, 2013, p.45).

To conclude this offering, Petre (2013, p.45) suggests that this online studio *"highlights key choices that underpin the effectiveness of the studio-based approach*

in distance education":

- *"the choice of open-ended problems"*
- *"the focus on process and practices"*
- *"the integration of the critical dialogue in the context of a smaller group of students"*
- *"the support for the teacher's role"*
- *"the facilitation of exploration"*

2.4 HCI Studios

There are a few software studios that are utilised for HCI (Human Computer Interaction) courses. These studios inherently work similarly to traditional design studios because the type of work that is performed within them is more akin to that of a product or graphic design, rather than, say, software architecture or development.

2.4.1 University of Oregon

The HCI design studio course offered at the University of Oregon is called *"Designing Software for People"*, and was *"created and taught in spring 2002"* (Reimer and Douglas, 2003, p.191). HCI courses are potentially well-suited to studio teaching, on the premise that the HCI curriculum is constructed around the ability to *"teach competence in designing interactive objects"* (Reimer and Douglas, 2003, p.191), something that the design disciplines have been doing for a while using the studio approach.

To understand the studio approach, Reimer and Douglas (2003, p.193) *"spent approximately a full term observing its use in the School of Architecture at the University of Oregon"*. Importantly, they summarise their observations into the following list (Reimer and Douglas, 2003, p.194-195):

- *"It involves experimental, immersive pedagogy: learning by doing"*
- *"Prior knowledge acquired from standard lecture courses is integrated"*
- *"Students produce realistic artifacts using realistic design processes taken from professional practice"*
- *"Students are active learners with teachers as resources or coaches"*
- *"Collaboration is a key process between teacher and student, and student peers"*
- *"Student assessment is based on presentations of design artifacts"*
- *"Learning environment between students and teachers, students and students, and even students and professionals is intense and highly interactive"*
- *"Primary teaching is through the 'design crit' "*
- *"Communication of the design and reflection upon both the product and process is critical to learning"*
- *"It involves specialized studio rooms and scheduling"*

Based on these observations, their studio course incorporated several aspects, which Reimer and Douglas consider critical to a traditional studio and summarise as: *"weekly design problems, collaboration between students and faculty, production of realistic artifacts in a realistic process, and weekly design crit sessions"* (Reimer and Douglas, 2003, p.203). However, they *"were not able to provide specialized studio rooms and real studio scheduling"*, which they admit would be a *"major improvement"* (Reimer and Douglas, 2003, p.203).

2.4.2 University of Calgary

At this university, HCI is taught using a studio approach. When facing the question of how to *"pass on the best practices of design studios within traditional programs*

that follow a standard lecture/tutorial format", Greenberg (2009, p.23) decided to explore a solution that he suggests created *"a design studio atmosphere within a lecture/tutorial time-frame"*. The students are required to sketch out ideas and publicly show the sketches to their peers for critique. Communication and idea exchange is encouraged, something he perceives conventional courses would call 'cheating'. A benefit to this approach is that *"students and instructors see each other's work as it is being develop[ed]"* (Greenberg, 2009, p.23). Greenberg feels that *"these programs point the way"*, but also acknowledges that currently *"they are few and far between"* (Greenberg, 2009, p.24).

In his implementation he set a cap of 15 students to avoid too many students making his studio *"unworkable"*. In his words, it was enough *"to form a critical mass, but was sufficiently small to encourage cross-student interaction"*, and also ensure that all students had sufficient time to *"demonstrate their work within the 2 ½ hour studio blocks"* (Greenberg, 2009, p.27). The success of this studio is expressed by Greenberg who is *"astounded by the quality and the diversity of the students' projects [...] all consistently surpassed my expectations"* (Greenberg, 2009, p.37).

Greenberg is also one of the few to include an acknowledgement of his prior experience with studios: *"I am not a formally trained designer. My early attempts to create a design-studio course relied heavily on advice by others"* (Greenberg, 2009, p.40).

2.4.3 Washington State University

Washington State makes an appearance again, but this time with a particular activity: the *Prototype Walkthrough*. This is another activity developed by Washington State University (in conjunction with the Open University) that is stated as a studio-based learning activity for HCI Courses (Hundhausen et al., 2011; Hundhausen et al., 2012). Hundhausen et al. (2011, p.117) reason that *"Because of its design orientation, human-computer interaction (HCI) education is an excellent*

venue for studio-based instruction".

The 'Prototype Walkthrough' is essentially an activity where *"a student project team simulates its evolving user interface prototype while a student audience member acts as a test user".* This student audience is encouraged to *"ask questions and provide feedback"*, encouraging peer critique and reflective practice (Hundhausen et al., 2011, p.117).

2.4.4 The University of British Columbia

The 'HCI learning studio' is another example of a studio used for a software course that is more akin to a design course, or product design studio, than a software studio because *"much of the work we do is not on the computer, and further, it involves non-electronic media – including physical and semi-functional mockups of actual interfaces, and lots of hand sketches"* (University of British Columbia, 2014, online).

2.5 Similar or Alternative Learning Approaches

Some approaches to learning will share similarities with studio-based education. Some of them are often mentioned in conjunction with, or as synonyms for, studio-based learning. The purpose of the following subsections are to provide definitions, and additional context, to some of the alternatives to studio education – some can also accompany a studio approach.

2.5.1 War Room

A war room is a collaborative space that is geared towards highly collaborative work with co-located colleagues, and are intended to *"support collective work"* (Fällman, 2007, p.7). It was traditionally a space designed around a central table, the main focus of the room, where highly collaborative work takes place and *"team members work together synchronously in all phases of a project, meeting for a vari-*

ety of purposes beyond status reviews" (Mark, 2002, p.89). A war room may house a variety of project members with various skill-sets, is typically used in conjunction with a project's design, and provides the inhabitants with immediate access to information and colleagues within the room (Mark, 2002). These environments are also suggested to *"lead to increased productivity"* (Mark, 2002, p.89). Mark (2002, p.89) goes on to succinctly describe the war room in the context of the type of activity it is used for: *"I call this type of design activity 'extreme collaboration' to emphasize working in an enclosed electronic and social environment to maximize communication and information flow".* An interesting benefit to these highly collaborative spaces is described as the 'cocktail party phenomenon':

> *"The cocktail party phenomenon, or when one attends to specific words in a noisy environment, is often observed in the war room, when keywords spoken aloud are clearly relevant to an engineer's particular subsystem"* (Mark, 2002, p.91).

On the surface these spaces sound similar to studio spaces. However, Fällman (2007) describes the difference between a studio and a war room by stating that a studio is a creative space, is semi-open, semi-personal, the furniture can be adjusted, and is semi-individual in nature. The war room, on the other hand, is described as a dedicated, shared, and private space that can be used by the currently most pressing shared project (Fällman, 2007). It is used when collaboration is required.

There are, however, documented drawbacks to such highly collaborative spaces. Noise, distraction and reduced privacy are common problems in war rooms, yet Mark still states that *"engineers comfortable with the noise and distraction of working closely together, a technology 'war room' [...] is the perfect environment for speeding delivery of [...] design proposals"* (Mark, 2002, p.89).

2.5.2 Project-Based Learning

At a glance, studio-based learning may appear to be the same as project-based learning. This is because studio education contains elements of project-based learning. A major component that separates the two is described by James Tomayko: *"It is the application of reflective practice that is the first distinguishing characteristic that separates the studio from a typical project course"* (Tomayko, 1991, p.301). Students working on group and individual projects, as well as often working towards real-world projects. Definitions of PBL can be quite broad and varied. PBL can broadly be defined as real-world problems explored in a problem-solving context. A summary of various definitions is also provided by John Thomas: *"projects are complex tasks, based on challenging questions or problems, that involve students in design, problem-solving, decision making, or investigative activities; give students the opportunity to work relatively autonomously over extended periods of time; and culminate in realistic products or presentations"* (Thomas, 2000, p.1).

PBL shares several similarities and benefits with studio education, although when properly administered, a studio can significantly improve learning over PBL (Tomayko, 1991). PBL can allow for improved reflection over traditional teaching, but it is possible that there is an even greater opportunity for reflective practice in studios. Some examples of this can be observed through the use of the culture of critique and co-location with other students; these can occur in a PBL course, but they are not required to. Another example is that PBL does not necessarily require a physical space that the students can call 'home'. For many project-based courses, students meet infrequently, at pre-arranged times, which is in contrast to a studio where student interactions are constant, spontaneous, and serendipitous. A dedicated physical space also affords the opportunity for students to produce design work that remains up on whiteboards or studio walls. Students can constantly refer back to their work, which is not possible in a classical PBL course because students do not have their own space. It is also suggested that those who administer PBL

courses should look into studio education to determine in which ways they can improve their PBL courses (Kuhn, 2001).

2.5.3 Peer Instruction

This is an alternative to traditional approaches to lecturing, pioneered in the Physics discipline by Mazur, and *"actively involves the students in the teaching process"* (Mazur, 1997, p.xi). Watkins and Mazur summarises peer instruction as an *"interactive teaching technique that promotes classroom interaction to engage students and address difficult aspects of the [taught] material"*, which is achieved by structuring *"time during class around short, conceptual multiple-choice questions"* called 'ConcepTests' (Watkins and Mazur, 2013, p.37).

Peer instruction, as described by Watkins and Mazur (2013, p.37), typically employs the following process:

1. *"The instructor starts with a brief presentation or summary of the material to be covered"*

2. *"The instructor poses a ConcepTest and asks students to think about the question and related concepts"*

3. *"After 1–2 minutes of thinking, students commit to an individual answer by using clickers, flashcards, a simple raising of hands, or writing down the answer on a piece of paper"*

    - *"If too few students respond with the correct answer, the instructor may revisit the concepts using lecture or try a different ConcepTest"*
    - *"If a large majority of students respond correctly, the instructor typically gives a brief explanation and moves on to the next topic or ConcepTest"*
    - *"If 30–70% of students answer the ConcepTest correctly, the instructor asks students to turn to their neighbors and discuss their answers. Students talk in pairs or small groups and are encouraged to find someone with a different answer"*

4. *"The teaching staff circulates throughout the room to encourage productive discussions and guide student thinking"*
5. *"After several minutes, students answer the same ConcepTest again"*
6. *"The instructor then explains the correct answer and [...] may pose a related ConcepTest or move on to a different topic or concept"*

Watkins and Mazur (2013, p.37) collates sources that state data shows *"improved performance in PI courses"*.

2.5.4 Flipped Classroom

Also known as 'flipped teaching' or 'inverted classroom'. This approach mixes offline and online learning by reversing the role of a lecture, taking *"advantage of the benefits of both collaborative learning and distance learning while at the same time targeting the millenial student"* (Gannod et al., 2008, p.785). Typically, lectures are used to present information to the students, and they work on projects and interact with teaching staff outside of the lecture session (often through an additional dedicated practical session). A flipped classroom is where lectures are recorded and made available online to students. They are expected to watch the video lecture before attending a classroom session; this session is used for working on the projects and interacting with the teaching staff. This could be a mechanism that can teach theory that some say is better suited to lectures, whilst still allowing for benefits offered by studio education.

2.6 Conclusions

This literature review was undertaken to explore what studios are, and to determine which software studios are currently in existence. A set of foundational elements found in studios was explored by reviewing literature from prominent thinkers in studio education literature, and several institutions' implementation of a software studio are presented.

Although at this stage it is difficult to distil clear elements that make up studio education, due to their complex nature and lack of an agreed definition (see section 2.2.1), it was apparent that many of the discussed elements had a focus on 'face-to-face' interactions. Three facets of studio education frequently described are reflective practice, critique, and coaching. However, specific implementation details are vague and vary across institutions.

According to the literature surveyed, there are not many institutions that have explored or attempted studio education in the software disciplines; although one publication stated that they have been involved with *"15 institutions"* with studio-based learning (Narayanan et al., 2012, p.165), but do not expand on who, where, or how. Of those in this literature review, most report success, with few very issues. Some report that the students are performing the same or better than their traditional course counterpart. Many are reporting an increase in soft skills, such as communication and collaboration.

Some institutions have focused on a few elements of studio education, namely cooperative learning and peer review activities, such as Hundhausen and Brown (2005) who focuses on the "design crit", and other institutions who followed in their lead, such as Estey et al. (2010) and Hendrix et al. (2010). However, this is only a narrow view of studio education, as Schön has set forward. Studio education is a complex and nuanced space, and set of activities, with the "design crit" making up just a portion of it. Looking at critique as a whole (see section 2.2.3), there are a variety of critique techniques used in studio education. With some institutions focusing primarily on a "design crit"; this does not adequately convey what sort of critique is being performed (e.g. 'desk critique', 'design review', or 'design jury'). This gives further credence to the notion that a lack of a concrete understanding of studio education exists.

Even though studio-based education is occurring more often, there are few that have completely overhauled their respective courses to be fully studio-based; often the studio is used for one of several modules that run during a term/semester,

and sometimes the studio is only used for critique and discussion and not, for example, the labour of the task (e.g. implementing code). Furthermore, students may spend limited time in the studio, in some cases they are being used *"roughly one day a week"* (Tomayko, 1996, p.121) – this is contradictory to what is observed of design or architecture studios (Schön, 1987). Software studios are often only seen as environments used for group work. The down side to this is that there would be no individual projects being worked on in the studio – design studios are also used for individual work, because the student benefits from being in an open environment around other students.

Table 2.1 provides a glimpse of the variability involved in each institutions' studio implementation. This table confirms that there is some shared understanding of core studio education values, such as reflective practice, critiques, and co-located students. However, specific implementation details are often vague; for example, how reflective practice is supported.

Evaluating software studios to determine if they are in fact a studio appears to be an under-developed area of research. Despite not knowing whether a given studio is indeed a studio, there are evaluations that compare an implementation of a software studio to its traditional (non-studio) counterpart, to determine its effectiveness at teaching the course. One such study states that the *"studio-based learning model offered significant benefits to students in terms of both course content mastery and programming achievement"* (Hendrix et al., 2010, p.509). Regardless of whether it is a studio, it is apparently beneficial to the students.

3 Understanding Studios: The Studio Framework

After a review of the literature in the previous chapter for architecture, design and art studios, no widely shared definition for studios was found. However, there were various definitions, descriptions, or even specific features of studio education that were common. To summarise some of the various definitions, studios are spaces where students are co-located and coached by teaching staff, they will potentially participate in some of the various forms of critique, and are observed as utilising reflective practice. However, it is apparent that an understanding of their complex nature resides in tacit knowledge (Löwgren and Stolterman, 2004) – knowledge that has had the opportunity to be passed down over centuries. This chapter therefore directly focuses on the research question:

RQ1) What is a studio?

Without a clear definition of "studio", software studios cannot be properly evaluated for their impact on student learning, nor can best and worst practices be shared between those who run studios. There is obviously a vision out there for creating a software studio, yet, with the lack of a definition or framework to follow, the authenticity of attempted software studios is brought into question. This means when people try to create a studio they often discover things for themselves. This is OK, but to make the studio move to a mainstream role where the benefits can be more widely felt, we need a better understanding of the ingredients that make a studio. This will allow such things to be transformed or managed to suit

different scenarios, or indeed the fact that it is a different discipline. But before we can do that, we first need to develop an understanding of the key factors at play within a studio, because *"there is a danger that an improperly administered studio will degenerate into yet another project course"* (Tomayko, 1991, p.301).

This chapter presents an exploration into achieving a better understanding and definition of the term "studio" and its use in education, and was previously published in ICSE'13: "Studios in Software Engineering Education: Towards an Evaluable Model" (Bull et al., 2013). This was achieved by conducting 15 interviews, with participants from arts and design disciplines, which were then analysed using qualitative methods. Design is an umbrella term which includes a wide variety of disciplines, such as industrial design, product design, service design, and graphic design, to name a few. The interviews involved a variety of people with a range of experiences in the use of studios, reflecting on their individual perspectives of studio education and their multi-faceted understanding of the environments. This chapter presents the results of the analysis of the interviews, and through that, offers an initial framework of studio education. The framework provides a set of criteria upon which to evaluate the authenticity and extent of a studio. The results suggest that there are many intertwined aspects that define studio education, but it is primarily the people and the culture that make a studio. On the other hand, the most contentious aspect of the studio is the use of digital technology. It can adversely affect the use of a studio by undermining elements in the framework. However, when properly recognised, digital technology can be very beneficial in a studio.

To explore the benefits of a software studio for software engineering students it is essential to first determine what a studio is.

3.1 Methodology

To achieve a better understanding of studios, leading to a framework to evaluate against, multiple peoples' perspectives and experiences of studio education would

be needed. 15 interviews were conducted with a range of designers, architects and artists – the interview participants would help inform a framework of what constitutes studio education.

The outline of the approach to this investigation is based on Grounded Theory of interview transcripts, originally proposed by Glaser and Strauss (1967), and is used to create a 'theory' from gathered data. It is also said to be best suited to understanding *"the process by which actors construct meaning out of intersubjective experience"* (Suddaby, 2006, p.634). This tactic is utilised because of the exploratory approach to understanding studios across multiple disciplines, essentially allowing a theory or hypothesis to emerge from the interview participants. The process used is described below.

As this research is about determining how people view studios and work within them, it was expected that different responses would be gathered from a variety of people as the interview participants come from various backgrounds and even different disciplines.

3.1.1 Interview Participants

A call for interview participants was sent out via email to Design and Architecture departments in several universities in the UK, asking for people with experience with studio education; no software engineers were interviewed during this investigation as the purpose was to understand the studios in the design disciplines first. One participant (A_1) was approached outside of this call for participation and directly asked to interview based on her experiences with multiple studios. 15 participants were interviewed and the respondents are used anonymously.

In the call for participation it was requested that the participants had experience in at least one of the following areas:

- Were educated in a studio
- Have taught others in a studio

All of the respondents had experience teaching and being taught in a studio. The

Table 3.1: Interview participants

Participant[†]	**Discipline**	**Studio Experience**	
		Educated	**Teaching Role**
A_1	Fine Art	✓	✓
A_2	Industrial Design	✓	✓
A_3	Industrial Design	✓	✓
A_4	Graphic Design and Art	✓	✓
A_5	Landscape Architecture	✓	✓
A_6	Product Design	✓	✓
A_7	Digital Art	✓	✓
A_8	Product and Service Design	✓	✓
A_9	Product and Spatial Design	✓	✓
A_{10}	Product, Ceramic and Industrial Design	✓	✓
A_{11}	Computer Aided Architectural Design and Visualisation	✓	✓
A_{12}	Graphic and Service Design	✓	✓
A_{13}	Service Design	✓	✓
A_{14}	Product and Industrial Design	✓	✓
A_{15}	Industrial Design	✓	✓

[†] Participants in this set of interviews are labelled as 'A' with a subscript number to differentiate between the 15 participants (e.g. 'A_{13}') – these are also used as a unique identifier throughout this thesis. Quotations from these informants will also be attributed to them following a similar citation style, e.g.: *"the physicality of ideas is also integral to the studio environment, in my mind"* (A_{15}).

participants also had a variety of experiences with different disciplines. Table 3.1 shows the breakdown of the participants, their experience, and lists their primary discipline(s).

The number of participants selected for interview was to be decided whilst the interviews and analysis were ongoing, using the concept of 'saturation' (Corbin and Strauss, 2008). The point of saturation is described as *"when no new data are emerging"* (Corbin and Strauss, 2008, p.143), and is further described as the point when categories are no longer being developed – categories are described in more in section 3.1.3. No further participants were needed to be recruited after participant A_{15} as the research reached saturation within the current list of participants; by participant A_{11}, the analysis of the subsequent interviews did not produce any new categories.

The final list of participants consisted of six universities, but a total of eight separate departments; for example, a participant may be from Design at a university, and another participant may be from Architecture at the same university. The participants had a variety of experience and positions; for example, some of the participants were lecturers/professors (A_2, A_4-A_{10}, A_{12}, A_{14}-A_{15}), studio tutors (A_1, A_3, A_{11}), and a course director (A_{13}). Although all participants currently work in UK universities, several of them have experience working and learning in studios abroad including USA, Portugal, Italy and Thailand. All of the participants agreed to anonymity prior to taking the interview. This allowed participants to speak freely, even though no sensitive issues were on the agenda.

3.1.2 Interview Design

The interviews were designed as semi-structured interviews, with no prescribed time limit (on average lasting approximately 45 minutes), as a tactic to gather richer information from the participants. As it was semi-structured, the participants could lead the conversations and discuss the aspects they thought were of significance.

As the interviews were semi-structured, there was no concrete list of questions, but there was a list of topics and starter questions (listed below). Some of these questions were intended to get the participant to describe an environment, and others were developed based on background reading (to compare perspectives of the literature and the interview participants); the rest of the questions were based on bouncing off of the participants responses. The starter topics and questions were:

- Outline your experience with studios
- Describe your studio(s)
- Describe a non-studio*
- Think of a specific project; how was it helped or hindered by using a studio?
- What is the ideal vs. reality of a studio?
- Why are studios good?
- Why are studios bad?
- What is the most important aspect of a studio?

As the interview process went on, an additional starter questions was included as it was implicitly discussed and resonated with all of the earlier interview participants: *"Describe a non-studio"*. The interviews were conducted on a variety of mediums including face-to-face and phone, but largely using video-calling software.

3.1.3 Analysis

The interviews were audio recorded and later transcribed. The transcripts were analysed in batches (three batches of five sequential participants, i.e. participants A_1-A_5, A_6-A_{10}, and A_{11}-A_{15}), with each batch analysed and integrated into a singular output – importantly, this provided an opportunity to reflect on the output of each previous batch.

The analysis went through a process of coding and categorisation (Glaser and Strauss, 1967), a method of identifying unique thoughts or ideas in text and later grouping them around common themes. Each transcription underwent three passes of analysis (identifying codes, grouping them into concepts, and finally grouping the concepts into larger categories) to distil the information into a presentable form: the categories in the next section. The first pass of each transcription was dedicated to identifying and listing codes - essentially, a code was anything that stood out, and they would emerge from the data (no pre-existing list of codes was used). Codes were identified by looking for keywords, phrases, or more holistically, ideas discussed during the interviews. A sanity check for each code was to ask whether it could form part of an answer to the broad questions 'What is a studio?' or 'What affects a studio?'. A technique used during this analysis was to perform constant comparisons of coded passages of text, reflecting on the generated codes and also previously coded passages of text to ensure consistency across the codes and text, and later the concepts and categories.

The second pass of the analysis went over each of the codes and grouped them into concepts. Each code was judged as to whether it belonged in any of the previously generated concepts, or whether it warranted being separated out as a new concept. The list of codes was subsequently checked and reflected on several times throughout the analysis as part of the process of constant comparison to ensure that each individual code belonged to the most appropriate concept.

The final pass in the analysis grouped the concepts into categories (or themes), culminating in the creation of a framework for studio education. This pass also benefited from the constant comparison. The categories in this framework are presented in the next section. The concepts within each category are presented in a succinct list alongside their associated category, referred to as "Parameters".

Finally, as part of the reflective side of the analysis, concepts that were discussed by only three or fewer participants were not included in the final framework as they are not deemed representative of the group of interview participants.

3.2 The Studio Framework

Interviewing participants with a wide and extensive range of experiences within many studio settings led to nine unique categories. Each category reflects a factor or property which was reported to affect the studio experience in some way. The studio framework is briefly presented in Table 3.2, and described in greater detail in the following subsections. Table 3.2 shows the categories and parameters that resulted from the analysis, and indicates how many participants discussed that parameter under the heading 'Occurrences'. A matrix of the framework's categories and parameters against the interview participants is also presented in Table 3.3.

The categories in Table 3.2 are presented in no particular order, and are discussed in more detail in a contiguous list below. The parameters are an accessible and succinct list of concepts that make up their associated category. Note that two categories were omitted from the framework as they are not essential in the creation and running of a studio: 'Defining the studio' and 'Digital technology' are anomalous categories, explained in more detail in a later section ('Further Analysis', section 3.3), that do not contain features of a core studio. However, they get an honourable mention because they were discussed significantly during the interviews. To address the inherent complexity involved in understanding the categories, each is presented with information and quotes from the interviews to help contextualise the issues they represent. For example, how important or dominant that category was to the interview participants, if there were any particularly important relationships with other categories, and further, what was considered good for the studio and what was considered detrimental. The informant's quotations used throughout this section are representative of the sample's responses, despite only a selection being used, as they provided succinct responses during the interviews.

Table 3.2: The studio framework

Categories	Parameters	Occurrences
Physical environment	Open environment	14
	Reconfigurable furniture (Flexible Space)	14
	Students control aesthetic factors	5
	Shared spaces	15
	Individual spaces	8
	Social spaces	7
	Private spaces	7
Facilitation of studio	Studio belongs to the students	8
	Staff do not dictate use of space	6
	24 hour access	7
	Food and drink allowed	6
	High availability of staff	5
	Small group size (about 10)	5
Modes of education	Switch approach based on activity	13
	Mentoring/coaching	11
	Peer-learning	12
	Impromptu teaching	5
Awareness	Visual work	13
	Displaying work	13
	Visual history of progress	6
	Easily observe other people's work	12
	Social interactions	8
Critique	Direct feedback	12
	Develop ideas	11
	Multiple formats (formal/informal etc.)	11
	Peer-coaching	10
Culture	Sharing	10
	Social	14
	Treated like second home	6
	Good work ethic	10
	Peer-support	6
	Serendipity	8
Individual's characteristics	Personalisation of space	9
	Private and quiet spaces	10
Inspiration	Proximity to other people	13
	Relevant available media	5
	Library of liked/fun things	5
	Playful space	5
Collaboration	Impromptu collaborative spaces	6
	Supporting equipment	7

Table 3.3: The studio framework's participant matrix

Studio Framework Parameters		**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**A10**	**A11**	**A12**	**A13**	**A14**	**A15**
Physical environment	Open environment	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Reconfigurable furniture (Flexible Space)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Students control aesthetic factors	✓		✓						✓		✓	✓			
	Shared spaces	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Individual spaces	✓	✓	✓	✓		✓		✓		✓			✓		
	Social spaces	✓		✓			✓		✓		✓		✓			✓
	Private spaces	✓		✓			✓			✓			✓	✓		✓
Facilitation of studio	Studio belongs to the students	✓	✓	✓	✓		✓		✓	✓	✓					
	Staff do not dictate use of space	✓	✓				✓		✓			✓		✓		
	24 hour access	✓	✓	✓					✓			✓		✓		✓
	Food and drink allowed	✓		✓					✓	✓	✓		✓			
	High availability of staff	✓	✓			✓				✓	✓					
	Small group size		✓						✓	✓	✓		✓			
Modes of education	Switch approach based on activity	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓		✓
	Mentoring/coaching	✓		✓	✓	✓	✓		✓	✓	✓			✓	✓	✓
	Peer-learning	✓	✓	✓	✓	✓	✓		✓		✓		✓	✓	✓	✓
	Impromptu teaching				✓	✓						✓		✓	✓	
Awareness	Visual work	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓	✓
	Displaying work	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓	✓
	Visual history of progress	✓	✓	✓			✓		✓					✓		
	Easily observe other people's work	✓	✓	✓	✓	✓		✓	✓	✓	✓			✓	✓	✓
	Social interactions	✓	✓	✓				✓		✓				✓	✓	✓
Critique	Direct feedback	✓	✓			✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
	Develop ideas			✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓
	Multiple formats	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓		✓
	Peer-coaching			✓	✓	✓	✓		✓		✓	✓		✓	✓	✓
Culture	Sharing	✓	✓	✓		✓			✓		✓	✓		✓	✓	✓
	Social	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Treated like second home		✓	✓		✓			✓	✓	✓					
	Good work ethic	✓	✓	✓	✓	✓		✓	✓		✓	✓		✓		
	Peer-support				✓		✓		✓		✓	✓		✓		
	Serendipity	✓		✓		✓			✓		✓		✓	✓	✓	
Individual's characteristics	Personalisation of space		✓	✓	✓		✓		✓		✓	✓	✓	✓		
	Private and quiet spaces	✓	✓	✓	✓	✓		✓				✓	✓	✓		✓
Inspiration	Proximity to other people	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓
	Relevant available media		✓		✓		✓		✓							✓
	Library of liked/fun things	✓	✓	✓	✓									✓		
	Playful space			✓	✓		✓				✓					✓
Collaboration	Impromptu collaborative spaces			✓		✓	✓		✓				✓	✓		
	Supporting equipment						✓		✓		✓		✓	✓	✓	✓

3.2.1 Physical environment

This category contains anything regarding the physical environment, surroundings and equipment; for example, furniture, layout and location etc. All of the interview participants discussed how the physical environment was important for a studio, but also how it was not the most important aspect, with one person remarking *"the studio is created by the culture, not by the furniture"* (A_9). When asked to describe a studio, another participant said *"the only sort of basic, generic kind of description I can think of is that it's space... I've seen it manifest in many different forms in different places"* (A_7). Here the studio is described almost as if it was a blank canvas, with the only constant variable being a physical 'space'.

The participants generally had a positive discussion when talking about the physical environment. One of the most important and commonly talked about aspects was the flexibility of the space (A_1-A_{14}), such as moving furniture, essentially saying that students should not have their activities limited by the rigidity of the space, as some setups are better suited to different activities. Participant A_6 describes spaces that do not do this as non-studios, which are *"planned in terms of efficiency of space, rather than supporting efficient activities"*; an example of why this occurs is because institutions fit *"as many people in there as you can and there is no space for communal activities"* (A_6). Another talked about aspect (A_1, A_3, A_9, A_{11}-A_{12}) was aesthetic factors such as lighting, heating, but more importantly, control over these aspects. Some other aspects that were discussed were the provision of shared space (A_1-A_{15}), individual space (A_1-A_4, A_6, A_8, A_{10}, A_{13}), the necessity for an open environment which can lead to better proximity to others (A_1-A_4, A_6-A_{15}), and social spaces (A_1, A_3, A_6, A_8, A_{10}, A_{12}).

Although most of the conversations were positive when discussing the physical environment, there were some negative and detrimental aspects that got some attention, such as openness of space. Although openness of space was generally desired amongst the participants, the participants frequently pointed out potential issues with privacy, security and disruptive noise; the suggested solution to

these problems was the use of private space (A_1, A_3, A_6, A_9, A_{12}-A_{13}, A_{15}). One participant claimed that meeting rooms, not just private space, are essential (A_3).

3.2.2 Facilitation of studio

This category contains everything relating to staff of a studio and the rules put in place to govern the space. Rules may have been set out by a facilitator (e.g. lecturer) or by the institution (e.g. university or department).

Most interview participants agreed that the studio belongs to the students (A_1-A_4, A_6, A_8-A_{10}), and the staff do not (strongly) dictate to the students how they should use the space (A_1-A_2, A_6, A_8, A_{11}, A_{13}). When the students feel that they have control over the environment they will likely use it for activities that are not work related – which is fine. This relates to 24 hour access, which several of the participants said was necessary for a comfortable studio (A_1-A_3, A_8, A_{11}, A_{13}, A_{15}). Restricted access to a studio is bad, as it stops the students from using the space as they want to, for example participant A_2 said *"you can sleep there, and I slept in my studio so many times, in my student year"*. Similarly, students are allowed to eat and drink in the studio too (A_1, A_3, A_8-A_{10}, A_{12}). Another aspect that some of the interview participants thought important was a high level of availability of staff (A_1-A_2, A_5, A_9-A_{10}). Someone always needs to be accessible to provide assistance if required. Participants A_2 suggested that certain timetabled slots were used just for the facilitators to be in the studio, walking around giving advice and getting a general feel of where all the projects are. Some of the participants (A_2, A_8-A_{10}, A_{12}) felt strongly about the size of the group (cohort) within a studio; depending on the size of the course, or year groups, you may need to split them up into multiple groups or studios.

Lastly, a facilitator could destroy a studio experience; devolving it from a studio to an open environment by imposing rules, reducing serendipitous interactions (like walking around and talking) and intruding on other forms of freedom that a studio can provide. Furthermore, participant A_{15} discusses the difficulties with setting

up a studio: *"I don't think it's easy to set studios up, these days; not certainly within a university environment"*. This is backed up with comments about how resistant universities are to providing space to be permanently used for a small number of people. Some of the participants talked about an ideal cohort group size for students being approximately 10-20 (A_2, A_8-A_{10}, A_{12}), and larger courses (for example, 100 students) would split their students into smaller groups. However, participant A_8 stated that institutions vary on the number of students in their studio, where it is not uncommon to have larger group sizes.

3.2.3 Modes of education

This category is about the specific learning activities that are performed in a studio. There are a range of activities that will be carried out in a studio, but most of them will be specific to the subject or even discipline of that studio. However, the interview participants did discuss some common activities, stating that the students will switch between approaches based on the task at hand (A_1-A_7, A_9-A_{13}, A_{15}). An important mode of learning is the mentoring or coaching (Schön, 1987) of the students, which the participants agreed with (A_1, A_3-A_6, A_8-A_{10}, A_{13}-A_{15}), where staff take on the role of coach for the student. Another frequently discussed aspect in this category was peer-learning, with most participants talking about it in one form or another (A_1-A_6, A_8, A_{10}, A_{12}-A_{15}). Peer learning surfaced in several forms, ranging from receiving critique from your peers during a group critique, to a form of peer-coaching. The importance of peer learning can be seen in participant A_8's comment: *"A lot of learning happens when you are talking to your colleagues, to other students, in information conversations with the lecturers, not necessarily always lectures."*

Another approach discussed with some of the participants (A_4-A_5, A_{11}, A_{13}-A_{14}) is the use of impromptu teaching. This technique benefits from being in a studio, as the lecturers are expected to be in the studio more often than other courses. Impromptu teaching can happen at any time, but an important trigger may be

that the lecturer walks around and sees a common discrepancy in the students' knowledge, or maybe even a student that the lecturer wants to use as an example to demonstrate something. Of course, another mode of working that was discussed was the solo approach.

3.2.4 Awareness

This category is all about the awareness that an individual will have of his or her surroundings, other people and their work, anything visual (especially visual work) and what this affords them in their studio experience. Several interview participants claimed that visual work in a studio is very important (A_1-A_6, A_8-A_{10}, A_{12}-A_{15}), with participant A_3 saying *"visual communication is one of the most important things"*, with this key skill being developed throughout the studio experience. Visual work is very important to the studio's ecosystem for many reasons: it allows students to see at a glance what others are doing (A_1-A_5, A_7-A_{10}, A_{13}-A_{15}), when displayed it enables more impromptu interactions with staff or students and aids better reflection (A_1-A_6, A_8-A_{10}, A_{12}-A_{15}), it allows students to see how much work has been done and show a visual history of progress for better longer term reflection-on-action (A_1-A_3, A_6, A_8, A_{13}), and it allows other people to see what you have done – this would lead to a shared awareness amongst all occupants of the studio. When you get stuck, you can easily go and see what other people are doing; making easy connections between all of the work that is displayed in the studio. When things are placed up on a wall or left lying around the studio, they become a visual representation of the process, past work and progress. An open environment allows teaching staff to easily observe similarities or common issues amongst a group, which ties in with the ability to perform impromptu teaching.

As studios are social open environments, there is more communication amongst peers, leading to a greater awareness of other people and therefore their work (A_1-A_3, A_7, A_9, A_{13}-A_{15}). Furthermore, shared spaces within the studio, such as

large desks, are used for sharing group work, explicitly improving the awareness of groups. Awareness can be reduced if studio spaces are not permanently assigned, as participant A_{14} explains: *"if you go to a lot of other places all the students will be sticking all their latest sketches on the wall and there might be loads of inspirational stuff"*. However, at his institution the room is not permanently assigned *"I guess, our students don't own their space. Within the course of the day there might be three sessions in there, first years, second years, and third years, all in the same space, using the same desk"* (A_{14}), so the room is cleansed after each session in preparation for the next group.

3.2.5 Critique

This category covers anything about the culture and use of critique (see background literature on critique in studios in section 2.2.3). All of the interview participants discussed some form of critique, and think it is important in the studio. This category could be argued as being a subset of the 'Modes of education' category, however, the topic of critique was discussed often by all of the interview participants, stating that it is an important part to the studio experience.

A critique is a mechanism with several benefits. The most prominent according to the interview participants is direct feedback (A_1-A_2, A_5-A_{10}, A_{12}-A_{15}), which can be continuously given. Critique also improves the students ability to communicate thoughts and ideas, is used as a vehicle to teach students how to contribute to the discussion of other peoples' work, and helps prepare students to present ideas and handle the responses. There are various types of critique, including formal/informal and group/desk/peer critique (see section 2.2.3), and many of the participants discussed their use of multiple formats in their studios (A_1-A_6, A_8-A_{10}, A_{13}, A_{15}). Participant A_4 shared the idea that the frequency with which you perform group critiques is proportional to the size of the class. A common discussion was the use of critiques as a mechanism to develop the students' ideas (A_3-A_8, A_{10}-A_{11}, A_{13}-A_{15}).

The location of a critique can be a sensitive issue, as they can be disruptive and difficult to hold in open public spaces where not everyone is involved in the critique itself. This is often countered by booking out a dedicated room for formal critiques. However, in timetabled sessions where lecturers are in the studio walking around, the lecturer gives feedback (informal critique) about what's good and what could be improved. Several of the interview participants stated that critiques can have adverse effects on students, as students grow attachments towards their work which then could potentially receive heavy negative criticism. Participant A_{12} stated that crits can sometimes turn into a *"pissing contest"*, whereby each member of teaching staff will essentially try to outdo one another by trying to find something wrong with a given piece of work. *"It really is just a 'I know more than you do; I'm going to disagree...' and you think 'what is being achieved here?'"* (A_{12}). This also resonates with Koch et al. when they discuss 'punishing critiques', described earlier in section 2.2.5: *"Without providing any preparation or guidance before critiques, there is nothing to ensure that the assessment environment will be a healthy and constructive experience for students"* (Koch et al., 2002, p.14). One possible solution, depending on the context of the critique, is offered by participant A_5 where advice or criticism is offered privately after a session. In the run up to a critique session, some students can get motivated to produce a similar volume or quality of work when compared to other students, which is enabled, for example, by being in an open environment with other's work visible.

The studio teaches the students to be critical of each other's work, which becomes part of their culture. The studio is a great platform for sharing ideas because of this culture of informal critique and enables the frequent use of peer-coaching (A_3-A_6, A_8, A_{10}-A_{11}, A_{13}-A_{15}). Participant A_5 describes how a formal peer-coaching process was used, which could also reinforce a culture of critique. He describes a weekly organised activity called "peer review": *"students can self select a group and they have to talk about their work in front of other students and these students have to comment on their work and this is all logged in a log book*

and the students then have to present that as part of their summative assessment at the end".

Participant A_{12} also described how critiques are not always as productive as others may think, because it is *"a method that is used because you've got the big space"*. He went on to describe a typical experience of a critique session (jury): *"let's get all the students together, let's stick all the work up on a wall and let's spend the whole day going through it and it isn't as productive as people think it is. It isn't realistic and it's not a good learning experience. It's not used in industry. It's just tradition"* (A_{12}). However, his experiences of critiques appear to be in the minority compared to the rest of the interview participants. It is clear that a critique can get out of hand, potentially with the the student's experience not a priority, but this can be properly managed.

3.2.6 Culture

This category contains all cultural aspects, group aspects, group behaviours and other encultured dynamics. This is one of the more dominant categories to come out of the analysis. Several of the interview participants identified sharing as an important aspect in the studio (A_1-A_3, A_5, A_8, A_{10}-A_{11}, A_{13}-A_{15}), contributing to the open environment by sharing equipment, spaces, and more importantly, sharing ideas. There were generally similarities with the participants' comments to what others have described: Studio culture often includes *"Late nights, exciting projects, extreme dedication, lasting friendships, long hours, punishing critiques, unpredictable events, a sense of community, and personal sacrifice"* (Koch et al., 2002, p.3).

Another aspect that the participants talked about was the social side (A_1-A_4, A_6-A_{15}), as the studio is considered not just as a work environment, but also as a social environment. There could be communal activities organised within the studio and talking is encouraged, as this apparently improves creativity which *"often happens in an informal way like in a conversation in a relaxing atmosphere"* (A_8); this

resonates with others who have said *"Software engineers rely on social and casual contact for important communication"* (Jazayeri, 2004, p.xxii). Similarly, another aspect that was discussed frequently was that the studio is treated like a second home (A_2-A_3, A_5, A_8-A_{10}) with students eating, sleeping and potentially having parties! Participant A_{10} discussed how parties were frequently organised, once every two months: *"to foster culture there is [...] parties in the studio [...] we have a DJ night where it's organised by the course director, [...] where staff, students and alumni come together"*. This also provided networking opportunities too.

Moving back to the work side of the culture, some of the participants (A_1-A_2, A_4, A_6-A_7, A_{11}-A_{12}) recognised that you could make a lot of noise in the studio. This would largely not bother some of the participants (A_2, A_4, A_7, A_{12}) as that is just the way of the studio, with an example noise mitigation technique discussed being the use of headphones (A_1, A_6) or moving to a private/quiet space (see section 3.2.7 below); however, other participants noted that all students, perhaps like in any environment, would develop a good work ethic (A_1-A_5, A_7-A_8, A_{10}-A_{11}, A_{13}) and become sensitive to others that they share the environment with and performing a lot of work outside of any scheduled sessions because the studio has a large amount of self-directed work. Another aspect is serendipity, which was discussed as happening often in studios (A_1, A_3, A_5, A_8, A_{10}, A_{12}-A_{14}) – the open and visual environment provides plenty of opportunities to stumble upon something that someone else is looking at or working with. Peer support is another aspect that was prominent in the interviews (A_4, A_6, A_8, A_{10}-A_{11}, A_{13}), as it was discussed as being a part of the students' attitudes towards each other; the informality of the environment means that students can support each other through explicit activities, such as peer-coaching, or even socially. The studio is said to be filled with students that support and motivate each other.

One last discussion point relating to culture was that empty studios are not as useful as a studio with occupants: *"studios can be really bad if people aren't in*

them because you're on your own a lot [...] the one thing about studios is they've got to be lively, I think, to be creative" (A_4). One of the purposes of a studio is about externalising thinking, for example through critique or collaboration, and is also benefited by the use of visual work. A studio is a mode of communication.

3.2.7 Individual's characteristics

This category contains aspects that relate to an individual's personality, their attitude in or towards the studio, and their social and personal well-being. Through the analysis of the interviews, and as observed by the participants themselves, each discipline, topic, and even person will value different aspects of the studio. Some people inherently prefer group activities, whilst others prefer solidarity. A studio should try and cater for both, even though it leans towards group or social environments. Sometimes there is a need for privacy or quiet, and some participants (A_1-A_7, A_9, A_{11}-A_{13}, A_{15}) suggested the need for dedicated rooms or spaces to cater for this – open environments have the possibility of becoming noisy and disruptive.

One sensitive issue that was discussed was how some students are uncomfortable essentially being on display, especially if they are new to such environments. Privacy can be an issue for some people, and the open environment does not help this matter, but it was discussed how learning to cope with this is more beneficial than providing permanent private space. Participant A_4 described feelings of being judged by others as they compared her work to someone else's on display. At the same time it was said to be a motivating factor as well, potentially making students feel good about their work when others enquire about it.

One last factor for this category is the importance of personalisation of space (A_2-A_4, A_6, A_8, A_{10}-A_{13}); for example having students displaying work where they want, leaving equipment on their desk and collecting inspirational items. This leads to a better sense of ownership of the space. However, messy people can possibly overwhelm the space and disrupt others. The interview participants

usually referred to personalisation of space as the process of turning a space into somewhere that resembled a second home, but others also described it as a process of making the space more functional: *"it's not about having the side of your computer with 15 little fluffy animals on it. It's about personalising in terms of making it work and making it be appropriate for your work"* (A_6).

3.2.8 Inspiration

This category contains anything about inspiring ideas and gaining inspiration. Many of the interview participants felt strongly about the need for inspiration and inspirational materials. Inspirational materials are generally context sensitive to the type of work being done, for example a furniture designer would typically have more items related to furniture. The core concept that participant A_2 discussed was that ideas require stimulation, and it is often referred to in the context of visual inspirational items around the studio (previous work, posters etc.); studios embrace this *"quite heavily"* (A_2). A feature of the studio that some of the participants with a Design or Art background sometimes talked about (A_1-A_4, A_{13}) was a *"library of things"* (A_2) or room of inspirational items. Participant A_2 also gave the example of seeing and buying things and then placing them in this library, building up a collection of items that inspired them. Another participant (A_4) claimed that a person that had less time immersed in similar environments, lacking time to build a store of tacit knowledge from these items, is less capable. Also, many participants (A_2, A_4, A_6, A_8, A_{15}) discussed the usefulness of having a variety of media on-hand in the studio, such as design magazines, to provide inspiration.

Inspirational studios were also said to be playful, with some participants (A_3-A_4, A_6, A_{10}, A_{15}) mentioning that playful spaces are good for generating ideas, and that the space would benefit from having *"toys"* (A_3). The toys could be relevant to the work, or something that the individual liked. This comes back to what one participant said about informal spaces being good for creativity.

One last aspect about inspiration was discussed as benefiting from proximity to

other studio users (A_1-A_5, A_7-A_{10}, A_{12}-A_{15}). Having other people in the studio allows people to frequently and easily share and discuss ideas and their progress.

3.2.9 Collaboration

This category covers all collaborative activities, not simply group activities taking the form of division of labour. As stated in other categories, the studio's usefulness relies upon having students occupy the space; because of this, most interview participants expected that collaboration would take place in the studio. One participant (A_6) said that the studio was a hub for collaboration and that the studio experience is diminished when people are not very collaborative. Another participant even claimed that an environment is not a studio and just an open space when its occupants work as individuals all the time.

An important factor of collaboration in the studio is the serendipitous and impromptu nature that collaboration can take. Firstly, the studio offers opportunities for occupants to collaborate with others that they may not have thought about, due to the open nature of the space. Secondly, the studio needs to offer impromptu collaborative spaces (that does not need booking or setting up), as some of the participants suggested that collaboration can take place at any time (A_3, A_5-A_6, A_8, A_{12}-A_{13}).

One last smaller point that several participants mentioned was the need for easy access to equipment that would support collaboration (A_6, A_8, A_{10}, A_{12}-A_{15}); the examples they gave were stationery, such as pens, paper, sticky notes etc. Other facilities that they discussed to improve collaborative activities were digital and online services, primarily for when collaborators could not come together, not as a replacement for face-to-face interactions.

3.3 Further Analysis

After the studio framework was identified, relationships between the categories and aprameters were identified – these are shared below in section 3.3.2. Further, the analysis of the interviews provided some anomalous categories, which are described below in section 3.3.3.

3.3.1 Necessary and Desirable Framework Parameters

The interview participants discussed a variety of aspects which they feel contribute to studio education, as presented earlier in Table 3.2 and Table 3.3. In an attempt to distinguish between essential and non-essential parameters, the studio framework is presented again on the following page in Table 3.4 with the highlighted parameters that fell above a two-thirds majority threshold (11 or more participants).

These highlighted parameters were not discussed by the participants as being explicitly essential or non-essential. However, the frequency with which they are discussed across the interview participants could be argued that they are important to a studio environment. Parameters that fall below the threshold could be considered desirable, but not essential.

Following Table 3.4, we can see that three categories fall entirely into the non-essential categorisation, as none of their parameters were discussed by the majority of participants: 'Facilitation of studio', 'Individual's characteristics' and 'Collaboration'. All of the other categories have at least one parameter that was discussed by the majority of participants.

Aspects that the participants did explicitly state as the most important aspect are presented below in section 3.3.4.

Table 3.4: The studio framework's dominant parameters

Categories	Parameters	Occurrences†
Physical environment	Open environment	14
	Reconfigurable furniture (Flexible Space)	14
	Students control aesthetic factors	5
	Shared spaces	15
	Individual spaces	8
	Social spaces	7
	Private spaces	7
Facilitation of studio	Studio belongs to the students	8
	Staff do not dictate use of space	6
	24 hour access	7
	Food and drink allowed	6
	High availability of staff	5
	Small group size (about 10)	5
Modes of education	Switch approach based on activity	13
	Mentoring/coaching	11
	Peer-learning	12
	Impromptu teaching	5
Awareness	Visual work	13
	Displaying work	13
	Visual history of progress	6
	Easily observe other people's work	12
	Social interactions	8
Critique	Direct feedback	12
	Develop ideas	11
	Multiple formats (formal/informal etc.)	11
	Peer-coaching	10
Culture	Sharing	10
	Social	14
	Treated like second home	6
	Good work ethic	10
	Peer-support	6
	Serendipity	8
Individual's characteristics	Personalisation of space	9
	Private and quiet spaces	10
Inspiration	Proximity to other people	13
	Relevant available media	5
	Library of liked/fun things	5
	Playful space	5
Collaboration	Impromptu collaborative spaces	6
	Supporting equipment	7

† 'Occurrences' which are highlighted are above a two-thirds majority threshold, that is they are discussed by 11 or more interview participants.

3.3.2 Relationships Between Parameters

Some parameters in the studio framework (Table 3.3) inevitably have relationships with other parameters, some more so than others. This is largely the parameters that were discussed by the majority or all of the participants. These relationships were identified using the participant parameter matrix (Table 3.3), and referring back to the interview transcripts. Also, when looking for relationships between parameters across the participant profiles (sharing discipline experiences), no relationships emerged beyond the parameters that were already identified earlier as being discussed by the majority of participants (see section 3.3.1). This could in part be because of the variety of disciplines, and the comparatively low number of participants interviewed; for example, there were only two participants with experience in graphic design, and one with spatial design. Some participant profiles were larger (e.g. five product design, five industrial design, and three service design), but despite this, their discussed parameters did not drastically differentiate themselves from one another.

Following are relationships that have been identified between parameters in the studio framework.

Displaying visual work:

Within the 'Awareness' category, the participants that discussed 'Visual work' (A_1-A_6, A_8-A_{10}, A_{12}-A_{15}), and its role in visually designing, also always discussed publicly showing the work ('Displaying work') to garner various benefits. There is also a high co-occurrence of those parameters with the parameter 'Easily observe other people's work'. Whilst these parameters could occur independently, they generally did not; for example, visual work does not need to be put on display, and a room may not to cater towards allowing the students to easily see each others work if, for example, it is not an open environment.

Culture of Critique:

Another relationship in the parameters is found in the 'Critique' category. This occurred with participants that discussed one of the purposes of critique as to 'Develop ideas', and not just assessment at the end of a project or problem (A_3-A_8, A_{10}-A_{11}, A_{13}-A_{15}). Of those, most of them also discussed the use of 'Multiple formats' of critique and 'Peer-coaching' (A_3-A_6, A_8, A_{10}, A_{13}, A_{15}). The use of multiple formats of critique is useful for creating a culture of critique, having students constantly question and justify problems and their solutions.

Students Control Flexible Spaces:

Participants frequently discussed how students are able to 'Switch approach based on activity' (A_1-A_7, A_9-A_{13}, A_{15}). Switching freely between different learning approaches is made easier with the availability of varied and flexible spaces; for example, instantly switching from a solo activity to an ideation space. Also, the solo mode of working needs to provide adequate provisions for a quiet environment, allowing the students to get on with their work.

Counteracting Noisy Environments:

Those that discussed the use of social spaces in the studio also often discussed the need for 'Individual spaces' and/or 'Private spaces' (A_1, A_3, A_6, A_8, A_{10}, A_{12}, A_{15}). Noise can be a problem with shared spaces, dependant on the number of students that occupy it at any one time. Headphones were also sometimes mentioned as a solution to dealing with the sometimes inherently noisy environments (A_1, A_6, A_9).

Room Layouts to Support Awareness:

The space provided to the studio can potentially reduce awareness of what is going on in the studio, for example if it has a bad layout, or there are other physical attributes that isolate individuals. Participant's that discussed how the studio

benefits from students being able to 'Easily observe other people's work' are often correlated with other parameters such as the benefits of 'Displaying work', and awareness of 'Social interactions'. The participants discussed how these benefit from occurring in an 'Open environment' and also 'Shared spaces'. Another issue is that there may be limited wall space to display certain types of work, especially in environments that are not dedicated to the studio.

Serendipitous Critique:

'Serendipity' in a studio was a common discussion point for the interview participants (A_1, A_3, A_5, A_8, A_{10}, A_{12}-A_{14}). Serendipitous awareness is helped by the open and visual environment, and a culture of sharing, as it provides plenty of opportunities to stumble upon something that someone else is working on by simply looking across the room or even walking back to a desk with a coffee. With this awareness follows the opportunity for the students to engage in 'Peer-coaching' and to 'Develop ideas'.

Personal Space:

Of the participants that discussed how the 'Studio belongs to the students', several of them (A_2-A_4, A_6, A_8, A_{10}) also discussed the use of 'Individual spaces', such as allocated desks, and 'Personalisation of space' as techniques that will improve student's sense of ownership over the studio space.

Inspired by Other Students' Work:

Having some awareness of your surroundings was discussed as being potentially be good for inspiration. Almost all of the participants that discussed the importance of garnering inspiration through 'Proximity to other people' also discussed the importance of being able to 'Easily observe other people's work' (A_1-A_5, A_7-A_{10}, A_{13}-A_{15}).

3.3.3 Anomalous Categories

There are two anomalous categories that have been omitted from the studio framework, 'Defining the Studio' and 'Digital Technology'. These were significantly discussed during the interviews, but do not make up any discussion of what a studio actually is.

Defining the Studio

This category consists of discussions around how to define the studio, grouping anything related to the studio's purpose (a high level definition) and whether the interview participants thought the studio was worthwhile; in essence, it is a representation of their perspectives of the studio, not a set of features. Some of the interview participants highlighted that studios are too complex to be captured by simple definitions or description, and that there was no perfect space.

A studio has multiple purposes and is *"a mix between the working space and social space"* (A_2), with most of the participants having discussed the importance of having other people in the studio (A_1-A_5, A_7-A_{10}, A_{12}-A_{15}). Generally, all participants preferred to use studios, with only one participant (A_{12}) expressing an indifference to their use. Participant A_{12} observed a trend where people attempt to replicate the studio setting that they had at university into a professional setting, or in their own teaching style. This participant summarises this by saying: *"people teach the way they were taught, in the same way that people speak the language that they grew up speaking. We are a product of our environment"* (A_{12}).

Other key aspects which were discussed were that the studio prepares you for work environments, and also that it is a creative space. One last important factor discussed was that the studio is not necessary – people can work without one, yet a studio was still a preferred environment for almost all of the interview participants.

Digital Technology

This is all about how digital technology can be beneficial or detrimental to a studio – primarily the computer and similar devices. This is an anomalous category, because unlike the others, it does not affect the definition of the studio; a studio can exist with or without any form of digital technology. Activities can be purely pen and paper-based for example. Indeed, studios were around centuries before the first computer. However, this topic cropped up in many of the interviews, and the interview participants felt strongly about this topic. Some participants (A_6, A_8-A_9, A_{15}) particularly noted how digital technology allows for easier access and the ability to perform work outside of the studio, but at the same time acknowledged that social interactions within a studio are not repeatable with digital media.

Some of the participants (A_6, A_7, A_{14}) thought that computers were indispensable, with several others also discussing significant benefits that computers provide, whilst conceding that their use is a trade-off. Participant A_{14} remarked that they were a *"massive help just in every single way"*, but the majority remarked on how detrimental they are to the studio experience, with participant A_4 stating that *"using the computer has taken us away from working in teams"*. Although participant A_6 discusses the benefits of digital technology in a design context, he also shares this negative view on the ubiquity of technology: *"people, sometimes, their first port of call is their computer [...] there is a danger that people always use a computer for everything when, sometimes, there may be a quicker or easier way of doing it"*. Computers can also create a culture of isolation, with all of your work and activities hidden behind a screen. Participant A_{13} supports this by saying: *"when you are looking at things on the screen you tend to be looking at one thing at a time"* – I refer to this as 'Monitor Vision' or 'Tunnel Vision'. Similar thoughts have been expressed by others outside of this investigation, in the field of Architecture: *"We fear that computers may devalue the art and craft of architecture, decrease collaboration, isolate students, and emphasize product over process"* (Koch et al., 2002, p.11). However, one advantage of using digital tech-

nology discussed by participant A_{13} revolves around her experience with an online element to the course she teaches on, stating that her students post work online and provide peer-critique to each other; they do not assess *"the quality of what they give in the feedback but the fact that they are doing it and engaging with other people's work"* (A_{13}), the asynchronous nature of peer-critique in this particular way also *"probably works better online because there is a bit of time for reflection"* (A_{13}).

Throughout the interviews it was made very clear that, despite being necessary for an activity or providing other benefits, the participants thought that computers are detrimental to studio culture (A_1, A_4-A_5, A_7, A_{13}-A_{15}). The points raised by the participants were often made independent of discipline specific activities, and as such are just as relevant to other disciplines (including software engineering), as they were not talking about a specific design activity being inhibited, but general interactions, communication and awareness. As software engineers, we obviously will not be moving away from computers, but this begs the question: what can we do to avoid the negative effects of computer work in collaborative environments? A further interesting question is: what innovation will replace the computer monitor, will our attention still be monopolised in a single direction, on a small flat surface, i.e. a screen?

3.3.4 Most Important Aspect of a Studio

All of the interview participants were asked the same final question: *"What is the most important aspect of a studio?"*; they all provided an answer, except for one. According to them, the most important aspect of studios are *"the people in the space"*. This is also reflected in the 'Culture' category of the studio framework above. Below is a list that summarises the 15 participants' responses:

- **7x** participants said that it was the "people" that were important.
  - *[Participants A_2, A_3, A_4, A_5, A_{10}, A_{12} & A_{13}]*
- **1x** Unable to answer the question (due to complex nature of a studio).

- *[Participant A_1]*

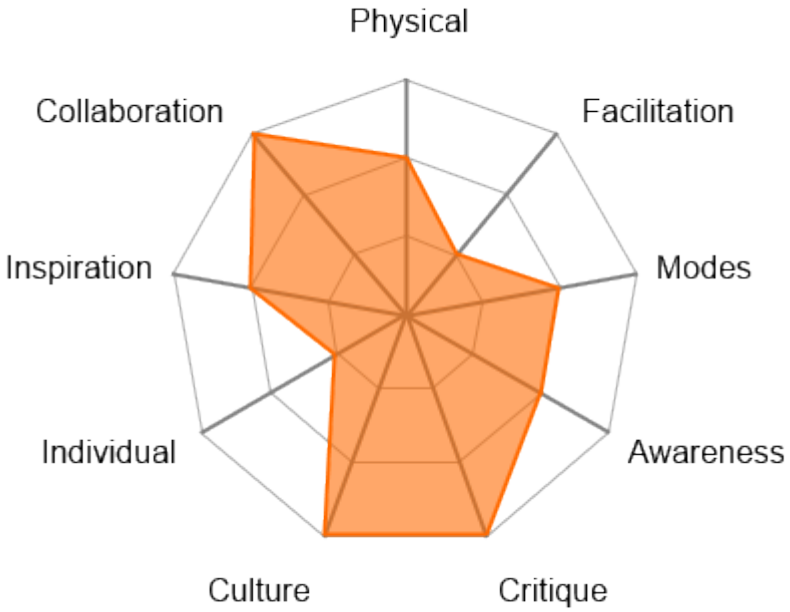
- **1x** Open and flexible space, that supports creative activities.
  - *[Participant A_6]*
- **1x** Movement around the space (i.e. not static rows of chairs).
  - *[Participant A_7]*
- **1x** Peer-learning; as a specific learning activity.
  - *[Participant A_8]*
- **1x** Facilitation of the space.
  - *[Participant A_9]*
- **1x** Freedom to work as you wish.
  - *[Participant A_{11}]*
- **1x** Serendipity of ideas from close proximity to other people.
  - *[Participant A_{14}]*
- **1x** The ability of the individual.
  - *[Participant A_{15}]*

3.4 Discussion

Taking the categories of conversation derived in the interview analyses, we can now say we have a better understanding of people's views of what studio education is. Based on the discussions with the interview participants, it is the people and the culture that make a studio, not the environment, although the participants did speak highly of the physical environment category, and how much potential it has to influence studio education.

Without an agreed definition or understanding of what a "studio" is, specialised studios, such as software studios, cannot be properly evaluated for their impact on student learning. Due to the complex nature of a studio it is inherently difficult to succinctly define what a studio is, as explored in section 2.2. The people and the cultures that create a studio cannot be encompassed within one definition. As such, the analysis in this chapter aims to provide a broader understanding of studios in place of a single definition.

Figure 3.1: An example visualisation of the studio framework.



3.4.1 Towards an Evaluable Model

Based on the enhanced understanding of studios presented in this chapter, an evaluable model can be provided to determine the authenticity and extent of a studio. The model is a representation of the studio framework (Table 3.2) presented in an accessible form, for example a radar chart. Determining whether a software studio is an authentic studio will clarify if the students that use it have the potential to benefit from the creativity that a studio supports.

The framework in its current form requires subjective interpretations of a studio for each of the categories. To easily represent and share the model after an evaluation, it is suggested that a radar chart is used; an example, with arbitrary values, is given in Figure 3.1. The nine axes are named as the framework's categories, and the values are provided on a short scale of 0-3 which describe how well each category/axis is implemented in a studio. The value given to each axis for a given studio is given by how many parameters are covered or implemented in a studio:

0. Not covered at all.

1. Slightly covered (very few parameters).

2. Somewhat covered.

3. Significantly covered (most parameters).

3.4.2 Applicability of the Framework to a Software Studio

The studio framework is generic enough that it is conceivable that all of the framework's parameters are capable of transferring to software studios (no parameters appear to exclude software engineering). Indeed, certain development approaches will work well in some of the framework's parameters, such as extreme programming (XP) as it *"encourages social interactions"* (Dubinsky and Hazzan, 2005, p.646).

There are three categories which could potentially be cause for concern: 'Physical Environment', 'Awareness' and 'Digital Technology'. Awareness could be severely limited due to the heavy reliance on computers. This is a possible example of limited awareness, as participant A_{13} stated earlier: *"when you are looking at things on the screen you tend to be looking at one thing at a time"*. This could have adverse effects, with most or all work hidden behind monitors. Another likely issue would be the physical environment as most computer-based rooms can often have the computers locked to the furniture and plugged into the floor, making tables difficult or impossible to move. Another possible concern is the digital technology aspect, as software engineering is inherently computer-based, how much will they affect the other aspects?

There are several avenues of future research regarding this framework; first of which would be to assess whether the framework is appropriate for evaluating studios. This would then naturally lead on to a series of evaluations of studios using this framework. Further research beyond that would then be able to determine if studios in software engineering are indeed effective and that the effectiveness is not simply a by-product unrelated to the studio. Finally, intriguing questions can be

asked about the differences between studios at separate institutions and whether we can easily identify and transfer elements of one studio to another.

3.5 Conclusions

The research presented in this chapter provides the tools necessary to determine the authenticity and extent of a studio by contributing three elements. Firstly, an analysis of the interviews is shared, which includes a list of the dominant categories discussed and how they interrelate, to positive or detrimental effect. This provides a broad understanding of studios to counter the lack of an agreed upon definition. Next, a framework to evaluate studios against is provided, which is derived from the analysis of the interviews. Lastly, a discussion is given about how the aspects of the studio framework may transfer across to software engineering and what the key challenges are.

Now that studios can have their authenticity evaluated, software studios can be properly evaluated to determine how effectively the studio aspects have been implemented. If a software studio is evaluated, it could then be determined whether the students have indeed experienced the benefits that a studio is said to provide and support – or whether there are any inherent incompatibilities between traditional studios and software engineering education. One potential avenue of future research, considering these possible incompatibilities, would be to identify missing or diminished attributes using the framework, develop and deploy a solution in the software studio and see if improving that attribute helps.

To conclude this chapter, one participant summarises a prominent opinion from the interviews. It resonates with many of the interview participants' views, and reinforces the variability and complexity of such spaces:

> *"[studios are] the product of the people that work in it, it's not a particular type of space"* (A_{12}).

4 State of Software Studios in Universities

The studio framework, presented in the previous chapter, is intended to provide an understanding of what studio education involves. The framework was constructed using the perspectives of people within the disciplines that studio education originated; a gap that required exploring due to the lack of a settled definition, as discovered in the literature review. However, a number of studio courses have already been designed and implemented in software-based courses, prior to this framework's inception, and they vary widely in their implementation (these are also presented in the literature review). For this reason, this chapter shifts focus to software disciplines – those that include elements of software engineering and development, e.g. software engineering, computer science and information technology. From this point forward they will simply be referred to as software engineering courses and software engineering educators. This chapter focuses on the research question:

RQ2) What is the current state of studio education in software engineering?

It is important to explore the state of studio education in software engineering courses to determine how their perspective of this pedagogy differs or persists when compared to the arts and design disciplines. Would these courses be recognised as studios in the eyes of artists and designers?

To achieve this, simply comparing these 'software studios' to the studio framework may not work, as the fundamental disciplines taught in these studios are fundamentally different to the design and art disciplines. There is also no history or implicit transfer of studio knowledge between generations of software engineers, unlike design studios where teaching staff remember studios they were taught in, because it is a new approach for software engineering courses. Instead, a similar investigation to the one that produced the studio framework in the previous chapter was performed, but this time interviewing educators from software studios.

The purpose of this set of activities was to determine what software engineering educators perceive as a studio; for example, would there be any difference of opinion to the previous design/art/architecture educators that is reflected in the design of their studios? The main focus here is to get the educators to reflect on software studio courses they are familiar with. Additionally, the interviews would be used to explore whether there are any elements specifically in a software studio (and not in a traditional studio) that support any part of a software development process? Essentially, how do software engineers interpret the studio?

This chapter presents the analysis of interviews with software engineering educators, following the a similar methodology as the previous chapter (described in detail in the following section). The analysis in this chapter is intended to assess the conformance of the software engineering studio interpretations to the studio framework presented in the previous chapter. The analysis presented here culminated in an extended framework that built upon the previous chapter's framework, referred to as the 'software studio framework', and represents the participants' views on software studio education. Conversational elements were categorised using the previous chapter's existing categories and parameters as a starting point, though through using constant comparison, new categories and parameters were permitted. The framework presented in this chapter consists of largely the same set of categories as the previous chapter's interviews, with the exception of the 'Inspiration' category. However, there is variable overlap with the parameters

found within these categories from both studies in this chapter and the previous chapter; this suggests that a broad understanding and implementation of studio education exists across software engineering and the design disciplines, although individual participants rarely discussed all of the categories. Interestingly, it is the parameters of the framework that differs from the previous chapter.

4.1 Methodology

With the exception of the chosen interview participants, and therefore their background or experience in software engineering courses, the methodology employed is similar to that described in the previous chapter, section 3.1. The methodology employed in this chapter for this series of interviews is theory-driven, as it uses the studio framework generated in the previous chapter as a starting point, and is briefly described below with emphasis on the differences (i.e. the participants); for a complete description of the methodological process, including approach to coding and categorisation of the transcripts, refer to the previous chapter's methodology section (3.1).

Studios in the design, art and architecture disciplines are immersed in their culture, and it is reasonable to state that elements of it are tacit within those disciplines. Due to how new the studio concept is to software education, and how varied each implementation is (see chapter 2, 'Literature Review'), software engineering educators' understanding of studios may widely vary – hence these interviews.

The semi-structured interviews followed the same interview design as the previous chapter. The analysis, however, starts with a set of already defined concepts (parameters) and categories. The codes and concepts drawn from the interview transcriptions were initially categorised using the previous framework's concepts and categories. The intention was to create new concepts and categories if the codes did not fit into the previous framework. However, no new categories arose from the analysis, but new parameters did.

In summary, educators from software engineering courses were interviewed, exploring their experiences and opinions of studios within software engineering courses. The interview transcripts were then analysed, generating codes, concepts (parameters), and categories, based on the previous framework (though not limited to it), which would formulate a new studio framework: the Software Studio Framework.

4.1.1 Interview Participants

Following in the footsteps of the previous chapter, the participants were expected to be mostly software engineering educators with studio experience. Potential participants were short-listed by reviewing publications on software studios, those explored in chapter 2, 'Literature Review'. A representative selection of those publications' authors were contacted and asked to be interviewed. The invited participants were selected based upon their variety of experience, geography, maturity of studio, and degree that is offering the studio; see Table 4.2 for more information. The final list of participants came from a range of institutions spanning three countries: United States of America, Switzerland, and Australia.

The final list of participants is shown in Table 4.1. The table lists the experiences of the participants; if they were educated in a studio, whether they have taught others in a studio, if they have used a studio in their working practice, and whether they have been involved in research on studio education. The subjects listed for each participant are the primary ones that they teach or research. None of the participants in this set of interviews participated in the previous interviews.

All participants were from software engineering courses, with the exception of B_3 who specialises in research on education. This participant has performed an *"empirical social science study of studios"* (B_3), and has subsequently acted as a consultant to some academic institutions, helping them implement studio education in software engineering courses. Participant B_6 was educated in a traditional studio environment for a 'Bachelor of Design Studies', followed by a second under-

Table 4.1: Interview participants (Software studios)

Participant[*]	**Subject(s)**	**Studio Experience**[†]		
		Educated	**Taught**	**Research**
B_1	SE		✓	✓
B_2	SE		✓	✓
B_3	Edu. Innov.[‡]			✓
B_4	CS		✓	
B_5	CS		✓	
B_6 [§]	IT	~	✓	
B_7	SE, IT		✓	✓
B_8	SE		✓	✓
B_9	SE		✓	
B_{10}	IT		✓	✓
B_{11}	SE	✓	✓	✓
B_{12}	SE	✓	✓	✓
B_{13}	CS	✓	✓	
B_{14}	CS		✓	✓
B_{15}	CS, SE, HCI		✓	✓

[*] Participants in this second set of interviews are labelled as 'B' with a subscript number to differentiate between the 15 participants (e.g. 'B_{13}') – these are also used as a unique identifier throughout this thesis.

[†] Studio experiences of the participants: Whether they were educated within a studio, taught others in a studio, or researched studios.

[‡] Participant B_3 is an educational innovations researcher; this participant has implemented software studios.

[§] Participant B_6 was educated in a traditional studio, but not a software studio.

Table 4.2: Representative set of interview participants

Participant	**Location**	**Maturity of Studio***
B_1	USA	1990+
B_2	USA	2005+
B_3	USA	N/A §
B_4	Switzerland	2000+
B_5	Switzerland	2000+
B_6	Australia	2010+
B_7	Australia	2010+
B_8	Australia	2010+
B_9	USA	1990+
B_{10}	Australia	2000+
B_{11}	USA	1990+
B_{12}	USA	1990+
B_{13}	USA	1995+
B_{14}	USA	2005+
B_{15}	USA	2010+

* To preserve anonymity, the starting year of the studio that a participant is involved with is approximated. The year given is based on the year of the initial publication (unless a date is given within a corresponding publication), but rounded down to the nearest five year increment (e.g. a studio starting in 1997 is presented as 1995+, and 2004 as 2000+ etc.).

§ Participant B_3 is an educational innovations researcher. He has researched studio education, but does not teach in any. See Table 4.1 for more information.

graduate course in 'Bachelor of Information Environment', which was taught in the manner of a traditional IT course. This led to a career which involved teaching a mix of design and computing knowledge, in a studio environment, on an IT degree program. Participants B_{11} and B_{12} were educated in a software engineering studio course. Participant B_{13} had some experience of being taught in an architecture studio, before switching major to computer science.

Throughout this chapter participants B_4 and B_5 refer to the studio as an 'atelier'. This name, described in the literature review section on the École des Beaux-Arts (2.1.1), is an alternative name for a studio (originating from the French language); for the purposes of this thesis, the names are used interchangeably, and unless within a direct quote (or referring to those participants who use that name), these practicums will be referred to as 'studios'.

No further participants were needed to be recruited as the research reached saturation with the current list of participants; by interview participant B_{11}, no new concepts fell into a category that did not already contain concepts from earlier analyses.

4.1.2 Interview Design & Analysis

As already mentioned, the interview design and the analysis followed a similar process as the previous chapter, sections 3.1.2 and 3.1.3 respectively. The interviews used the same initial set of questions as the previous interviews, but software engineering specific questions and discussion surfaced through the semi-structured nature of the interviews. These interviews were audio recorded, transcribed, and then underwent a process of coding and categorisation (Glaser and Strauss, 1967). This particular analysis was theory-driven by using the studio framework (section 3.2) as a starting point, unlike the previous chapter which was data-driven. Codes, concepts, and categories were reused where appropriate, but with new codes, concepts, or categories created if they did not fit into the existing framework. This analysis followed the same approach to constant comparison of new codes etc.

The second phase of the analysis was to compare and contrast both of the generated studio frameworks. Differences and commonalities of the frameworks are described, and finally the two frameworks are merged and presented as the final extended 'Software Studio Framework'.

4.2 The Software-Studio Framework

The results of this chapter's analysis are placed inline with the previous chapter's studio framework (chapter 3) for easier comparison. The resulting table (Table 4.3) lists each of the categories and parameters from both analyses. This representation of the studio framework shows that both sets of interviews generated largely the same set of categories, with the exception of the 'Inspiration' category, which was only discussed by the original framework's interview participants. The main differences are the parameters that reside within each of the categories, with some categories having strong overlaps across the studies, whilst others have new parameters added but very few overlapping parameters. This may suggest that different parameters are a symptom of different domains or priorities. A matrix of the framework's categories and parameters against the interview participants from this chapter is also presented in Table 4.4. Further comparative analysis between the two frameworks is also conducted later, in section 4.3.3.

In this studio framework, shown in Table 4.3, you will notice the inclusion of two column headings ('Study 1' and 'Study 2'). These columns provide information on which analysis the parameters were found in; they can either be found in the first study in chapter 3 ('Study 1'), the second study in this chapter ('Study 2'), or both. A check mark in the column for 'Study 1' denotes the associated parameter's existence in the original studio framework (the previous chapter), and a check mark in column 'Study 2' denotes its existence in the software studio framework variant. Using the first parameter as an example ("Open environment"), the table tells us that this parameter appears in the first framework, but does not appear in the second.

Table 4.3: The software studio framework

Categories	Occurrences		Parameters
	Study 1*	Study 2†	
Physical environment	14	-	Open environment
	14	10	Flexible space (Reconfigurable furniture)
	5	-	Students control aesthetic factors (lighting, heating)
	15	8	Shared spaces
	8	-	Individual spaces
	7	-	Social spaces
	7	6	Private spaces
	-	8	Multiple spaces
Facilitation of studio	8	-	Studio belongs to the students
	6	-	Staff do not dictate use of space
	7	-	24 hour access
	6	-	Food and drink allowed
	5	8	Availability of multiple expert staff
	5	-	Small group size (about 10)
	-	6	Expected time spent in the studio
	-	5	Define own projects
	-	5	Learning to learn
	-	10	Flexible facilitation
Modes of education	13	9	Flexible learning/Switch approach based on activity
	11	12	Mentoring/coaching
	12	10	Peer-learning
	5	-	Impromptu teaching
	-	13	Project-based learning
	-	13	Real-world projects
Awareness	13	-	Visual work
	13	10	Externalising thinking/Displaying work
	6	-	Visual history of progress
	12	10	Peer's experiences/Easily observe other people's work
	8	8	Social interactions
Critique	12	8	Direct feedback
	11	11	Developing ideas
	11	14	Multiple formats (formal/informal, individual/group)
	10	-	Peer-coaching
	-	13	Peer critique
	-	10	Continuous critique
Culture	10	7	Sharing
	14	8	Social
	6	-	Treated like second home
	10	8	Good work ethic
	6	7	Peer-support
	8	-	Serendipity
	-	7	Permission to fail
Individual's characteristics	9	-	Personalisation of space
	10	5	Private and quiet spaces
Inspiration	13	-	Proximity to other people
	5	-	Relevant available media
	5	-	Library of liked/fun things
	5	-	Playful space
Collaboration	6	7	Impromptu collaborative spaces
	7	10	Supporting equipment
	-	12	Group size considerations
	-	11	Soft skills
	-	7	Equal effort

* Parameters that arose from the analysis of the first interview-based study in chapter 3: interviews with designers, architects, and artists.

† Parameters that arose from this chapters' interview analysis.

Table 4.4: The software studio framework's participant matrix

Studio Framework Parameters		**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**	**B8**	**B9**	**B10**	**B11**	**B12**	**B13**	**B14**	**B15**
Physical environment	Flexible space (Reconfigurable furniture)		✓		✓	✓	✓	✓			✓	✓		✓	✓	✓
	Shared spaces	✓	✓		✓	✓			✓		✓		✓			✓
	Private spaces	✓			✓		✓		✓	✓	✓					
	Multiple spaces	✓			✓	✓		✓	✓	✓	✓		✓			
Facilitation of studio	High availability of staff	✓	✓		✓	✓	✓	✓		✓			✓			
	Expected time spent in the studio	✓		✓		✓	✓			✓				✓		
	Define own projects			✓	✓	✓	✓	✓								
	Learning to learn	✓		✓		✓	✓			✓						
	Flexible facilitation		✓	✓		✓	✓	✓	✓	✓	✓			✓	✓	
Modes of education	Flexible learning/Switch approach based on activity	✓	✓	✓	✓	✓	✓	✓		✓	✓					
	Mentoring/coaching	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓
	Peer-learning	✓	✓		✓	✓			✓	✓	✓	✓		✓	✓	
	Project-based learning	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	Real-world projects	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Awareness	Externalising thinking/Displaying work	✓	✓	✓	✓	✓	✓		✓			✓		✓		✓
	Peer's experiences/Easily observe other people's work	✓	✓		✓	✓	✓		✓			✓		✓	✓	✓
	Social interactions					✓			✓	✓	✓	✓	✓	✓	✓	
Critique	Direct feedback	✓	✓	✓	✓		✓			✓			✓		✓	
	Develop ideas	✓	✓	✓	✓	✓		✓		✓	✓	✓			✓	✓
	Multiple formats	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
	Peer-critique	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓
	Continuous critique	✓	✓	✓		✓	✓		✓	✓		✓	✓		✓	
Culture	Sharing	✓			✓	✓		✓			✓		✓	✓		
	Social		✓						✓	✓	✓	✓	✓	✓	✓	
	Good work ethic		✓	✓		✓	✓			✓				✓	✓	✓
	Peer-support		✓	✓			✓			✓	✓	✓			✓	
	Permission to fail	✓		✓	✓					✓		✓	✓	✓		
Individual's characteristics	Private and quiet spaces	✓			✓		✓			✓	✓					
Collaboration	Impromptu collaborative spaces		✓		✓				✓	✓	✓	✓		✓		
	Supporting equipment				✓	✓	✓	✓	✓		✓	✓	✓	✓		✓
	Group size considerations	✓	✓		✓	✓		✓	✓	✓	✓	✓		✓	✓	✓
	Soft skills	✓	✓	✓		✓		✓	✓		✓	✓	✓		✓	✓
	Equal effort		✓		✓	✓			✓			✓	✓		✓	

There was also a general trend separating the first two frameworks, with the first set of interviewees providing a balanced emphasis on students' group and individual aspects to studio education. Whereas the second set of interviewees, from the software engineering courses, had far greater emphasis on studios for use with group and collaborative work. Even though some participants did discuss 'Individual characteristics' in the second set of interviews, it was significantly outweighed by the conversations about team efforts from all of the participants.

The participants continued to share the difficulty of defining what a studio is. For example, one participant remarked:

> *"I actually haven't tried to define it. [. . .] there's a set of practices, [by] that I mean, a complex set of practices that interweave in a consistent way so it's kind of like a complex hole. Yes, it would be hard for me to define it, I guess maybe I could, each definition would really just be a characteristic"* (B_3).

Similarly, another participant shared the opinion that the *"studio itself is more of an idea rather than the physical makeup"* (B_1). This idea is supported by all of the participants stating that the physical space is not the most important aspect to a studio, though still important (B_1-B_{15}).

Table 4.3 shows the second studio framework, a software studio variant specific to this chapter's series of interviews. Descriptions for each of the categories that make up this variant of the studio framework are presented below in the following subsections (4.2.x). Whilst there is significant commonality across the two frameworks categories, the 'parameters' (the elements that make up the category) have more variability compared to the previous framework. As an example, the category 'Inspiration' was not discussed during this chapter's interviews. The descriptions of these categories are also accompanied with some select quotations from the participants.

4.2.1 Physical Environment

The physical environment (the location and physical contents of a studio) plays an important role in a studio, according to most of the participants (B_1, B_4-B_{15}), with one participant stating: *"I'm convinced that the room, the space, has a major impact"* (B_4). The studios were often a space shared by all of the students (B_{15}). However, not all participants felt it is a crucial element (B_2, B_3). For example, most of the participants (B_1, B_3-B_7, B_9-B_{12}, B_{14}-B_{15}) discussed how they have dedicated studio space, even purposefully designed as a studio space, whilst some others have simply repurposed already available classrooms (B_2, B_8, B_{13}). One participant, B_2, used *"just your regular classroom with just normal chairs"*. This indicates that a studio does not need to be a special space, but it is what you do within that space that makes it a studio. This is also reflected in participant B_4's final remarks in his interview on what he thought was the most important aspect of a studio, stating it is *"the teacher, followed by the physical space"*.

Although this opinion was in the minority, participant B_{15} claimed that a physical space is potentially not necessary; at the time of the interview he was exploring *"completely online studios"*. As it was ongoing research he added that the *"verdict is still out"* (B_{15}). When pressed about what he thinks a studio is, he replied saying that he is *"not necessarily thinking of a specific room, although a lot of the work I've done so far, most of it in fact, has been face to face"* (B_{15}). He thinks that the challenge for online studios is how to *"integrate the expert perspective into this online environment"* (B_{15}). This is also reflected in another participant's comment, stating that *"the space is a lot less important than it used to be"* (B_{13}), due to most work being computer-based and remotely accessible.

The following subsections describe the interview participant's perspective on all of the parameters that were discussed relating to this category: Private spaces, Flexible space, Multiple spaces.

Flexible Space

During the interviews, there was a lot of emphasis on the flexibility of space within a studio (B_2, B_4-B_7, B_{10}-B_{11}, B_{13}-B_{15}). Participant B_2 discussed how the flexibility of a space has a knock on effect on collaboration: *"a fixed classroom, where [the students] can't actually congregate together around the group or around the table or around some kind of physical space, that would kill it"*, adding that that the students *"have to be able to face each other"* (B_2). Participant B_{15} discussed his experience of a flexible studio space which aids collaboration: *"The tables are very configurable, we have the projector in the front of the room, there are whiteboards all around it, and so it provides opportunities for different learning modalities, including studio works, small group work"*. Another example scenario of flexible spaces in a studio was described by participant B_4, which uses partition walls:

> *"when we started, we had a big room that we divided with partitions that we could separate different areas into little rooms, so little cubicles, but with high partitions. So, depending on how many students were working on a project, they could form their own little cubicle for a period of time while they were working on the project. When the professor was going to talk to them about some topic they would move the partition so that dynamically rearranged the physical partitions. [...] it really helped. The motivation was high, people were very interested and very active, but after two years we moved into a new building that was built for us and here, unfortunately, there is a large open space, but we don't have the facility to form individual partitions, so it is just an open space and when several people are working together, it's loud and people don't have much control over their environment. I can definitely see that it's definitely not as conducive to this style of teaching"* (B_4).

When asked if not having the partitions in the room can yield any benefits, B_4 replied saying that, in his experience, not having partitions in the room can have *"potentially positive benefits, but in practice it's only been negative"*. Partitions

add to the flexibility of the space, but detract from the openness of the space. Partitions can also add an element of privacy from other groups within the room, finally stating that *"you form the office based on what you need to do"* (B_4). This also reflects on a conversation with B_5, who stated that on the *"first day [the students] get a laptop and they keep it throughout their studies. [...] There is no need for having a special lab environment set up"*. He discussed how laptops provide the students with an important element of flexibility for their work environments, but *"you need to have infrastructure, like connectivity"* (B_5) to support this further – limited connectivity reduces flexibility opportunities.

Shared Spaces

Another common discussion was the importance of shared space (B_1-B_2, B_4-B_5, B_8, B_{10}, B_{12}, B_{15}). This aspect has several overlaps with other categories and parameters, such as awareness and collaboration, which benefit from occurring when students are co-located in a shared space, however, a shared space is not essential for those aspects. Despite this, these participants discussed the importance of choosing and using shared physical spaces.

On the other hand, according to some participants (B_6, B_8-B_9, B_{15}), some benefits afforded by a physical shared space can also be obtained in a virtual online environment. Whilst these participants did not explicitly discuss that an entirely virtual space could completely replace a physical one, they did discuss that they can compliment each other. For example, one participant discussed how they try and achieve the same benefit of a physical shared space in a virtual environment for times when students are not co-located in a physical space: *"students were posting up work-in-progress, for their groups to keep track of, for the other groups to see what they're doing"* (B_6).

Private space

One of the types of space that the participants sometimes spoke about were private spaces (B_1, B_4, B_6, B_8-B_{10}), somewhere for the students to isolate themselves (even just partially) from each other. This was mentioned as a benefit to both individuals and groups. Participant B_4 discussed experiences of two studios (an earlier studio, and then the newer studio which they were moved into), in particular the use of partition walls. These walls would sit atop desks, and could easily be moved to accommodate user's needs. In his first studio, participant B_4 stated that these partition walls were sufficient at enabling private space for collaborating groups. However, in the subsequent studio there were no partition walls. Whilst he admits that this is not a significant failing in a studio, he did go on to clarify that *"you shouldn't take what I say as 'no partition is bad', but what I'm saying is, if you want students to work on a project together, you need to provide them a space where they feel like they have a private space"* (B_4). This also resonates with participant B_5 who stated that it is important for a studio to provide *"lots of different small meeting corners for when they're working in pairs for projects"*.

Lastly, a couple of participants discussed problems with disruptive noise in open spaces (B_4-B_5). Although this is not unique to studio education, noise levels are an inherent potential problem with any activity or teaching approach that uses open spaces with multiple people occupying the space. In participant B_4's experience, the problem with noise in open spaces starts to become an issue *"when there is more than ten people in this open space and they start talking to each other, the noise becomes unbearable"*.

Multiple Spaces

Several participants (B_1, B_4-B_5, B_7-B_{10}, B_{12}) also discussed the necessity for having multiple types of spaces in a studio (e.g. private spaces and collaborative spaces), with participant B_5 succinctly saying that *"ideally the space should be multiple spaces, banded together"*. One of the environments described by this participant

was *"a room that is full of whiteboards,"* which he admits specifically *"helps to do a certain type of work"* and is not necessarily a multi-purpose room (B_5). In this instance B_5 discussed a couple of separate rooms, but a few other participants described a singular space with multiple areas for similar purposes (B_4, B_7). Both were considered part of the studio courses. B_5 discussed his experiences of the primary studio space not offering dedicated areas within it: *"They don't really have [...] their own corner, or their own meeting place that is somehow isolated from the others"*, adding that those would be ideal.

4.2.2 Facilitation of Studio

This category refers to how a studio is run – how it is organised, the teaching staff, the rules that govern the course and space(s), and conveying adequate expectations to the students.

High Availability of Staff

The high availability of numerous staff was a common discussion amongst the interview participants (B_1-B_2, B_4-B_7, B_9, B_{12}). According to B_2, an ideal studio would have multiple *"expert tutors"* amongst the students, stating that *"it would be really nice if [...] there were multiple of us walking around, who could do that"* (B_2). Other participants (B_2, B_4, B_{10}) also emphasised the importance of the teaching staff, saying that how the teaching staff lead the students is the most important aspect of the studio – participant B_{10} pushes this notion by saying *"you can have the worst designed studio, but ultimately it is that mentor that motivates and encourages"*. He went further and stated the importance of the capability of the teaching staff being well versed in both theoretical and practical skills: *"If you have the right professor it works out very well because it's very difficult to actually organise and teach this kind of course because it needs not just the conceptual theoretical knowledge but to be able to also be well versed in very practical aspects so help the students to debug the project that they are working on"* (B_{10}). Later,

B_4 wanted to clarify that the studio is *"not very difficult [to teach] but it's more difficult to teach such a class than a traditional lecture"*. Participant B_{15} also shared his observation that *"if you set students loose on the critiques of each other's work without expert guidance, they quickly become frustrated"*.

Participant B_5 discussed the use of teaching assistants in conjunction with other expert teaching staff. It was said that the students frequently *"can go and ask the TA"* because the teaching assistants are in his studio with the students for two hours a day, three days a week. In this studio's example, the students can also approach the main teaching staff, with B_5 saying that *"they can come and ask me"*, as staff are also in every day (though at different or overlapping times to the teaching assistants).

Expected Time Spent in the Studio

The time that the students spend in the studio is important, and is significantly impacted by the expectations that the teaching staff convey to the students. Participant B_5 discussed how the frequency that the students attend the studio is in part down to how the course is structured around it; for example, timetabled sessions that occur outside of a studio (e.g. lecture, seminar etc.) obviously reduce time spent within the studio. Whilst no participant explicitly expressed how long students should spend in the studio, several participants (B_1, B_3, B_5-B_6, B_9, B_{13}) described their specific implementations, and shared their expectation that students spend a significant amount of time in the studio each week. Participant B_5 discussed how at one time all students would attend their studio *"every afternoon"*, but there has since been some *"restructuring of the curriculum, so this has been reduced to three days per week"*. He discussed the reasons behind this as providing more time to other courses in their degrees.

Define Own Projects

Another important element that some participants discussed (B_3-B_7) was the choice to let the students define and lead their own projects, rather than have them specified for them. Whilst this is not necessarily indicative of a studio, it can improve student's motivation and enrich the studio culture. Participant B_5 acknowledges the benefit to student motivation, saying that *"sometimes we give it to them but most times they actually come up with the idea themselves. [...] We, of course, help them to scope it [...] but they are much more motivated. [...] they really work hard to make it work."*

Learning to Learn

Some participants (B_1, B_3, B_5-B_6, B_9) discussed the importance of not just learning the facts of a given topic, but learning how to learn. One of the reasons given by participant B_5 is because *"by the time you have started the bachelor there will be some technologies, and by the time you finish your studies, these things will be obsolete. So, the goal is to teach students how to learn new technologies and pick them up on their own"*. Another reason to emphasise learning how to learn is so that the students can become proficient practitioners: *"People might be overwhelmed by all the information that they find [...]. Sometimes they need to learn how to search for solutions. So, they have the opportunity to find solutions themselves"* (B_5). Other participants (B_1, B_6, B_{12}) have also described how they encourage their students to critique their choice of software tools and employed processes over other alternatives. This can equip the students with the necessary techniques and experience to cope with the constantly evolving technology field. This emphasis on learning to learn benefits from having significant interaction with teaching staff.

Flexible Facilitation

Lastly in this category, several participants discussed the necessity for flexible facilitation (B_2-B_3, B_5-B_{10}, B_{13}-B_{14}). One participant in particular described the importance of flexible facilitation and time management, stating that *"depending on how fast the students are going, or depending on whether there is a need for extra discussion or longer time for feedback"* (B_5). This should be more easily obtainable with members of staff who are closer to the students, for example in a mentoring or coaching role.

4.2.3 Modes of Education

This is one of the more significantly talked about categories, with many of the participants discussing a variety of flexible learning techniques that take place in their studios.

Flexible Learning

The participants discussed how many of approaches used are not forced on the students, how the students are free to switch their approach to a problem as they see fit, and how these techniques can surface because of the flexibility that studio education affords them (B_1-B_7, B_9-B_{10}). Participant B_5 provides an example, saying that *"it depends on what we are doing. There are some days where it's more of a lecture setting, so everybody is facing me. Other days where it's hands on"*. He also stated that the students often switch between working independently and in groups, which they decide upon dependant on the given activity: *"there are other days when they work independently or as a team"* (B_5).

Mentoring/Coaching

Mentoring, or coaching, was frequently discussed during the interviews (B_1-B_5, B_7, B_9-B_{13}, B_{15}). The participants consider it an important aspect to studio education because the students receive rich and frequent interactions from teaching

staff. However, participant B_2 states that this mode of education will suffer if the number of students grows too large. Most of the participants described how coaching can take place in their studio, with B_3 stating that one time when coaching occurs is when *"the students are working at their own tables and the instructors are wandering around talking to them one on one"*. Participant B_4 shared a similar story, stating that *"the professor or teaching assistants go around and observe what they're doing and then they comment and critique"*.

Mentoring requires significant effort to perform appropriately, as participant B_1 states that *"when you mentor a full-size project, it's equivalent to teaching a full class with the amount of time you put in"* – this is why many participants describe the need for multiple teaching staff in a studio course (see section 4.2.2, 'High Availability of Staff'). In B_1's specific situation, he describes training these mentors: *"We actually have a programme for new mentors which simply put here is the master-apprentice type model, you have somebody follow you for the whole project"*. However, in participant B_1's institution, there is a problem with *"recognising mentoring as a teaching activity"*, as he discussed how mentoring a student project may not fall under a teaching staff's contracted teaching time (e.g. hours spent lecturing). This is an interesting problem that some institutions may need to overcome.

Peer-Learning

Another aspect often discussed in the interviews was the prominence of peer-learning (B_1-B_2, B_4-B_5, B_8-B_{11}, B_{13}-B_{14}), or *"the ability to learn from each other"* (B_5). During project work, participant B_5 states that *"students learn really well from each other"*. When working on projects, particularly group projects, student's work is very open and accessible to other students. In these environments, participant B_5 described what it is like being amongst the project groups: *"we show our problems, we show where we are struggling, and everybody else can see, and either sit back, or learn from us [...] having a shared view of what is going on,*

this is very important". There are a variety of ways that peer-learning can manifest itself, with participant B_4 describing a situation where there is one student *"helping the other, one doing the work and the other just watching"*.

Another approach to peer-learning, sometimes used at participant B_5's institution, is where a student will lead a *"hands-on"* session: *"We call him 'the guy that drives the beamer,' so he's doing some toy or some step-by-step work, solves small problems, that the others. Since he's the one of the beamer, the pace of the work is, sort of, different than what we agreed the teaching assistant would be doing, because we would already know how to do it and people would struggle to keep up. If we have a student drive that, the speed is more the right part, because when he struggles everybody struggles and the others can help him, if he needs help"* (B_5). A student led session *"becomes more of a group discussion"* (B_5). This activity implicitly contains reflective practice; The student leading the session, the beamer, gets *"the feedback at the end, so that they correct their mistakes"* (B_5) – this is reflection-on-action. Furthermore, *"during the whole time they can ask questions, we can support [them]"* (B_5); this is an example of reflection-in-action. A similar style of student-led session was also attempted at participant B_8's university, but the students' feedback on the course was not positive: *"we thought they actually did a reasonable job of presenting the content, but they didn't do a reasonable job of convincing the other students that the delivery could be trusted"* (B_8). For this reason it was dropped from the course, which also had an added benefit because he *"noticed it was also being a distraction, there seems to be this odd thing where students can't see past their next piece of assessment"* (B_8).

Project-Based Learning

Another significant aspect of studio education, according to the participants, is the use of project-based learning (B_1-B_{13}). Studios are seen as *"very hands on"* and *"entirely exercise driven"* (B_2). There is an emphasis on learning-by-doing, with participant B_4 stating: *"the practical experience really seems to deepen their un-*

derstanding of the topic". Whilst discussing project-based learning and pedagogy in general, participant B_3 shared his opinion on how effective studio learning is: *"I was pretty impressed that the studio interactions I saw tend to align pretty well with what we know about how people learn effectively [...] they align much closer with what we know about how people learn than what you would find in a traditional curriculum"*; he refers to traditional teaching as lecture halls or discussion-based seminars.

One of the benefits to students working on projects is that they learn more than just what is prescribed in the curriculum, with participant B_5 saying that *"most of them actually take off and learn new things on their own so that they can be more effective in doing their projects"*. This is supported by the flexible approach that B_5 and other teaching staff take to project-based education, as he stated in his interview that *"in general, we leave them free to work among themselves, as they see fit"* (B_5). He further adds his opinion that *"they need to be capable of doing that, for the rest of their career"* (B_5). All of the participants discussed this to some extent as it relates to those who discussed real-world projects or projects that are defined by the students. Participant B_2 is a prominent example of this as he stated that he has used open-ended project briefs; he gave the broad example that he has had his students *"design an app for 60 to 70 year olds"* (B_2). The students are then expected to *"go interview some people"* and explore the domain (B_2).

Despite the prominence of discussions around project-based learning, one participant shared his scepticism of the importance of projects in a studio by saying *"I wouldn't say [there are] fundamental reasons [that] doing the project in the atelier makes it somehow better. It's just that the whole organisation and set up of the atelier is supported to project-based learning"* (B_5). The participant's discussions on the use of projects varied, with one participant B_3 discussing his experience with how flexible projects are: *"One of the things I was surprised to find was a lot of constraint in studio classes, projects in some cases, are very clearly specified [...]*

it's a complicated balance of constraint and flexibility" (B_3). Many participants described their experience and observations of studios as being largely collaborative in project work (B_1-B_2, B_4-B_{15}), although participant B_3 again shared a slightly contrasting view, stating *"typically the students would be engaged with individual projects"*, showing that studios vary in how much collaboration takes place.

Despite the positivity towards project-based learning, there are numerous challenges that teaching staff face, which are typical for project work and not specific to the use of studios. One of these challenges is mirrored by the flexibility offered to students, as participant B_3 states *"you can't ensure standardisations and uniformity in learning outcomes"*, which he acknowledges that *"it's true of any more open project work in a learning environment"* (B_3). One of the reasons for this specifically is because *"when you have a more open ended project base type of approach, every student is doing a different project, there's no guarantee that all students are learning the same thing [...] that's unavoidable, that's, sort of, a side effect of the approach"* (B_3). Upon further discussion, asking if there are potential solutions to this, he went on to say that this is actually not an issue: *"It turns out they are quite comfortable doing it and they don't consider it to be a problem [...] for the most part the projects that are assigned are highly constrained with various sets parameters that students have to conform to and the parameters are carefully chosen in such a way as to guide student learning along a certain trajectory"* (B_3).

Lastly, participant B_5 discussed the benefit of not having exams in a project-based course because *"in the normal lectures, the pressure comes later, from the exams that are in-between semesters. In the case of ateliers, we don't have any exams, so people just evaluate it based on their projects"*. There can be continual assessment, or simply an assessment of semester/term long activities that form a project.

Real-World Projects

Further to the use of project-based learning, several participants (B_1-B_{12}, B_{14}) discussed the specific use of 'real-world' projects – projects that attempt to simulate real-world practice. The argument, in a similar vein to benefits of project-based learning described above, is that there is more to education than learning the raw facts, and that learning in a simulated real-world context provides additional opportunities to learn. Participant B_4 reasons that *"we are trying to teach end to end, the whole spectrum of the problem, the big picture, not just the little deep topic"*. This, he claims, is *"one of the advantages of the studio"* (B_4). Real-world projects can be quite common, with B_3 stating that inherently *"it's very common for the problems to be, I mean, really closely grounded in the real world"*, and it provides a greater opportunity for the students to engage *"creative design work"* (B_3). With these types of projects a side-effect is that there is a lot going on, something that participant B_3 has observed. He states: *"there's a lot more action in parallel than a traditional classroom which tends to be more serial or sequential with one person talking at a time and everything mediated by the instructor [...] there's a lot of stuff going on simultaneously"* (B_3).

Often with these real-world projects companies may also get involved, taking on the role as a client. Participant B_5 said that *"there are companies that approach the universities"* with project ideas, which the university then approaches their students with to *"propose them to the students. If the students like them, they can directly get in touch with the company"*, they will then *"have to spend time to interact with the customers. This also gives them experience to communicate with"* (B_5). With these types of projects, participant B_5 also thinks *"they're good for student motivation because they get to be motivated to learn if they have to build something"*.

Real-world projects are not uncommon, and although the participants mostly consider it important, they are not necessary for a studio environment. To illustrate this point, participant B_3 draws a comparison between certain types of

studio: *"there aren't really any real world problems with fine art painting, right, but with industrial design, absolutely"*.

4.2.4 Awareness

This category refers to the student's awareness of their surroundings and what other students are doing or have previously done. This was a fairly popular category, with participant B_1, for example, stating that visibly displaying work is *"critical"* to the studio experience. This category has several elements which make it up, all of which cross-over with each other; key points are pulled out and described here.

Externalising Thinking

The participants felt that a significant element of a student's awareness is informed by the activities which externalise their thinking (B_1-B_6, B_8, B_{11}, B_{13}, B_{15}). A common technique that supports this is to display work around the studio. By doing these activities, the students are making certain elements of their work (and thinking) visual, and then displaying it for others to see. Participant B_2 described his use of digital technology to aid this by bringing *"tablets into the classroom for them to actually be able to exchange drawings in a very easy way. [...] the ability to externalise what they're thinking is crucial"*.

Participant B_3 also expressed strong opinions that the studio *"doesn't work if you're not externalising your ideas and continuously doing it [...] I don't think you can do studio without that"*. One of the techniques he described was putting work up on the walls, *"in some cases they're pinning them up on the wall or, you know, it depends on the nature of the activity that particular day"* (B_3). By putting work on walls, and leaving it up, you can aid visual awareness of projects' progress. Participant B_3 also thinks that externalising thinking is *"absolutely central"*.

Another approach is the usage of whiteboards. Participant B_4 specifically discussed his thoughts on how whiteboards aid awareness by enabling and promoting

a culture of sharing: *"with the whiteboards we have it's to encourage public discussion and just share it [...] we have whiteboards all around [...] that is very nice because they have an area to share their design"*.

Participant B_{15} also feels that *"the visibility of work is essential to the critiquing process [...] outside of critiques, this peripheral awareness of what other students are doing and what challenges they're having is critical"*. He also emphasises an obvious problem that software engineering courses have in regards to visibly sharing work: *"To try to simulate that, or facilitate that, in courses that focus primarily on coding, you know, on programming - there's a visibility issue"* (B_{15}).

Peers' Experiences

Something else described by the participants was the awareness of students' peers' experience (B_1-B_2, B_4-B_6, B_8, B_{11}, B_{13}-B_{15}). This can be achieved socially (e.g. through informal interactions) or even through being able to see peers' work. It can also be a side-effect to peer-coaching or peer-learning, for example. Participant B_5's institution has experimented with two techniques for visualising work, *"one of them is a whiteboard. That is a physical way for people to record their sketching and to discuss, and sometimes also to do this with a bigger class"*. There is, however, a more formal technique used by B_5's institution that enables students to observe certain other student's experiences around a given task: *"[...] the other one is, of course, the beamer"*; the 'beamer' is described earlier under 'Peer-Learning' in section 4.2.3. The student who leads one of these sessions, the 'beamer', is given a problem to solve in front of the class; *"his work, his pace, is visible to everybody else"* (B_5). This not only forces that student to externalise their thinking for the class, but it provides an opportunity for all students to reflect on the tasks and how a solution unfolds.

Social Interactions

Studios are social environments which mix *"instruction, course work and social interactions"* (Herrick, 2011, p.65). The interview participants also recognised that awareness of social interactions is important (B_5, B_8-B_{14}). For example, an implicit benefit for students that increase awareness of their work (e.g. by leaving work up on whiteboards) is the open invitation to receive critique, as participant B_4 put it: *"the more you open up what you're doing the more you're open to criticism"*. Informal serendipitous interactions may occur due to work being accessible to others in an environment. Participant B_4 concluded this discussion by saying that *"students shouldn't be shy about showing their ignorance. Actually, the more you expose your ignorance the more chance you have of fixing it"*.

4.2.5 Critique

All of the participants are familiar with the use of critique in studios, and discussed its various manifestations. As an example, participant B_1 was not familiar with the term 'critique' as a studio activity; however, after a brief description of the various ways critique may potentially manifest, he acknowledged that they do indeed do it and support it. On the other hand, participant B_{15} claims that the *"culture of critique"* is the most important aspect of a studio, adding that *"it's a way of perceiving the world"* (B_{15}). He also shows that he has knowledge of the area, as he stated that he is *"particularly inspired by the literature on critiques, especially the work of Donald Schön and his narratives of pretty detailed interactions in the studio between students and instructors"* (B_{15}).

Direct feedback

Direct feedback as part of the learning process, and not simply feedback received after a submission, was considered important (B_1-B_4, B_6, B_9, B_{12}, B_{14}). When incorporated into a critique process, where students would justify ideas, solutions, and processes, they can receive feedback from teaching staff or peers directly during

presentations, demonstrations, or even impromptu interactions. Of course, direct feedback does not strictly need to occur during a critique, as it could potentially be tailored towards discovering weaknesses in a project. One participant described the inherent challenge of providing feedback during her projects, saying that she tries *"giving feedback and giving them ideas, whilst still giving them the ability to manage the process themselves"* (B_{10}).

Developing Ideas

Critique is often used to develop ideas, not just for finding problems with a solution, which several participants acknowledged (B_1-B_5, B_7, B_9-B_{11}, B_{14}-B_{15}). Participant B_2 described the use of a specific exercise called the *"silent sticky note exercise"* to encourage everyone to participate in a critique session. This is where everyone involved in a critique will note down their comments (positive and negative) on a sticky note, then after a presentation or discussion the students place all of their notes on a board or wall at once. This is an attempt to avoid a few people dominating a session, and allows everyone a chance to engage in the critique. It also adds an element of permanence to the feedback, as it is written down.

Multiple Formats

There are a variety of formats that a critique can take place: design jury, design review and desk critique (discussed in greater depth in section 2.2.3). The participants often discussed the use of multiple critique formats (B_1-B_{13}, B_{15}), not just a single type. Although the participants discussed a variety of critique types, they would tend to discuss scheduled presentations or demonstrations (e.g. design juries and design reviews) – more details below. One participant held the opinion that critique in a studio is important because *"it naturally elicits design rationale, which is so important"* (B_{15}). It was also discussed that frequent critiques work better with *"a much smaller number of students, like 15 or 20"* (B_2). This participant claims that frequent critiques did not work in his studio because they *"typically*

Table 4.5: Critique types used by participants

Critique Type	**B_1**	**B_2**	**B_3**	**B_4**	**B_5**	**B_6**	**B_7**	**B_8**	**B_9**	**B_{10}**	**B_{11}**	**B_{12}**	**B_{13}**	**B_{14}**	**B_{15}**
Desk Critique	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓
Design Review			✓	✓	✓		✓		✓	✓	✓		✓	✓	✓
Design Jury	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓

work with about 50 or 60 [students]" and *"you don't have time to go through each individual design in as much detail that a critique would require"* (B_2). To counteract this, B_2 attempts to *"promote that critique from within the group"*. The types of critique discussed by each participant can be seen in Table 4.5.

Design juries are infrequent, formal, and often form part of an assessment. Participant B_1 discussed an example of this type of critique, which were formal presentations at the end each semester: *"when they first see those they think the faculty is tremendously mean because we are very critical of everything they do"*, in an attempt to get the students to justify their work. Not everyone initially looks favourably upon this approach: *"people have looked at our questioning teams during their end of semester presentations akin to throwing a person with bloody cuts on them into shark-infested waters"* (B_1). However, it promotes critical thinking as the students are encouraged to *"justify"* their work (B_1). Participant B_5 also described the use of a jury critique: *"at the end of the semester they do the final project presentations, when they present the work in front of all of their colleagues of the faculty and so on"*. Participant B_{15} discussed his interest and focus on formal critique, stating that *"I haven't really studied the informal critique process"* – as he further clarified, this does not preclude the existence of informal critique in his particular studio, just its explicit existence within set activities.

Design reviews are normally informal and occur frequently. Participant B_4 gave an example where *"every few weeks [the students] present what they've done to the whole class and they get feedback from the whole class"*. Participant B_{15} also provided an example, despite him downplaying the frequency of informal critiques, where *"students were required to develop solutions to algorithms and present them*

for feedback and discussion". The students would present their work at *"the front of a classroom with an LCD projector projecting their solution on to the screen"*, whilst potentially *"a student audience volunteer interacted with their prototype"* (B_{15}). This is done in the style of a group critique because *"the audience can essentially chime in at any time with feedback or suggestions"* (B_{15}).

Desk critiques (the most commonly discussed critique type amongst the interview participants, see Table 4.5) are typically an informal, even spontaneous, one-on-one critique (or even group-on-one). Participant B_4 described how some of these occur in his studio: *"the professor or teaching assistants go around and observe what they're doing and then they comment and critique"*.

Peer Critique

Peer critique, simply critique from your peers, occurs frequently and is encouraged by the participants (B_1-B_{10}, B_{13}-B_{15}). Participant B_1 states that students critique each other's projects at the end of semester project presentations. The students also infrequently *"present to each other"* (B_1). At this specific institution the students engage in a critique activity called '360 peer reviews', at multiple times during a semester, where students are asked (anonymously) what other student projects have done well, what was bad, and what they can improve upon – this specific activity is said to have *"worked extremely well"* (B_1). As the name suggests, *"it's called a 360 because we also want them to review themselves"* (B_1).

Not all participants felt that critique amongst the students is common (B_{11}, B_{12}). When participant B_4 was asked whether there exists a culture of critique amongst the students, outside of set activities, he replied saying that *"some students may do it on their own, but that's because that's their thing, but I don't think the culture exists"*. Peer critique is encouraged by the other participants, and made explicit in certain activities. Participant B_5 specifically discussed the use of peer-critique through open collaboration: *"we manage to establish this team collaboration spirit, in that sense that works, so people are actually criticising each other"*.

Participant B_{15} helped develop a process used for reviewing code *"where small groups of students, led by an experienced moderator who is usually a graduate student in computing, were required to participate in, more or less, the formal code inspection process [...] they basically walk through their code a line at a time looking for problems in the code, and discussing anything that comes up, and brainstorming solutions"*. This process encourages students to critique each other in a formal manner, and can elicit reflective thinking and potential support a richer culture of informal critique outside of these activities. After describing this activity, the participant added that the instructor *"would engage the class in a reflective discussion of the design principles and issues that arose"* (B_{15}) during the review.

Continuous Critique

Several participants have acknowledged the importance of continuous critique (B_1-B_3, B_5-B_6, B_8-B_9, B_{11}-B_{12}, B_{14}), which helps the students reflect on their work. Participant B_3 specifically said that critique is *"constant"* and *"it's critical"*, and also said that it is *"almost like a key defined feature of the studio"* (B_3). This participant also claimed that a particular use of critique was used for *"continuous refinement and commentary on the on-going work"* (B_3). He went further to describe that it is implicit in design processes: *"I think it's not even just the studio, it's the design process even in the professional world"* (B_3). However, not all participants subscribed to the notion of continuous critique (B_4, B_7, B_{10}, B_{13}, B_{15}). As participant B_{15} was quoted as saying earlier, his focus is on formal scheduled critiques; the frequency of these critiques were *"anywhere from just once in the semester to possibly five times a semester, and those are at set intervals depending on when assignments are due"*. He confirmed that they *"aren't spontaneous interactions. Those are definitely, you know, scheduled"* (B_{15}).

Participant B_3 also described a process called 'continuous improvement', which is an agile process of critique: *"There's commitment to, sort of, continuous im-*

provement and the belief that the process itself results in good work [...] I think critique is just one of the ways of contributing to that on-going process of continuous improvement". He added that it is *"the emphasis on getting the work out in to the world as early as possible and learning the process and approach to continually refine the work"* (B_3).

Some participants also claimed that critique occurs constantly (B_1-B_3, B_5-B_6, B_8-B_9, B_{11}-B_{12}, B_{14}), with one participant stating that *"the feedback, at different levels, it happens pretty much continuously"* (B_5). At his institution this is also supported with weekly student milestone presentations, saying that it is *"a way for them to practice and improve, have a lot of opportunities to present and improve what they're doing"* (B_5).

4.2.6 Culture

This category contains all cultural aspects, group aspects, group behaviours and other encultured dynamics.

Sharing

Several of the participants discussed the importance of sharing and openness (B_1, B_4-B_5, B_7, B_{10}, B_{12}-B_{13}). These attitudes also help support other categories, such as awareness and critique.

Participant B_1 provided an example where this took place in his studio where *"team leaders will get together once a week to talk about what's going on and what's working and not".* This provided a platform to critique team dynamics, as well as design decisions.

Participant B_4 claims to explicitly *"try to promote openness".* He also admits that whiteboards are an implementation of the sharing culture: *"with the whiteboards we have, it's to encourage public discussion and just share it"* (B_4).

Social

Social dynamics are considered important to the success of a studio (B_2, B_8-B_{14}). This could be because of the heavy emphasis on all the potential interactions with other people, through group work or critiques, for example.

When graduate students of participant B_2 are asked which course they thought was most important, they said the studio course: *"it taught them stuff about themselves, stuff about how to work together, and also, you know, thinking much deeper than they ever thought they needed to think"*. Certain disciplines will also benefit from a social context, as one participant stated that *"software engineering is inherently a social endeavour"* (B_8).

Good Work Ethic

Students are expected to have a good work ethic in a studio (B_2-B_3, B_5-B_6, B_9, B_{13}-B_{15}), as a large amount of the work can be self-directed and benefits from students engaging with each other about their projects.

One participant, B_3, voiced a view not expressed by the other participants during their interviews. He stated that a studio is not a studio *"if the learners aren't engaged in their own work. I guess that's the one way you can make it be not a studio. If the students are sitting there not doing anything"* (B_3). The emphasis during this portion of his interview was that a non-functioning studio contains students that do not interact with each other, and do not have a good work ethic. Individual work can take place in a studio, and therefore there is limited or no interaction between individuals at times, but all of the participants agree that there needs to be some interaction between the students at some point. This may be achieved through serendipitous interactions, being social, critiques, or group work, to name just a few.

Peer-Support

Peer-support refers to the encouragement (social or study based) and autonomous activities (such as critique or coaching) that may occur amongst students without direct staff intervention – even though this facet of the culture may be facilitated by the staff in the first place. Whereas peer activities (such as peer-learning or peer-coaching) may occur as explicit activities, peer-support here is categorised as the culture and attitudes that are encouraged amongst the students to support each other. It was a common discussion with the interview participants (B_2-B_3, B_6, B_9-B_{11}, B_{14}). Participant B_2, for example, states that peer-critique is encouraged within his studio's culture, so it will occur even when no staff are directly involved.

Permission to Fail

This was a fairly popular aspect talked about during the interviews (B_1, B_3-B_4, B_9, B_{11}-B_{13}). The idea is that in a studio or project course, students have the opportunity to fail at something throughout the course, but rather than it being considered final, they can learn from their mistakes, rectify them, and progress further. Other courses, for example where exams are used, do not necessarily have that leniency built in to assess a recovery from a failure. Participant B_1 succinctly described the intention of this way of thinking: *"if you succeed and don't know why, you've failed, but if you fail and you know why, you've succeeded"*.

Another part of having permission to fail is that students are encouraged to explore, which can give them real-world experience. As participant B_4 explained, when students experience failures, they *"really understand it because they have actually run into problems in their projects using particular protocols"*. A good metaphor was also given by another participant, stating that the mentors *"stop the team from falling over the cliff entirely, but you kind of want them to sort of stare over the precipice. You kind of want them to get to the edge and work out that they're at the edge and why they're at the edge, and ideally have them pull themselves back if something isn't quite working"* (B_9).

4.2.7 Individual's characteristics

This category refers to characteristics of a studio that benefit individuals, as opposed to groups. Several participants (B_1, B_4, B_6, B_9-B_{10}) discussed the need to support students as individual learners in a studio, as well as part of a group. This category was discussed a lot less than in the previous chapter's interviews, with emphasis in this series of interviews instead on team efforts.

Private and quiet spaces

The participants described how private and quiet space can be necessary for students to work (B_1, B_4, B_6, B_9-B_{10}), ensuring they are interrupted less by their peers and their surroundings. This may be necessary for sensitive conversations, for example with project clients or teaching staff. Participant B_4 discussed the necessity for students to have space for them to work as individuals in a studio setting, stating that *"there is definitely a time when they work individually"*. When asked when students may require individual space he replied saying *"during the implementation phase [of a project] they work individually"* (B_4). This also draws attention to the fact that the way students work on a project changes as it progresses.

Noise levels could impede a working environment, with more people in a space will inherently be louder; When a studio does not directly cater for individuals, participant B_4 discussed how the students formulate their own mechanisms for dealing with a given situation. For example, when working individually, *"many of the students [...] have these big earphones so they're not disturbed by the noise outside"* (B_4).

4.2.8 Collaboration

This is another one of the more significantly talked about categories; throughout the interviews portrayed in this chapter, discussions often revolved around collaboration and team efforts. Collaboration is very important in a studio, with

comments stating that the studio is *"mostly for collaboration"* (B_4), and *"especially team work"* (B_5). Participant B_{15} even stated that if you *"remove any kind of collaborative interaction and process"* you will end up with *"a pretty boring, non-studio like environment"*.

Collaboration happens a lot for software developers. Participant B_2 emphasised this point by saying *"There's a bunch of stuff that software developers do on their own, but there's also a heck of a lot that they do together"*. Collaboration occurs *"every day"* (B_2). Participant B_5 goes as far to suggest that teamwork is the most important aspect to studio education.

Impromptu Collaborative Spaces

These are spaces that support collaboration at any given time, either as an always accessible space, or a space that can be easily transformed for collaborative needs. Impromptu collaboration is a desirable part of studio education, according to the interview participants (B_2, B_4, B_8-B_{11}, B_{13}). As participant B_4 explains, *"Since they are in the same area and they are working by themselves and suddenly they pick up some issue they need to talk to somebody else, it's very easy [...] it is easy to get together and go work on a group project"*.

As projects progress, the need for collaborative spaces increases and decreases depending on project demand. When asked what activities benefit from being in a studio, participant B_4 replied saying that *"when you have to work together, when you have to work on a design with other people, so if you have a group which [is] in the early stages you do, and also in the debugging phase"*. Spaces should support collaboration when the students or project require it.

Collaborative spaces can also come in very handy as an attempt to contain noise. Whilst many participants have previously indicated that noise is something that comes with a studio, collaborative spaces can be used to reduce some unnecessary noise or ensure that students don't have to worry about disturbing their peers as much. Participant B_4 described one reason why this is useful: *"people can talk to*

each other without disturbing other people in the room and even yell at each other, I don't know whether it's our students who like to do that but when they get excited they like to raise their voice".

Participant B_5 claimed that the rigidity of a schedule for collaborative activities can be beneficial. He reasoned that *"when you have to work on your projects, people have to meet, so it's good to logistically support this, [with] the schedule. [...] logistically, if you have five or six classes a semester and every class gets a team project, then basically the students will spend all their time chasing each other, and trying to meet"* (B_5). For these reasons, participant B_5 prefers to encourage team work in the studio setting: *"the only class where people are supposed to work on team projects, is the atelier. We, sort of, use this as a container for those kind of activities".* They avoid team projects in other courses, outside of the atelier, and *"is something that we decided to enforce"* (B_5).

Supporting Equipment

As collaboration is an important part of the studio, tools and equipment that support collaboration are also important – which resonated with many participants (B_4-B_8, B_{10}-B_{13}, B_{15}).

Software tools are an obvious area that support people in software engineering courses, with participant B_5 claiming that *"a lot of progress with collaboration tools".* This participant also emphasised the importance of software versioning and repositories, for developers, as they help students *"become productive and effective, in collaboration"* (B_5). He also explained that laptops are an effective tool that aid creating flexible work spaces: *"People take them with them, and sometimes we actually find them in strange places sitting down with their laptops"* (B_5). They can also help create impromptu collaborative spaces.

Putting software tools to one side, non-software approaches are also beneficial. With several participants (B_4-B_5, B_7, B_{11}-B_{13}, B_{15}) frequently discussing the importance and simplicity of using whiteboards. Participant B_4 explains that

whiteboards see more use during certain activities or project phases, stating that *"in the early stages [of a project] they do the brainstorming or idea formation, they work together with the whiteboard"*. In this particular studio, they don't have a few, instead they *"have whiteboards all around, so it is common to see the students standing at the whiteboard together and working out a design"* (B_4). A significant benefit that B_4 sees in using whiteboards is that they *"encourage public discussion and just share it"*. This perspective is shared by participant B_5, who states that *"the people can, in parallel, stay at the whiteboard, and sketch, and discuss things"*.

Group Size Considerations

The participants in this set of interviews discussed group size considerations (B_1-B_2, B_4-B_5, B_7-B_{11}, B_{13}-B_{15}) in conjunction with a previously defined parameter, in the earlier framework under category 'Facilitation of Studio', which referred to what the staff members considered a reasonable cohort group size to encourage – this parameter is distinct in that it refers to the students' group size within a cohort, e.g. working on a project.

At participant B_5's institution, the students experience a variety of individual and group exercises throughout the year: *"Every week they have to do some individual programming assignment, so they get familiar with the foundations. Then, at a certain point we also make them do some pair assignments, and we always switch the pair every time. Then, eventually they get to the teams, which is like four or five people"* (B_5). However within a cohort of students, one participant advocated for larger groups of students working together: *"if you're doing things in small groups, you don't encounter the problems that motivate software engineering practices"* (B_8).

Soft Skills

Soft skills are defined[1] as "personal attributes that enable someone to interact effectively and harmoniously with other people". These skills were described as being integral to becoming a software engineer and being an effective part of a team (B_1-B_3, B_5, B_7-B_8, B_{10}-B_{12}, B_{14}-B_{15}).

Participant B_5 argues that a studio is about learning the subject matter, whilst also *"learning soft skills like teamwork, cooperation, working together and communicating. [... Studios] are good for teaching soft skills [...] and all these things that, usually, in traditional curriculums, they are somehow under-represented"*. He adds that *"computer science students are a kind of shy people, introverts, they are always in front of their computers, so we get them to practice these other things that are also common. I think they are very important in professional life"* (B_5). This emphasis on soft skills can lead to better team dynamics, which in turn can benefit situations like one that participant B_5 describes: *"within the team I think that people are quite open and if there is a problem they point it out then somebody has to fix it"*. This view is reflected by B_{11} who discusses the use of agile processes, specifically burn-down charts, as they help promote and improve soft skills.

In contrast to all the other participants, B_3 stated that the studio is often not collaborative: *"most of the time no, it wasn't collaborative"*; however, he went on to say *"I suppose you could think of the whole environment as collaborative in a sense"* (B_3). Although the students did not work on group projects, he eluded to the fact that the students were collaborative as they created a *"studio culture environment of constructive criticism where they're all trying to help each other become better"* (B_3), inherently improving certain soft skills, such as communication.

Equal Effort

With any work that involves more than one person, questions of the level of contribution inherently ensue. Ensuring equal amounts of effort can be a potential

[1] `http://www.oxforddictionaries.com/definition/english/soft-skills`

issue for any workload that involves multiple people, and was somewhat discussed by the participants (B_2, B_4-B_5, B_8, B_{11}-B_{12}, B_{14}). Participant B_5 discussed how they approach this in his studio, saying that as well as the students presenting their work *"they also have to present how they organised, in terms of collaboration, who does what, so that the pressure is on everybody"*. To do this, he conveys clear expectations of the students, adding that *"we make sure that they know that everybody has to talk, everybody has to present, everybody has to own a part of the project"* (B_5).

The other consideration is how to evaluate group projects. As part of the overall evaluation of a project once it is finished, participant B_5 stated that *"we explicitly value the level of collaboration and the way they organise themselves"*. Making the value of collaboration explicit also ensures that it is considered prior to assessment. However, he admits that *"I still haven't found the right way to evaluate teams"* (B_5).

4.3 Further Analysis

Beyond the analysis of the studio framework, there are some important elements that warrant further analysis. This section reflects on this chapters framework against the essential and desirable parameters identified in the previous chapters framework (see section 3.3.1), and subsequently compares the two frameworks in section 4.3.3. Finally, a summary is presented of the interview participants perspective on what they considered the most important aspect of a studio.

4.3.1 Necessary and Desirable Software Studio Framework Parameters

The participants in this series of interviews discussed many aspects of studio education which they felt contribute to studio education, as seen earlier in Table 4.4. Following the same logic as the previous chapter, certain parameters that fall above

Table 4.6: The software studio framework's dominant parameters

Categories	Occurrences		Parameters
	Study 1*	Study 2†	
Physical environment	14	-	Open environment
	14	10	Flexible space (Reconfigurable furniture)
	5	-	Students control aesthetic factors (lighting, heating)
	15	8	Shared spaces
	8	-	Individual spaces
	7	-	Social spaces
	7	6	Private spaces
	-	8	Multiple spaces
Facilitation of studio	8	-	Studio belongs to the students
	6	-	Staff do not dictate use of space
	7	-	24 hour access
	6	-	Food and drink allowed
	5	8	Availability of multiple expert staff
	5	-	Small group size (about 10)
	-	6	Expected time spent in the studio
	-	5	Define own projects
	-	5	Learning to learn
	-	10	Flexible facilitation
Modes of education	13	9	Flexible learning/Switch approach based on activity
	11	12	Mentoring/coaching
	12	10	Peer-learning
	5	-	Impromptu teaching
	-	13	Project-based learning
	-	13	Real-world projects
Awareness	13	-	Visual work
	13	10	Externalising thinking/Displaying work
	6	-	Visual history of progress
	12	10	Peer's experiences/Easily observe other people's work
	8	8	Social interactions
Critique	12	8	Direct feedback
	11	11	Developing ideas
	11	14	Multiple formats (formal/informal, individual/group)
	10	-	Peer-coaching
	-	13	Peer critique
	-	10	Continuous critique
Culture	10	7	Sharing
	14	8	Social
	6	-	Treated like second home
	10	8	Good work ethic
	6	7	Peer-support
	8	-	Serendipity
	-	7	Permission to fail
Individual's characteristics	9	-	Personalisation of space
	10	5	Private and quiet spaces
Inspiration	13	-	Proximity to other people
	5	-	Relevant available media
	5	-	Library of liked/fun things
	5	-	Playful space
Collaboration	6	7	Impromptu collaborative spaces
	7	10	Supporting equipment
	-	12	Group size considerations
	-	11	Soft skills
	-	7	Equal effort

* Parameters that arose from the analysis of the first interview-based study in chapter 3: interviews with designers, architects, and artists.

† Parameters that arose from this chapters' interview analysis.

a two-thirds majority threshold could be considered as essential by the interview participants. Those that fall below that threshold could be considered desirable. These parameters are highlighted in Table 4.6. The parameters in this table highlighted because the majority of the participants discussed them, however, that does not entail that they are necessarily critical to studio education, just that they were popular discussion points. Aspects that the participants did explicitly state as the most important aspect are presented below in section 4.3.4.

4.3.2 Relationships Between Parameters

This section focuses on relationships between parameters found in this chapters analysis alone. A comparative look between this chapter and the previous one follows in the next section (4.3.3).

Real projects:

The participants in this chapter put a lot of emphasis into 'Project-based learning' (B_1-B_{13}). Of these, almost all (B_1-B_{12}, B_{14}) also talked about how these should simulate or replicate 'Real-world projects', which are projects intended to simulate real-world practice on some dimension (e.g. scale of project, simulated or real clients etc.). Further to this, many of the participants stated that 'Private spaces', are important to these real-world projects, to support the students if they have sensitive projects or are meeting with a client, for example.

Students Justifying Work:

Some participants discussed how the development of students' soft skills was important (B_1-B_3, B_5, B_7-B_8, B_{10}-B_{12}, B_{14}-B_{15}). It was frequently discussed in proximity to the use of critiques to 'Develop ideas' and part of building up a culture of critique amongst the students through 'Peer critique'. During critiques of various types, students are expected to learn to be able to effectively communicate their ideas, problems, and solutions.

Counteracting Noisy Environments:

Similar to the previous chapter's interview participants, this chapter's participants discussed how larger environments can suffer from noise when too many people are co-located in a small space.

To counteract this, some participants suggested the use of 'Private and quiet spaces' (B_1, B_4, B_6, B_9-B_{10}), enabling students to get on with quiet or individual work if they required it. One participant made the observation that his students are already working around this problem by wearing earphones if it gets too noisy in the studio space (B_4).

Flexible Collaboration:

Discussions with the participants about the flexibility of a studio space were quite common (B_2, B_4-B_7, B_{10}-B_{11}, B_{13}-B_{15}). At the same time, most of those participants also talked about 'Supporting equipment' for collaboration (B_4-B_7, B_{10}-B_{11}, B_{13}, B_{15}). A common example within these participants was their discussions around a piece of equipment: whiteboards (B_4-B_5, B_7, B_{11}-B_{13}, B_{15}). Another example includes the use of moveable partitions (B_4, B_{10}), which can be set up and taken down as needed.

Culture of Critique:

When the participants implicitly or explicitly discussed the use of 'Continuous critique' (B_1-B_3, B_5-B_6, B_8-B_9, B_{11}-B_{12}, B_{14}), it was in conjunction with the use of Design Reviews and Desk Critiques (see section 2.2.3 for more information on the types of critique). Design Juries happened infrequently, or at the end of a project (if at all). Participants often discussed encouraging a culture of critique amongst their students, with the intention that 'Peer critique' would occur autonomously and frequently within the studio (B_1-B_3, B_5-B_6, B_8-B_9, B_{14}).

These participants also discussed the use of 'Multiple formats' of critique (B_1-B_3, B_5-B_6, B_8-B_9), to aid with the continuous critique.

Making 'Permission to fail' Feasible:

A somewhat common parameter was providing the students 'Permission to fail' (B_1, B_3-B_4, B_9, B_{11}-B_{13}). The goal for students is normally not to fail at what they are doing, but these participants acknowledged that this was actually an important step to learning when working with complex ideas or projects. The participants often stated that if you know why you have failed at something, then that in itself is a success.

This notion was often discussed by participants that discussed the importance of 'Mentoring/coaching'. To better facilitate this notion of giving permission to fail, the continuously critiquing work and coaching the students should catch problems earlier and provide he students with a valuable learning experience as they navigate around this problem.

4.3.3 Comparing the Two Frameworks

The analysis of the first interview study was data-driven, but the analysis for the second set of interviews (in this chapter) was theory-driven, using the studio framework from the previous chapter as a starting point. This allows for some insights to be drawn from comparing the output from both studies.

A brief overview of the two studies compared can be seen in Table 4.3 or Table 4.6, which lists the number of participants that discussed each parameter for each study. The frameworks have a lot of overlap, discussed below in this section. All of the categories from the studio framework, except 'Inspiration', had concepts emerge from this chapter's analysis. Some of these categories also have strong overlap of the parameters with the studio framework; these include 'Modes of education', 'Awareness', 'Critique', 'Culture', and 'Collaboration'. However, as well as 'Inspiration' (already mentioned), two categories had very little overlap with their parameters: 'Facilitation of studio', and 'Individual's characteristics'.

Whilst this chapters analysis will be biased towards the studio framework in the previous chapter, this was counteracted by asking the same starter questions as

the previous chapter. Furthermore, during the analysis there was a commitment to constant comparison between newly generated codes, concepts, and categories (see section4.1.2 for more details). The intention was to freely allow new concepts and categories to emerge.

Physical environment

This category was significantly discussed in both studies, but there was more emphasis on it in the first study. Two of the prominent parameters from the previous study ('Flexible space' and 'Shared spaces') were also somewhat discussed in this chapters study. However, the third prominent parameter from the previous study ('Open environment') was not discussed during the second study – this could be in part due to the use of computers in the software studios, and how the work is often not visual (as evidenced by the lack of the parameter 'Visual work' in study two). It was more common in the discussions with the participants for them to focus more on the students' work, and how they students collaborated, rather than the environment they were working in. Some participants discussed how a studio space was just a space for the students to touch base, as they would frequently work on their project outside of the studio space. However, both studies did also discuss the importance of private space.

As previously mentioned, 'Open environment' is missing from study two, but also 'Students control aesthetic factors', 'Individual spaces', 'Social spaces'. So the participants in the second study discussed a lesser variety of spaces available to their students.

The final parameter from this chapter's study is 'Multiple spaces', which is new and not discussed in the previous study. The participants from study two expressed the importance of having multiple separate spaces for the studio. However, the previous study participants did not; they would often discuss how a space would be repurposed or transformed for their needs. In some other instances where separate spaces were explicitly discussed, they were spaces such as a workshop for wood

working, and were not the sole property of a studio course, or were even shared across several studio courses and non-studio courses.

Facilitation of studio

This category is one of two categories that contains parameters, but has very poor overlap with the studio framework in the first study. Neither study had dominant parameters in this category. The only parameter to overlap between the two studies is 'Availability of multiple expert staff'.

Despite the lack of overlapping parameters, this category was well represented in the interviews. This is because several new parameters were identified during the coding process, and were suited to this category: 'Expected time spent in the studio', 'Define own projects', 'Learning to learn', and 'Flexible facilitation'.

Modes of education

This category had a strong overlap of parameters with the previous chapter's studio framework. One of the parameters ('Mentoring/coaching') was identified as essential by both studies. Whilst two others ('Flexible learning', and 'Peer-learning') were essential in the first study and not in the second, they are still highly regarded in both.

The parameter 'Impromptu teaching' was only found in the first study, although it is only considered desirable. This however could be because of the second study's highly discussed dominant parameters, 'Project-based learning' and 'Real-world projects'; it is possible that impromptu teaching, to benefit a room or group of students, is less likely when working on different projects, for example when students get to define their own projects.

Awareness

This category had a strong overlap of parameters with the previous chapter's studio framework. Two of the previous study's parameters, 'Displaying work' and aware-

ness of 'Peer's experiences', are dominant parameters. Whilst the second study also had these two parameters, they did not quite make it above the two-thirds majority required to be considered essential. Awareness of 'Social interactions' was also equally discussed in both studies.

There are two other parameters that were not identified in the second study. The first is 'Visual work', which was considered as essential in the first study. This was described by the participants of the second study as not necessary in a software-based disciplines because the type of work is often not considered visual work; this would inherently have knock on effect with other parameters that rely on visual work. The second parameter that was not identified in the second study is 'Visual history of progress'. This is somewhat related to the visual work parameter; if it is perceived that there is not much visual work, then it would be expected that there is no history of it.

Critique

This category had a strong overlap of parameters with the previous chapter's studio framework. The parameter 'Direct feedback' was essential in the first study, but not in the second study (although discussed by many participants, it did not quite go over the two-thirds threshold). Furthermore, the other two essential parameters, 'Developing ideas' and 'Multiple formats', were also discussed a lot in the second study, and considered essential in both.

Next there is two slightly related parameters, the first ('Peer-coaching') was highly discussed, but only in the first study. The other parameter ('Peer critique') was considered essential by the second study's participants. Peer-coaching can have elements of critique (and therefore peer-critique), but the emphasis during the interviews was on how a peer could teach you something. Effectively being coached by someone who may know something and teach it to you. Peer-critique on the other hand assumes that both peers are equal, and critique from one peer to another can come from the perspective of learning and understanding the other

peer's approach to a problem.

One final parameter was created during the analysis of the second study: 'Continuous critique'. The participants in the second study emphasised the importance of continuously critiquing the students, whilst the participants in the first study did not discuss continuous critique. That is not to say that they did not do it, but that it did not come up during the interviews.

Culture

This category had a strong overlap of parameters with the previous chapter's studio framework, with three of the parameters equally represented in both studies: 'Sharing', 'Good work ethic', and 'Peer-support'. Another parameter was also discussed in both studies and was considered essential in the first, but not in the second.

Two other parameters were only found during the first study: 'Treated like second home' and 'Serendipity'. The first of those two could be explained by the reduced discussion into the physical environment, and as discussed previously, several of the participants in the second study did not have dedicated or permanent studio space.

One final parameter that was added during the second study's analysis was 'Permission to fail'. This concept was explicitly discussed by several people, but only in the second study.

Individual's characteristics

This is the second category to have poor overlap with the previous study's framework. It is however a small category with only two parameters. Both of the parameters were highly discussed in the first study, the participants in the second study did not discuss 'Personalisation of space', again, following the trend of not really discussing the physical space that a studio may reside in. However, some did acknowledge the need for 'Private and quiet spaces', but this was primarily in

response to the need to support private space to hold sensitive discussions with (e.g.) clients of the earlier mentioned 'Real-world projects'.

Inspiration – The Missing Category

All of the categories from the previous chapter's interviews were also identified during this chapter's analysis of its interviews, with the exception of this category: 'Inspiration'. There were not enough participants in this chapter discussing the parameters within this category to warrant including it. Only two participants discussed how proximity to other people can provide inspiration (B_5, B_{13}), and only one participant discussed the need for inspirational media within the studio (B_8). However, getting inspiration from your surroundings and other people was significantly discussed with the previous interview participants as an integral part of a design process.

Despite not being discussed during the interviews, it does not indicate that it is not important to the teaching staff. However, it can be inferred that it is less important in relation to the creation and running of a studio in software disciplines when compared to the other categories.

Collaboration

This category had a strong overlap of parameters with the previous chapter's studio framework, but more importantly the analysis of the second study generated three additional parameters. The two parameters identified in the first study were equally discussed in the second study: 'Impromptu collaborative spaces' and 'Supporting equipment'.

Collaboration and group effort was discussed more during the second study's interviews. This also resonates with the 'Individual's characteristics' categories which received significantly less attention by this study's participants. The interview participants in the second study would tend to have a far greater emphasis on studios for use with group and collaborative work. There were three new para-

meters identified during the second study's analysis, supporting the notion that collaborative work was generally a highly discussed topic. The first of these two are deemed essential, 'Group size considerations' and 'Soft skills', and the last was still discussed by several participants ('Equal effort').

4.3.4 Most Important Aspect of a Studio

All of the participants were asked the same final question: *"What is the most important aspect of a studio?"*. In the previous chapter the participants generally agreed that it was "the people in the space" that is the most important aspect of a studio (A_2-A_5, A_8, A_{10}, A_{12}-A_{13}). With this chapter's participants there is a much broader spread of responses, with the slightly more prominent opinion being that the mentor (or teacher) is the most important aspect in studio education. Below is a list summarising the responses.

- **3x** participants said facilitation (the mentor or teacher).
  - *[Participants B_2, B_4, B_{10}]*
- **2x** participants said reflective practice.
  - *[Participants B_1, B_9]*
- **2x** participants said critique.
  - *[Participants B_3, B_{15}]*
- **2x** participants said personalised projects (motivation from custom projects).
  - *[Participants B_5, B_7]*
- **2x** participants said collaboration.
  - *[Participants B_8, B_{11}]*
- **1x** participant said culture (participating in a good work-ethic culture).
  - *[Participant B_6]*

- **1x** participant said communication.
  - *[Participant B_{12}]*
- **1x** participant said openly sharing ideas.
  - *[Participant B_{13}]*
- **1x** participant said peer-learning.
  - *[Participant B_{14}]*

4.4 Conclusions

This chapter presents a qualitative analysis of interviews with university teaching staff in software engineering courses, focusing on RQ2: *"What is the current state of studio education in software engineering?"*. To that end, the interviews were used to ask participants immersed in studio activities and environments about their experiences and opinions on studio education. These interviews were transcribed and analysed, using a similar process to the previous chapter's studio framework, culminating in the creation of a further studio framework in this thesis, the 'software studio framework'; this framework is representative of the current state of studio education in software-based disciplines. It is intended to present a better perspective of studio education within software engineering courses.

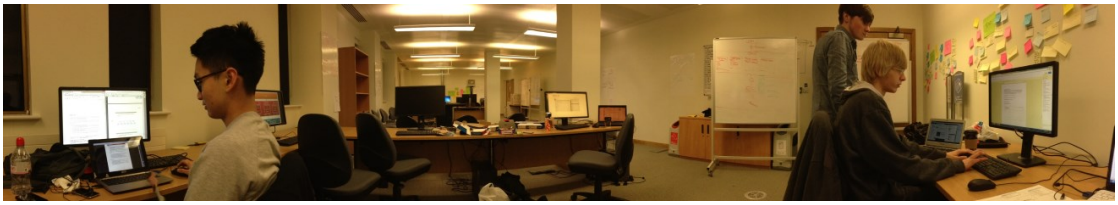
The analysis of the interviews informs us that there are various software studios that support different aspects of studio education. Software studios particularly focus less on the categories 'Inspiration' and 'Individual's characteristics', the latter could be explained by the prominence of group work in software studios. However, no discussions were had during this set of interviews about 'Inspiration' within the studio, which could infer that it is not considered critical to studio education – which is contrary to the opinions of the design studio interview participants in the previous chapter. Alternatively, it could suggest that this part of studio education has not transferred across to the software disciplines.

5 Lancaster University's Software Engineering Studio

During the research presented in this thesis, a software studio was conceived and implemented for software engineering at Lancaster University. This is a new endeavour, and a move away from the traditional lecture-based teaching styles used in the University's School of Computing and Communications. The intention is to offer a dedicated studio, that the students can use like a 'second home', spending the majority of their time there, and forms a central component to their education. This studio is used for Part II undergraduate students (second and third year) studying the Bachelor of Science (BSc) in Software Engineering.

The purpose of this chapter is to provide a description of Lancaster University's software engineering studio in preparation of the next chapter which presents an observational study and analysis of the studio. This chapter therefore describes the studio in its first and second years (2012-2014), including a description of the motivation behind implementing the studio, and how it is implemented. The information presented in this chapter was acquired from a mix of experiences, conversations and documentation regarding the setup and continued growth of

Figure 5.1: A student's panorama of Lancaster University's SE studio.



the studio.

Some of the studio's students have created and published videos online about their studies. These videos cover what the studio looks like and some of the work they do within it[1], and a video describing one of their projects[2].

5.1 Motivation for Lancaster's Studio

Traditionally at Lancaster University teaching spans across two primary approaches to teaching: the predominant lecture, and the oft accompanying computer lab sessions – where content of the lectures can be practised. A computer lab is booked for one or two hours for each of the taught modules that contains a practical element. The computer labs themselves are scheduled for various modules, and are often always filled with students; when one group of students leaves, often another is taking its place. In this regard, the computer lab acts like a central meeting point for many courses. In computer lab sessions students are relatively autonomous, with pre-set tasks, and request help from the teaching assistants when required. These labs appear to be largely a product of their time: a room for the main purpose of supplying computer resources, which is not as useful in the current state of ubiquitous and connected computing.

The software engineering studio was conceived as a course that emphasises hands-on work and experimentation, whilst the students work on large complex problems. To gain greater insight into the reasoning behind the introduction of the software engineering studio, an interview was conducted with Prof. Jon Whittle, who proposed and implemented Lancaster's studio. Whilst discussing his motivations for implementing a studio-based software engineering course, he described his experiences of teaching software engineering prior to the studio: he has been teaching for approximately 10 years, and pointed out that *"when you first start teaching you don't think deeply about how you teach; you think about how you will*

[1] *Smart Lab System - Lancaster University*, www.youtube.com/watch?v=YJ7X3Y_zoJk
[2] *Final Year Software Engineering Group Project*, www.youtube.com/watch?v=pzlAcsL_wto

get through it, or the structure of nice slides". He then went on to say that, at some point, he asked himself *"am I really communicating concepts? What are they actually learning?"* This was followed by a discussion where he stated that he eventually got bored by the traditional mode of lecture-based teaching, realising that lectures may not be the best way to teach software engineering; this was further clarified with a discussion about how learning software engineering in sterile environments can be boring, unless you experience it. It was argued that the benefits of software engineering practices can largely only be experienced on large or complex projects. Finally, the trigger event that caused him to propose a studio approach was an activity at an annual course review where he and some colleagues were asked how they would improve a degree course. The result of that ideation session was this software engineering studio, presented throughout this chapter, which created *"palpable excitement"* in the department.

5.2 Implementation

There are several core elements that make up Lancaster University's studio implementation, including the physical space, equipment and curriculum. However, it is also important to present the studio correctly to other staff members involved. Here is how the studio was implemented and set up.

5.2.1 Guiding Principles

The intent of Lancaster's software studio, as set out by course proposal documents, is to serve as *"both a lab for students engaged in conceiving, designing and developing software products as well as an approach for teaching software engineering in the lab which emphasizes practical hands-on work and experimentation with a variety of software engineering techniques."* Further to this brief description, a simple list of five guiding principles were also provided in the course proposal documents, to better serve the teaching staff in understanding the purpose of the studio and

the roles they would play:

1. *"A focus towards on-demand, self-directed learning that favours practical experimentation over traditional lectures"*
2. *"A cohort approach to training in which groups of students work together in a mutually supportive shared space"*
3. *"An emphasis on experimentation where students gain hands-on knowledge of competing software development processes and techniques and learn how to choose between them based on practical experience"*
4. *"Close collaborative work with academic staff who act as mentors rather than instructors"*
5. *"A focus on underlying principles rather than the fads and notations of the day"*

5.2.2 The Room

Physical space is only part of the studio puzzle, but an important one nonetheless. The room originally assigned for use as a studio was not good enough – it was too small for its intended purpose. So a better one was requested, and obtained. This is likely an inherent issue with the administrative staff of the university not being familiar with what is necessary for a studio room (space to work, present, critique etc.), a sentiment which was also echoed throughout the interviews by the interview participants in chapters 3 & 4.

Figures 5.2 & 5.3 show the studio in a bare-bones state, before term started and before the students made their mark on the room. These two images are not representative of the studio-nature of the room, but instead provide an initial reference point for what the room's skeleton looks like. The subsequent images in Figure 5.4 - 5.7 show the studio after term has started. Of particular note in those images are the use of sticky notes, flipchart paper, and the whiteboards, which contain system designs and team organisation.

Figure 5.2: The SE studio at Lancaster University, before term commenced.

Figure 5.3: The SE studio central area.

Figure 5.4: Overview of the room during term time; during students' free time.

Figure 5.5: The front of the studio, including mounted TV

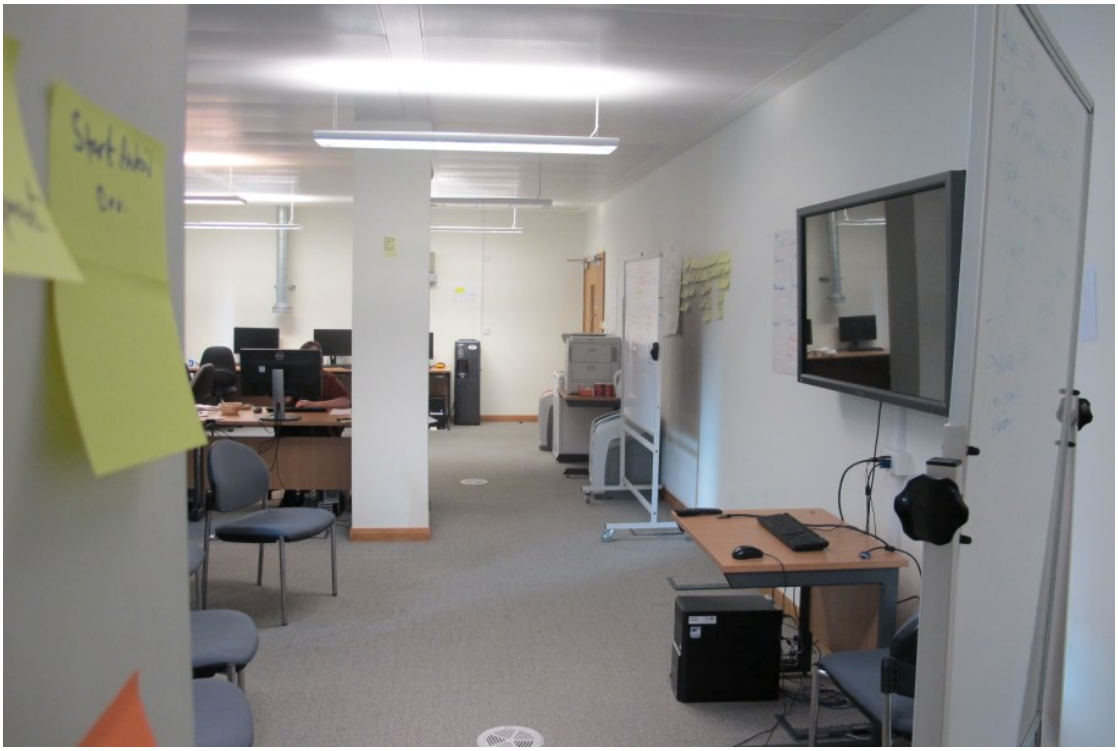
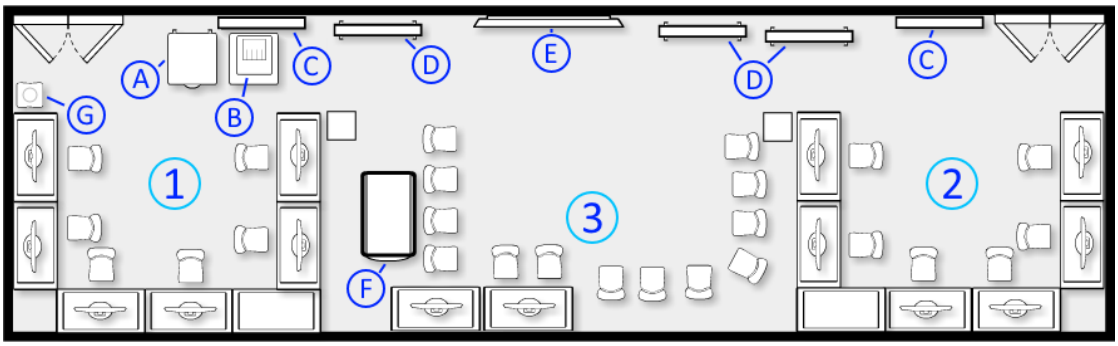


Figure 5.6: Student's using the walls for organising their project.

Figure 5.7: Some studio equipment (fridge, printer, whiteboards)

Figure 5.8: The SE studio floor plan.



Layout

An annotated layout of the studio's floor plan is given in Figure 5.8. The room is split into three main areas: two areas with computers, (1) & (2), are laid out on tables in a 'U' shape (on the left and right side of the room); a central area, (3), contains no desks but has many chairs – providing space for activities away from computer screens. The left and right sections are where the students do the majority of their individual and group work. The central area is used for activities that benefit from the freedom of space, for example: conversations, presentations, and impromptu demonstrations.

Equipment

In addition to the aforementioned desktop computers, the room is furnished with several other features, not all of which are typically found in a computer lab. To provide a space conducive to the amount of time the students will inherently be in the studio, they were provided creature comforts (annotated in Figure 5.8) which include: (A) a fridge, (B) a printer, (C) two wall-mounted whiteboards, (D) three mobile whiteboards, (E) a wall-mounted touch TV, (F) a horizontal touch table, (G) a plumbed water cooler, bookshelves, cupboards, and coat stands. All of these enable the students to work as long as they need in the studio – becoming a more comfortable working environment than a computer lab. The bookshelves and cupboards were removed from the studio during the studio's first year, as they were rarely used. However, this made additional room for the mobile whiteboards

to be manoeuvred around the studio.

At the start of the course the students were provided with a few core pieces of equipment (intended to help with design and collaborative activities): flipchart paper, Post-it® notes and Magic Whiteboard™ sheets. The students increased their usage of these as the year progressed, shortly becoming an integral part of their project experience. The whiteboards on the other hand, especially the three mobile ones introduced after a few weeks, were used always used. These mobile whiteboards could also be moved to change the dynamic of the space. For example, bringing one into the 'U' shaped work spaces, splitting it down the middle and altering the dynamic of how the collaboration takes place.

Rules

As with any space, there are rules that govern its use. However, the studio offers generally more flexible rules, reinforcing the notion that the students own their studio space. To give you an idea of how the students were expected to treat the studio, here are some of the prominent rules expressed to the students by the teaching staff during the studio's first session:

- The students were allowed to eat and drink in this studio – this is not commonly allowed in computer labs, but it can be beneficial (Herrick, 2012).
- 24-hour keycard access to the room was given to the SE students and staff. The students were also allowed to bring in friends if they desired.
- They were allowed to use the room socially. This was, in fact, encouraged.
- Finally, the students were encouraged to rearrange the room's furniture and computers if they wanted to, or if they thought it would help their projects or group dynamics.

5.3 The Course and Curriculum

The degree provides three compulsory studio-based modules for second and third year software engineering students, described below (see Table 5.1 for a brief summary of the differences). Each of the studio modules involve a group project, with teams of four to six students, and only consist of coursework-based assessments. The degree also shares core modules with the computer science degrees, to more efficiently teach essential skills and concepts that are common across the degrees. All first year software engineering and computer science students do the same traditional lecture-based initial year – they specialise and diverge once entering their second year.

The studio is not implemented across the entire degree, instead it is implemented as three distinct modules over the students' second and third years: In the students' second year they participate in one module, *SCC.230 Core Studio* (this is the main module observed later in this thesis), and in the third year they participate in two modules, *SCC.330 Networked Studio* (also observed) followed by *SCC.331 Live Studio*. During an interview with Prof. Jon Whittle he was asked why the studio was implemented using modules within the degree (sharing other modules with the CS degrees), as opposed to implementing the entire software engineering degree as studio-based. He stated that the reason for using studio modules was because it was more cost-effective to share modules with the CS degree, and also because of the practicalities of implementing a degree, integrating it with the current curriculum and introducing new teaching techniques. Although the studio is module-based, *"the studio modules are a significant part of the Part II teaching. Although only three modules, two of them [the third year modules] are double credit modules. So in [the students' second and third years], students take a total of 120 credits per year. In year two, only 15 of these are studio credits, but in year three, 60 credits are studio credits."* He further went on to state that *"the real reason for choosing [module-based studios] is practicality. SE students still need to learn basic CS skills, and it makes sense to share those modules with CS. Otherwise, we*

Table 5.1: Summary of the studio modules

Studio Module	**Summary**
SCC.230 Core Studio *Second year,* *Term one & two*	**Project** • Student-defined projects. • Development process: Iterative/Incremental • Introduced to broader software engineering concepts **Assessment and Deliverables** • 100% Coursework (40% group, 60% individual) • Group report, working demo and/or oral presentation • Individual portfolio (personal contributions) • Project mentor's report on each student
SCC.330 Networked Studio *Third year,* *Term one*	**Project** • Pre-defined project: The smart lab project • Development process: Introduction to Scrum • Predetermined weekly Scrum-like sprints • SE techniques for medium scale networked projects • More realistic requirements • Integration and networking of software modules **Assessment and Deliverables** • 100% Coursework (30% group, 70% individual) • Reflective reports • Weekly group and individual assessments, including: ⇒ Group demonstrations ⇒ One-to-one feedback
SCC.331 Live Studio *Third year,* *Term one*	**Project** • Pre-defined project: The Smart Environment Toolkit • Development process: Scrum • Emphasis on disciplined development • Emphasis on real-world practice, including: ⇒ Business context ⇒ Software engineering principles ⇒ Professional competency **Assessment and Deliverables** • 100% Coursework (20% group, 80% individual) • Focus on individuals' attainment of SE principles. • Group demo and oral presentation • Individual portfolio (personal contributions) • Individual demo sessions

wouldn't have the resources to do everything. I also think that not all subjects are suited to a studio... although could be convinced." This is a sentiment that has been reflected by others, though it does appear to be an unanswered question – which subjects are best suited to studio education? With one final word on the matter, Prof. Whittle concluded that *"if we had studio teaching in our CS degree, it would be a different matter. We could then consider a fully studio based CS degree."*

Second year students also take SCC204 Software Design alongside their second year studio module, a non-studio lecture-based software engineering module shared with computer science students. It provides them a basic theoretical underpinning of software engineering concepts.

The studio modules have no lectures, and are 100% coursework-based assessment. Instead of lectures, the emphasis is on coaching and regular workshops, which are intended to be interactive sessions, with a variety of staff present. The students are allowed and expected to use the studio outside of their timetabled sessions. A lot of emphasis is put on the significance of the studio module, and a large amount of self-study is necessary outside of the workshops.

Below, in sections 5.3.1 to 5.3.3, the individual studio modules are briefly described. The second year module (SCC.230 Core Studio) and the third year module (SCC.330 Networked Studio) will be described in further detail in chapter 6 with descriptions and discussions of the observations carried out in the studio throughout the academic year 2012-2013 and into the academic year 2013-2014.

5.3.1 SCC.230 Core Studio

In this second year module the students are introduced to the studio. The students put a variety of learned software engineering techniques into practice in a group project, giving them hands-on practical experience within small groups. Projects are carried out in small groups of four students each, and are defined by the students. The students develop ideas for their project at the start of the module

with guidance from members of staff to ensure that there is suitable coverage of the curriculum. It has timetabled studio sessions, called 'workshops', every two weeks, for three and a half hours over the first two terms (20 weeks, October-April). Group supervisors also meet with their groups weekly. This module is the main module observed in this thesis in chapter 6. This module is a 15 credit module – the students are expected to put in approximately 150 hours of work in this module.

Whilst working on this module the students are also enrolled on 'SCC.204 Software Design', which is ran in parallel to the 'Core Studio' module for the first 10 weeks. It is a compulsory second year module, which is not a studio, lecture-based, and an introductory software engineering module. It is shared with computer science students.

The timetabled workshop sessions are used by the students to learn and practice software engineering processes by using or integrating them into their projects. The students work on their projects within each workshop, but each session has a theme that the students must focus on. The workshops focus on both technical skills (e.g. requirements, design, testing etc.) and transferable skills (e.g. project management, presentation skills). The transferable skills, or non-technical soft skills, are expected to make better software engineers (Jazayeri, 2004). As an example, the focus of one of the workshops is entirely on presentation skills, which is taught by an actor (Lee et al., 2015). They are also an opportunity for the staff to touch base with the students. These sessions are geared towards coaching and reflective practice, and have a high staff to student ratio when compared to traditional lecture-based sessions – even when taking the lecture's accompanying practical sessions into consideration. A typical workshop contains anywhere between one to four lecturers (usually two or three) and one teaching assistant. Whilst the studio appears to require a high level of staff input, it is balanced by the reduced time required of lecturers, both in preparation and attendance of fewer overall sessions, i.e. attending one studio session every week, as opposed to attend-

ing a lecture and multiple practical computer lab sessions. Studios also encourage lecturers to focus on flexible in-session action rather preparation of materials beforehand. The studio houses approximately 10 to 15 students per year group, with the 2014 intake of this module consisting of 16 students. Any large scale expansion would require further research to determine the effect on staff resources.

As the students defined their own projects, it provided them with a greater sense of ownership of the course. This also meant that all of the groups worked on vastly different projects, which were therefore not sand-boxed, and had the opportunity to take it further and deploy their solutions outside of the course if they desired. The projects were monitored by staff through weekly supervision meetings, as well as by all staff present during each workshop session, to ensure that the curriculum was covered in each project. Outside of these workshops, the students still spend a significant amount of time in the studio, working under their own steam.

5.3.2 SCC.330 Networked Studio

In this third year module the students work on larger projects. These projects are expected to be more complex than the previous years projects, with more realistic requirements, and more closely resemble the scale of an industrial project.

The project selected for the students is a Smart Lab System; this project aims to turn the software engineering studio into an interactive smart lab. This is achieved through designing a system which contains sensors and actuators. A common sensor that the students are supplied with is the Oracle SunSPOTs, shown in Figure 5.9. It is an Internet-of-Things device with several sensors and connectivity[3]). The smart lab projects consist of services to monitor and collect information about the environment, for example including occupancy numbers and light levels. These services are then integrated into smart lab applications. This module was also observed as part of the analysis in chapter 6. This module is a 30 credit module – the students are expected to put in approximately 300 hours of work in this module.

[3]Official SunSPOT website – `http://sunspotworld.com/`

Figure 5.9: The SunSPOT devices.

This year the students are introduced to SCRUM, an agile software development process, and they use it throughout this module (10 weeks long). Each week of term is a SCRUM-like sprint. Each sprint is planned in regards to workload and outputs, with one of the students acting in the role of SCRUM Master, managing that week's sprint. A different student rotates into the role each week, spreading the experience of managing a software project.

Each sprint consists of two main activities, where staff are present: weekly stand-up meetings and weekly workshops. This contrasts to last year where workshops were held every two weeks, but is necessary because of the pace of the weekly sprints.

The stand-up meetings are intended to be short meetings with a mentor to discuss progress of the project in that week's sprint. This is an opportunity for the member of staff to provide guidance if necessary, but is primarily used for the students to reflect on the week's progress and what is left to be done before the

next timetabled workshop session.

The workshop sessions are for the students to seek help from members of staff, but are primarily used for demonstrating, presenting, critiquing, and marking the projects. With it being a project-led module, the students are expected to seek help on their own, emphasising a self-learning process. Though they do have plenty of timetabled interaction with members of staff.

5.3.3 SCC.331 Live Studio

This third year module follows on from the previous one (SCC.330 Networked Studio), as the students extend their Smart Lab Systems and develop toolkits for smart living (Smart Environment Toolkit). They experience non-trivial industrial size and strength projects, with a focus on building a live system that will be deployed and could have commercial or research value. The module has a similar foundation to the previous one in that agile development methods are employed, and the workshops are used to demonstrate progress each week. This module is a 30 credit module – the students are expected to put in approximately 300 hours of work in this module.

The students are familiar with the agile development from the previous module, and it is utilised again within this module. The approach to the agile development in this module contains a strong emphasis on disciplined development, the importance of software engineering principles, and real-world practice. This is achieved by focusing on the business context, focusing on software engineering practices and principles, and professional competency.

This module has a focus on individual attainment, which is supported by a software engineering badge system (Lee et al., 2015). Each badge corresponds to a specific software engineering principle. This badge system allows students to be assessed on a particular subset of software engineering principles that they choose, allowing students to specialise their assessment, but is considered in addition to core assessment criteria and expectations. The students must complete the re-

quirements for two compulsory badge, the Personal Software Process (PSP) levels 0 and 0.1, but have freedom to select two others from seven predefined badges:

- Refactoring
- Test-driven development
- Reverse engineering software architecture
- Design patterns
- Automated testing
- Usability testing
- Code quality assessment by metrics

5.4 Summary

The software engineering studio at Lancaster University has been designed around the idea of experiencing the software engineering principles first-hand and emphasises experimentation. All studio modules are formed around group projects (with groups of four to six students), and take place in the students second and third year of their software engineering degrees. These build upon knowledge and experience obtained in other modules of their degree, including first year 'SCC.110 Software Development', and second year 'SCC.204 Software Design'.

The studio modules focus on technical software engineering skills (requirements, design, implementation, and testing), as well as transferable skills (such as reflective practice and presentation skills). These skills are put into practice with increasingly larger and complex projects as they enter each new studio module, with the third year modules (SCC.330 and SCC.331) being closer in scale to industrial problems (Lee et al., 2015). The inclusion of agile development methodologies in the third year studio modules also allowed reflective practice to be built into all

of the weekly sprints, enabling students the opportunity to reflect and improve for each subsequent week.

6 Analysis of Lancaster University's Studio

Existing software studios state that they are indeed studios, and achieve that status by integrating some activities or principles that are common in design studios, for example certain types of critiques or a focus on reflective practice (see section 2.3). These software studios also often report that the courses are a success. The problem is that there is not a widely accepted definition or understanding of what precisely constitutes studio education (chapters 3 and 4 aim to resolve this), which is evidenced in the variety of implementation details of each studio. Therefore, it is unclear whether those environments closely represent studio education, and whether studio education can indeed be transferred across to software engineering.

RQ3) Does studio education transfer to software engineering education?

To answer this question an examination of Lancaster University's software engineering studio was performed, and the analysis was correlated against the newly defined studio framework (Table 4.3). Following on from chapter 5, where Lancaster University's software engineering studio was introduced, this chapter describes research focused on that studio implementation. Observations of the studio were made during its initial year, with these observations intended to gather information on how the studio was implemented and make comparisons with the studio framework. A focus group with the studio's students was also performed

to complement the observations. These primary data capture techniques were performed on the studio's initial year (with second year students). To further complement this data, observations were continued partially into the next year (following the students into their third year).

6.1 Methodology

Data gathering was conducted throughout the academic year of 2012-2013 (October '12 - March '13), with additional observations taken for half of SCC.330 (November '13 - December '13); these primarily involved observations (specifically, a type of ethnography called participant observation) and is complemented by a focus group with the students. This section describes the approach to participant observation and the focus group, and then a summary on how these approaches and their participants are annotated throughout this chapter.

6.1.1 Participant Observation

Participant observation (DeWalt et al., 1998) was the primary method of information gathering on the studio. Observations were made and field notes taken whilst the author was immersed in the environment as a teaching assistant (TA), as well as that of the observer, not simply as an onlooker. By taking on both roles simultaneously, it allowed the observer to perform 'moderate participation' (DeWalt and DeWalt, 2002). Being a TA to the students allows a mixture of direct involvement with the students, but also provides a level of detachment from them, allowing for objectivity during the sessions (as TAs do not constantly interact with students). The students were made aware that this particular TA would also be observing. Observations took place throughout the entire second year module (SCC.230 Core Studio) over a 20 week period (during the workshop sessions).

Observing as a TA provided the ability to follow Howell (1972)'s phases of participant observation: 1) establish rapport, 2) submerge into community, 3) record

observations, and 4) analyse data. Establishing a rapport is particularly important, as it is described as *"an essential element in using participant observation as a tool as well as the goal of participant observation"* (DeWalt and DeWalt, 2011, p.47). Moderate participation does not lend itself to establishing a rapport and submerging into the community as much as greater levels of participation (i.e. active or complete participation); an example of complete participation would be to have a student within the studio performing the observations. However, in the author's experience, TAs build a strong rapport with their students. Being perceived and utilised as a TA allows observation of behaviour from a teaching perspective, yet TAs are generally not strictly treated as an authority figure. TAs frequently have friendly interactions with the students, and are very approachable.

The field notes taken during the participant observation were analysed using the same process as the interview transcripts in chapters 3 and 4, see section 3.1.3 for a full description. The observations in the field notes are coded and grouped around emerging themes in the data. These themes are described in this chapter and reflected against the studio framework.

In addition to the observations of SCC.230, the first term of the subsequent academic year (2013-2014) was also observed; continuing the observation of the same group of students into their third year. This was to provide a contrast to the previous year's observations by seeing the students within a new module (SCC.330 Networked Studio), working on new projects, with different team dynamics. Time constraints only permitted the final five weeks of this term to be observed, skipping the first five weeks. However, this allowed the students to settle into their new projects and get used to sharing the studio with the new second year students before observations in that year started.

6.1.2 Focus Group

The focus group was used to complement the observations of the studio. The focus group took place prior to the students' third year, towards the end of their

second year, and was approximately two hours long. It was deliberately organised before any deadlines usually associated with the end of the year started to mount up, so it was not too disruptive for the students. To achieve a variety of input from them which was representative of all the students in the course, at least one student from each of the three project groups was invited to participate, as well as the course's student representative; the four invited students were 'S_1', 'S_5', 'S_9', and 'S_{10}'. The purpose of the focus group with the students was twofold: 1) to discover what, in their opinion, worked or did not work in the studio, and 2) look for student opinions that would support or contradict observations made throughout the year. Some observations were shared with the students during the focus group to stimulate discussion.

The focus group was designed around semi-formal discussions, with some questions and discussion points written down before the session (listed in *Appendix A: Focus Group Questions and Discussion Points*); their purpose was to start conversations on a range of topics, and was informed by observations in the field notes, and grouped under the studio framework categories. However, the students were encouraged to discuss anything they deemed relevant, so the conversation topics were not constrained by the questions. The students' conversations would then suggest what they thought was important.

6.1.3 Annotating Research Participants

Quotations, observations, and references to participants are annotated throughout this chapter. As the participants are anonymous, this style of annotation provides a way to follow the participants' identities. The technique used also allows for succinct references and quick identification of certain elements of research (i.e. an interview participant, or an observed student). An introduction to these annotations is given in Table 6.1, followed by some examples.

When the interview participants from previous chapters are mentioned in this chapter, they reuse the same notations that were assigned in their original chapters,

Table 6.1: Legend for quotation annotations

Research Activity	
Annotation	**Description**
Obs[*course*:*week*]	**Obs**ervations of a specific studio session E.g. "Obs[230:5]" is module '230', week number '5'.†
Fg	**F**ocus **g**roup
Unique Participant Identifiers	
S_1	**S**tudents on the studio course at Lancaster University
T_1	**T**eaching staff in Lancaster University's studio

† Week numbers are used instead of explicit session numbers; some modules have multiple sessions in a week (e.g. main session/workshop and stand-up meeting), and some observations are also made outside of a studio session (in the students' "free time").

for consistency. Briefly re-iterating what is mentioned in the 'Interview Participants' sections of chapters 3 and 4, neither sets of those interviews shared participants between the chapters; each participant ID refers to a unique person, that is to say that interview participant A_1 is not the same person as participant B_1. There are a total of thirty interview participants: A_1 - A_{15} and B_1 - B_{15}. The letter 'A' (e.g. A_4) signifies that the participant is from the first set of interviews from chapter 3, 'Understanding Studios: The Studio Framework'. The letter 'B' (e.g. B_4) signifies that the participant is from the second set of interviews from chapter 4, 'State of Software Studios in Universities'.

The annotations consist of the research activity, separated with a comma, and then followed by a unique identifier that represents a research participant, e.g. "(Fg,S_9)" or "(Obs[330:9],T_2)". Some annotations may not include a reference to a specific person; for example, a general observation of a studio session (such as SCC.230, in week 7) would be annotated without reference to a person like the following: "(Obs[230:7])". Here is an example of one of these annotations in use:

"The studio is, like, I almost feel like it's our little space, kinda thing. And yeah, I'd bring one or two friends around, if they want help with

something, but usually whenever I need to do any work, I'm literally in there. I think I go almost every day, except for the weekends"
(Fg,S_5).

In total, there were twelve students at Lancaster University on the studio courses in the observed year group (S_1 - S_{12}). There were also seven teaching staff who participated over the course of the two years (T_1 - T_7), not including the author.

6.2 Analysis: Observation Themes

The analysis of the participant observation field notes resulted in a list of observations that were grouped around themes, presented in Table 6.2. Each of the observations listed in the table have at least one associated category from the studio framework (see Table 4.3). Each of these observations is described in more detail in the next section (6.3, page 167), specifically with regards to its association with the studio framework.

The analysis of the field notes and focus group also offered some general observations that do not fit in the studio framework, which are discussed in the following section, 6.2.2.

6.2.1 Observation Themes

Below are brief descriptions of each of the themes that emerged from the participant observation field notes. Each of the following themes are expanded upon in the next section, as each observation within a theme (Table 6.2) is described and related to the studio framework.

Defining Own Projects

The students defined their own projects in SCC.230. This had several benefits as well as driving engagement and enthusiasm for the projects.

Table 6.2: Themes from participant observation's field notes

Themes	**Observations**[†]
Defining Own Projects	• Students favour tasks that they have experience with. *[Facilitation]* • Giving students a specification vs. not. *[Facilitation]* • Reduced risk of plagiarism. *[Culture]*
Role of Software Models	• Hacking vs. software architecture. *[Modes of Edu.]*
Theory vs. Practice	• Flipped classroom. *[Modes of Edu.]* • Project courses. *[Modes of Edu.]*
Staff	• Managing expectations. *[Facilitation]* • Mentoring/coaching. *[Modes of Edu.]* • Multiple staff per session. *[Facilitation]* • Breaking down barriers between staff and students. *[Culture]* • Staff need to be maestros. *[Facilitation]* • Enthusiasm of staff. *[Culture]* • Task: Requirements change challenge. *[Modes of Edu.]*
Student Engagement	• Procrastination and breaks. *[Culture]* • Timetabled vs. non-timetabled interactions. *[Facilitation]* • Mobile whiteboards. *[Awareness]*
Student Learning	• Peer learning. *[Modes of Edu.]* • On-demand learning. *[Modes of Edu.]* • Learning from failure. *[Culture]* • Learning how to communicate. *[Collaboration]* • Learning how to learn. *[Facilitation]*
Group Dynamics	• Interaction across groups. *[Culture]* • Importance of a group leader. *[Collaboration]* • Team working. *[Collaboration]* • Collaborative document writing. *[Collaboration]* • Limited lurkers. *[Facilitation, Physical]* • Competition between project groups. *[Culture, Inspiration]* • Using social networks as a file sharing tool. *[Collaboration]*
Interacting with Digital Media	• Touch screens. *[Collaboration, Critique]* • How people sit around a monitor. *[Physical]*
Physical Space	• Permanence of design work. *[Awareness, Physical]* • Visibility of design work. *[Awareness]* • Sense of home. *[Culture]* • Sleeping in the studio. *[Individual]*
Agile Methods	• Delivery of product at each iteration (SCC.330/331). *[Critique]* • Continuous integration/deployment. *[Collaboration]* • Naturalness of design iteration. *[Critique]* • Pair programming. *[Collaboration]* • Burn-down charts. *[Collaboration]* • Stand-up meetings. *[Modes of Edu.]*
Critique	• Stressfulness of critique. *[Critique]* • Constant critiquing. *[Critique]* • Peer critique. *[Critique]*
Assessment	• Group assessment. *[Collaboration]* • Frequency of assessment. *[Modes of Edu.]* • Role of video integration. *[Facilitation]*

[†] Each observation has an associated studio framework category listed within the square brackets (see Table 4.3 for details of the categories).

Role of Software Models

There was very little use of formal software models when the students took control of setting their milestones and deliverables (Obs[SCC.330:1-10]). Although formal models were rarely used, *"informal models are relied on heavily"* (Whittle et al., 2014, p.2). Informal models are not created using professional software tools (opting for whiteboards or a pen and paper approach) and may not strictly adhere to formal notations. However, the *"physical studio allows students to display their informal models, which, although they are generally not updated as the design changes, act as a visible reminder as to how design decisions were arrived at and therefore continue to serve as a point of reference"* (Whittle et al., 2014, p.2).

Theory vs. Practice

Students attended lectures for other modules, and had the opportunity to apply the theory from some or all of these in the studio. A prominent example at Lancaster is the students' attendance at SCC.204 'Software Design' (briefly described in section 5.3) – this lecture-based course runs in parallel to SCC.230, but just in the first ten weeks of the first studio year (the second year of the students' degree). This allows the students to experience a broader set of software engineering principles (as a project-led course will only require a certain set of software engineering techniques to be practised). Students liked this approach, claiming that they could practice some of the theory that they have learnt in other modules (Fg,S_5). In an interview with T_1, he stated that this made it easier to integrate the studio into existing degree structures, as opposed to introducing a completely new studio-only degree. However, it is not known whether a studio-only approach is appropriate to theory-heavy subjects such as software engineering.

Staff

In Lancaster's studio the staff are a valuable asset. The studio framework describes how staff create and complement the culture that is pervasive within a studio, and

studio style of teaching was new to the staff as well as the students.

Student Engagement

Students appear very engaged in the studio. Depending on the amount of assigned coursework in other modules, the students would spend a lot of time in the studio. The students would also sometimes use the studio to work on other modules' coursework: *"we used it several times a day, just for normal coursework and other coursework as well"* (Fg,S_1).

A prominent reason for the students' engagement during SCC.230 was due to them defining their own projects and *"the appeal of, like, making your own application"* (Fg,S_1). The students agreed with a statement made by one of the focus group participants, in that *"if you are going to make something that no one else is going to really use [...] it's not very motivating"* (Fg,S_1). However, this could also go the other way, as not all students in a group may be invested in a project idea.

Student Learning

Learning comes in various forms in Lancaster's studio, including peer learning, on-demand learning, and improving soft skills to name a few. The various and flexible learning techniques resonate with the studio framework.

Group Dynamics

The studio was intended to be highly collaborative by design, but the space also benefited from the students being very social. With the amount of time the students spent together, there are many observations regarding their group dynamics.

Interacting with Digital Media

The students had access to, and owned, various digital media: the studio's desktop computers, their own laptops, personal tablets and phones, wall-mounted touch

TV (with connected computer), and touch table (a horizontally-mounted touch screen). The variety of devices allowed the students to work as they required.

Physical Space

The physical space forms a literal foundation that supports the studio culture. Students felt ownership over the space, and consequently treated it with respect, with one student remarking that *"because there's that, freedom we respected it"* (Fg,S_{10}).

Agile Methods

One of the most significant differences between the second year module (SCC.230) and the third year modules (SCC.330 and SCC.331) was the agile development methodology employed: a version of SCRUM. More descriptions on this are given in section 5.3.2. Each week a different student acted as the SCRUM master, taking on leadership responsibilities. All the students had an opportunity to do it, fitting well with the SCRUM sprints. Feedback given to the SCRUM masters often included how to be a better leader, even though it is not explicitly graded in this course. A few of the students who shied away from leadership or presenting work last year were put in charge for at least one of the weekly sprints during those modules. This had a noticeable affect on their interactions within their group, for example, with them leading conversations during the stand-up meetings.

Critique

Throughout the studio course, the students gave several presentations and demonstrations of their work and solutions to the staff and other students within the studio. These short, often semi-formal, presentations provided further opportunities for them to engage with each other, encouraging critique across groups and providing yet another avenue for constant questioning. Most questions were not aimed at simply how students performed tasks but were crafted to ask "how" and

"why". The audience of these presentations were also encouraged to reflect on the presentation and questions, which was observed when students incorporated that interaction into their thinking, such as seeing how another group approached a particular problem.

Assessment

With the introduction of the studio course there is considerable consideration towards assessing the students. The students are assessed individually and as their groups.

6.2.2 General Observations

Some observations arose from the analysis of the field notes and focus group which do not fit in with the studio framework. These observations still offer valuable insight

First Year as a Proving Ground

The first year, SCC.230, is seen as a proving ground for the students in preparation for the second year (SCC.330 and SCC.331).

Something often seen with students when they initially start in a studio is that *"in early stages of the practicum, confusion and mystery reign"* (Schön, 1987, p.20). This is good reasoning to focus on introducing the students to the studio. In an interview with T_1, it was also remarked that the staff and students took a year to get used to *"palpable energy"*. As the studio progressed, the students gained increased motivation and enthusiasm for the studio. The students opened up throughout their degree, showing increased confidence and maturity.

Happiness Of Students

Enjoyment of university courses is very important and can affect engagement and motivation. Generally the students were always seen as happy – the groups have a

social working ethic. Many of the students enjoyed the studio modules more than all others that they took (Obs[330:10],S_{11}). This was further discussed in the focus group. Staff got the impression the students were enjoying the course before any course feedback was received; in a conversation with T_1 between studio sessions he exclaimed: *"When was the last time you had a student asking for more work?"*, as a student had requested more to do on top of waht was already expected.

Training The Staff

Studios were new to the staff, as well as the students, during its first year. It took staff and students about a year to truly get used to the studio, with the first term especially emphasising the process of getting used to the new approach of studio education. Some staff on occasion fell back to a lecture mode of teaching. During reflections on how the studio has progressed, in an interview with T_1, he stated that *"not all staff 'got it' straight away"*. This was followed by a discussion where he clarified that this was not a criticism of the staff, but an ongoing process of learning for the teachers, as well as the students. This could be overcome by *"training the trainer"* – an approach adopted at participant B_{12}'s university. During a discussion with this participant, he stated that people are specifically trained to be able to coach students in their studio.

6.3 Analysis: Reflecting on the Studio Framework

Observations and reflections on the studio were made frequently over the course of the participant observation. The observations were analysed by looking for emergent themes. These were grouped together to form categories of observations, or 'themes' (see Table 6.2).

The following sections (6.3.x) are titled as the studio framework categories. Each section contains a brief description of the studio in regards to that category, and then lists observations from the analysis of the participant observation field notes that relate to that category (the 'Observations' from Table 6.2). These

observations are placed within the context of the studio framework categories listed below, from chapter 3 and 4, with each briefly describing how Lancaster's studio has satisfied that particular category, or not, with concrete examples given.

6.3.1 Physical Environment

The room is laid out to be spacious and supports a lot of movement. The layout of the room, shown earlier in figure 5.8, was designed to support a variety of activities, and is particularly suited to group activities. It is also intended to be flexible, with equipment and furniture expected to be moved around the room to suit the students' needs. The room is secured by card access, which ensures that the students can leave personal or work items in the room and on the sides, providing an element of comfort.

How People Sit Around A Monitor

Observation Theme: Interacting with Digital Media

Framework Parameter: Flexible space

Students would often create multi-monitor setups, even though the room is geared towards one monitor per machine, and one monitor per-person. The students would often enter the studio and sit straight in front of a computer, even though on some occasions no work was going to be done on the computers. As one student said, *"it's just what we do"* (Fg,S_5).

The studio provided at least one desktop computer per student, but it was observed that students brought their own laptops and plugged them into unused monitors (or they simply logged into a nearby desktop in addition to their laptop). This created unique group dynamics, but also allowed information to be kept on secondary screens. We found that this increased the potential for peripheral information, and better supported reflection-in-action for a student's current activities.

Permanence Of Design Work

Observation Theme: Physical space

Framework Parameter: Open environment

As the studio space belongs to the students, they can leave design work up and around the space. Unlike other spaces that get reused for different courses and have to have their work taken down at the end of each session. Which can aid students' awareness of work around the space. This is only possible with an open space which is dedicated to studio work.

Some of the students' work is semi-permanent. Work is made visible on whiteboards, is occasionally updated, and persists for several weeks. Eventually the work is wiped clean to make space for more work. However, paper based work (e.g. standard A4 paper, flip-chart paper, or sticky notes) often persisted throughout the entire year.

The use of whiteboards and wall space was used for design ideas and documentation, and led to some observed instance of reflection; groups would start discussing a project feature, but would refer to displayed design work to discuss its integration or impact on their designs. This is not something that could occur in a transient setting, such as a lecture, because the work would have to be taken down for the next lecture. The persistent presence of the studio allows students' work to stay up for weeks or months at a time. This relates to our very earliest understandings of art/design studios (constant visualisation of product prototypes). It might also provide insight into why digital tools for documentation (e.g. the touch screen) were not used. Work goes away when the machines are turned off.

6.3.2 Facilitation of Studio

The way a studio is facilitated will have a significant impact on the studio and the culture within. For this reason the staff do not dictate how the space should be used and have always encouraged the students to use it as they require (e.g. not allocating desks to students). This explicit encouragement from the staff has

been very useful; staff members coming to terms with handing over responsibility of the room to the students is one thing, but the students will still share a frame of mind that restricts their activities within the space, because technically they are guests in the university owned space. This is why making the expectations explicit is important, and even reminding them during the first few weeks. Further, the students were told at the start of the course that they could be as flexible with the space as they wanted (move tables, chairs, desktop computers etc.). In practice this rarely happened, finding instead that the space should support semi-permanent placement of students, as they tended to "nest" themselves in particular places within the environment and stayed there.

Giving Students a Specification Vs. Not

Observation Theme: Defining own projects

Framework Parameter: Flexible facilitation

In modules where the students define their own projects, they are not provided with a project specification. Students build their own specification for their project which is checked over by the students' mentor – ensuring adequate coverage of the curriculum for the module. This requires a level of flexibility from the teaching staff, as they enforce a curriculum, but still support the potential variability of the scope of a project.

Managing Expectations

Observation Theme: Staff

Framework Parameter: Studio belongs to the students

During the focus group the students were asked about what they thought of the studio module before it started, and briefly into the start of their first term. Firstly, not all students knew they were doing a studio-based module, with some of the students in the focus group making no explicit decision to learn software engineering this way (the students were told about the studio module towards the end of

their first year). However, some of the students were familiar with the intention to learn in the new studio, with one of the students stating that they even changed to software engineering from the computer science degree scheme because of this studio module. However, they did not know what to expect.

During the first studio session T_1 suggested that the staff and students sign a "contract" agreeing to a set of expectations they had of each other moving forward (Obs[230:1],T_1); the contract was in fact just a piece of large flip-chart paper, and is an exercise in setting out the staff's expectations of students and students' expectations of staff. It was open for discussion and disagreement, and each point raised was written on the sheet of paper. T_1 would also put other members of staff on the spot and ask what they expected of the students, which was also written down. Once all suggestions were exhausted, everybody involved in the studio course (staff and students) were asked to sign the bottom to indicate that they understood and agreed to all of these expectations. This document was then put on the wall, where it stayed throughout the course. This was done in part because of the use of a new and different approach to learning for both the students and the staff, but also to reinforce that this course is their studio.

Multiple Staff Per Session

Observation Theme: Staff

Framework Parameter: Availability of multiple expert staff

It was observed that having multiple staff members in a studio session brought multiple perspectives to the students' work. The students frequently had the attention of a staff member, therefore maximising the potential benefit to their project and the time spent in the workshops. An alternative course with few staff members, or much larger numbers of groups or individuals would mean that students would be limited by how much interaction they have with a member of staff.

A question of whether a studio is cost effective arises when having multiple

members of staff attending individual studio sessions. In Lancaster's studio a typical session would have three members of staff present. When interviewing T_1, he suggested that it may not be as much as originally perceived, as staff prepare less material before a session, as the emphasis is on in-session coaching. Also, only one member of staff is required to do any preparation, as the others benefit the students by just attending the sessions. Furthermore, a significant portion of assessment takes place within the studio session, especially in the students' third year (Obs[SCC.330:5-10]). This is in contrast to some interview participants in chapter 4, B_9 and B_{12} in particular, who claim that studios are hard because of the high levels of staff interaction and time requirements.

The students also talked about the rich level of interaction from having numerous staff members (Fg). They discussed the importance of having the space and time to work on their projects, but with frequent high levels of rich interaction for numerous staff. They also stated how they liked having multiple perspectives from the different members of staff. Having a higher staff-to-student ratio can support better reflection during studio workshop sessions because of the various perspectives each staff member brings to the studio – take a project proposal to an architect, and she will discuss the architecture, take it to a cybersecurity researcher, and he will discuss security and the robustness. Having more staff available during workshop sessions led to plenty of opportunities for rich feedback and discussion, feeding back into the reflective process of constant questioning.

Staff Need to Be Maestros

Observation Theme: Staff

Framework Parameter: Availability of multiple expert staff

It has been discussed in an interview with T_1 that staff members who teach in a studio setting need to be fluent in a wide variety of software engineering principles. This is because project courses require a variety of experiences. Therefore, people with experience or training in a narrow area of software engineering may not be

able to offer adequate coaching to students at all times throughout their projects.

Another reason staff need to be maestros is because of the students' assessments. Some projects, or aspects of projects, can be graded with a checklist of expected outcomes. However, with large complex projects there is a lot that can occur that will not fit with a simple checklist. This is an argument for assessing projects holistically.

Timetabled vs. Non-Timetabled Interactions

Observation Theme: Student Engagement

Framework Parameter: Expected time spent in the studio

The students would routinely meet each week (Obs[SCC.230:7-20]) and work in their groups. Most of the work that the students did was performed outside of timetabled sessions. On several occasions observations of the studio were made outside of the routine set out by the timetable. Outside of the timetabled sessions, the students often helped other groups' students. It was not uncommon for the students to work late at night, or even all night and was a weekly routine for some of the students (Obs[SCC.330:5-10]).

Within the timetabled sessions they received help from teaching staff if they needed it, with some support given by their peers if it was convenient. During timetabled sessions the groups would even show some competitive humour towards each other leading up to, and sometimes during, the assessed demonstrations (Obs[SCC.330:5-10]).

Learning How To Learn

Observation Theme: Student Learning

Framework Parameter: Learning to learn

On several occasions the students had to learn something for their project that is not on the curriculum. This is a symptom of having the students define their projects; as the projects are not thoroughly prepared by experienced teaching staff

prior to the start of term, there will likely be elements of uncertainty which the students will need to resolve themselves. For example, this can be in the form of research libraries to achieve or support a specific functionality, or it may be searching through documentation for a device they intend to use. This ability to research on their own is an important professional competency.

One of the students specifically wanted to learn PHP for a specific set of components in their project. This particular language is not taught as part of their degree, so during the focus group he was asked how he learnt it – he humorously replied: *"Google"* (Fg,S_9). The use of this language was decided by the group, and was a challenge that this student wanted to tackle. As such he now has experience with learning a technology on his own. Moreover, he has experience with how to learn a new technology. This resonates with numerous discussions had during studio sessions where students referred to the ability of learning new technologies by researching them online.

Role Of Video Integration

Observation Theme: Assessment

Framework Parameter: Flexible facilitation

In a stand-up meeting S_{10} asks about submitting a video instead of a report for the following week (Obs[330:9],S_{10}). He was encouraged to do this and told that as long as the criteria that is looked for in the report is covered then it would be acceptable to do so. Links to these videos are given in chapter 5.

6.3.3 Modes of Education

Lancaster's software engineering studio is geared towards practical group-based learning, and the flexibility of educational approaches has been heavily supported by the attitudes of the staff and the layout of the studio space. Workshops are longer than normal lecture sessions, at three and a half hours, and there is usually a theme for individual sessions, but that does not mean that the theme will be the

only topic explored. The length of the sessions allows for topics to be introduced and then practised in the context of the students' projects; a sizeable chunk of teaching staff's time is spent mentoring and critiquing individual groups. Often the staff have performed impromptu teaching, taking an associated topic, or an element the students are struggling with and performing a quick demo on-the-fly. Techniques used in the workshop sessions typically included practical exercises, student presentations and demonstrations, discussions and critiques.

Hacking vs. Software Architecture

Observation Theme: Role of Software Models

Framework Parameter: Flexible learning/Switch approach based on activity

One of the benefits of a studio, flexible learning, can also be a detrimental effect to the students. During the observations there were numerous occasions where staff members told the students to stop "hacking", that is, to stop neglecting software engineering principles and processes. It was observed that when the course intensity was high (which it often was), there was an increase in instances where the students resorted to hacking development techniques, at the detriment to the software engineering principles or practices.

Clarity and insight into this was provided during some conversations with a student (S_{10}). He described a trend, which correlates with the observations made, where the intensity of the course encouraged hacking. He described situations where he felt that the product was emphasised over process because of the workload – he acknowledges that he was aware that they were hacking at the start of the studio's second year, but believed that it was infeasible to complete the weekly milestones, and they had to focus on programming and developing features set forward in their requirements documents. Despite the accusation of hacking, the students were heavily using informal models and updating documentation that was placed around the room. On reflection, S_{10} realised that they were hacking, and acknowledges how the software engineering principles would have helped, sug-

gesting that it could have also helped with coordinating the group in some cases. However, he finishes by saying that he does not know how else it could have panned out, given their workload and deadlines for other courses as well.

A badge reward system was introduced in the subsequent term (SCC.331, term two) as part of the process of continual improvement to the course. Each badge represented a software engineering practice or technique and were awarded to the students who performed those tasks during their studio projects. Students had to pick a minimum predetermined number of badges from a larger set to satisfy the course requirements. This had the added benefit of, in T_1's words, tackling the "hacking" issue as well. This technique also added some game elements to the course.

6.3.4 Flipped Classroom

Observation Theme: Theory vs. Practice

Framework Parameter: Mentoring/coaching

A flipped classroom is a particular approach to teaching where students explore course material before a session (e.g. through reading, or recorded lectures) and then attend a session to discuss or practice what they have learnt. This allows the teaching staff to better focus their attention on helping students apply their knowledge or even resolving common shared problems.

Students were expected to have completed some set reading before a studio workshop. Similarly, for some sessions the students are expected to have performed some specific activities (or reached some milestone), e.g. the groups must have started code development by a certain session so there is a code-base for a session exploring testing techniques. Feedback is then given during these sessions based on progress they make throughout the preceding weeks.

In some earlier sessions the students did not do all of the reading or activities (Obs[230:3-10]). This led to situations where the workshop sessions had to adapt to accommodate this. However, the students soon reduced this, as they saw it was

necessary to do it for each of the sessions.

6.3.5 Project Courses

Observation Theme: Theory vs. Practice

Framework Parameter: Real-world projects

Large and complex projects are expected by the studio students. Identifying solutions to large and complex problems, those that you are expected to experience as a practitioner, is not supposed to be trivial. Project scale and complexity will inherently encourage reflection because it is difficult to determine what the "right" solution is: Does the solution complete the project's goals? How well does it complete them? Reflective conversations were observed within groups as a direct result of the project's size, for example, discussing how to better manage the development of a subsequent milestone compared to a previous one. Large projects can provide an increased opportunity to experience big problems and creative problem solving.

Furthermore, during the focus group it became clear that the students were inspired to work on their projects because they came up with the project idea, and therefore had a vested interest in it (Fg,S_9). Project size and complexity provided greater scope for reflective practice, but working in a studio better emphasized reflective practice.

Mentoring/Coaching

Observation Theme: Staff

Framework Parameter: Mentoring/coaching

Aside from the coaching received during studio workshops/sessions, each student group is assigned a supervisor for the duration of a project. This is someone they meet regularly outside of studio workshop sessions, acting as an anchor for the course. They provide support and guidance towards the projects and group members, and ensure adequate coverage of software engineering principles.

Each of the assigned supervisors coaches the group. Groups meet with their supervisor at least once a week to articulate the project's problems and solutions. Supervisors also constantly questioned the students, encouraging them to reflect. So, for example, rather than suggesting alternatives to a problem or proposed solution, they would pose questions to students, encouraging them to generate their own possible or alternative solutions.

Part of a coach's repertoire is the ability to perform impromptu teaching. This is where a staff member teaches or demonstrates to a student, a group, or the entire room, at any time, based on his or her assessment of whether the students require it or not; flexibility of teaching is an important part of a studio, as set forward in the studio framework. It provides additional opportunities for the students to reflect on their understanding of a subject, but staff can also reflect on students' progress and the course itself, providing impromptu teaching where necessary, for example, with demonstrations or short discussions.

Requirements Change

Observation Theme: Staff

Framework Parameter: Real-world projects

The 'requirements change' task was intended to ensure that students experience what it is like to have a requirement change in a real project, and then dealing with it within their project confines (Obs[230:14]).

In the week preceding the task forcing the requirement change on the students' projects, T_6 openly reflected in front of all the students about how all of their projects had emergent requirements and that the students had adapted to cope with these (Obs[230:13]).

To simulate a real-world issue, the students were forced to reflect on their work in its current state and re-evaluate the project to address a new or edited requirement. Each group's advisor formulated the new requirement on a per-project basis: it was estimated to take approximately a week's worth of effort and got injected into the

brief late in the project. Among other things, this encouraged reflection-on-action as the students reflected on their earlier design decisions and asked if they could have done something differently or they changed something to make addressing the requirement easier. In some instances it helped reinforce why good software engineering practices are important.

Peer Learning

Observation Theme: Student Learning

Framework Parameter: Peer-learning

Students working in the studio learn from each other. Peer learning frequently emerges serendipitously as one student asks for help from anther student. One example that was given was the use of pair programming, with a student adding that *"two heads are better than one"* (Fg,S_9) – they had not been taught about pair programming at this point.

Near the start of the students' third year, the staff and researchers involved with the studio were discussing adding student-driven presentations. The idea was that each week a student would research a topic or report on experiences and results of using a particular software engineering technique (not just a demonstration of the work), and present and teach the rest of the group. This is yet to be implemented, but it is a technique that has seen some success in other studios (Shaw et al., 2006). Another idea, that also encourages further self-learning, is to get the students to read the 'required reading' and do a short weekly presentation.

On-Demand learning

Observation Theme: Student Learning

Framework Parameter: Impromptu teaching

Workshops are planned sessions with certain learning techniques planned or expected to occur. On-demand learning is the ability to adapt to emerging learning needs. One example observed was an optional Android workshop that was set

up in response to two of the groups intending to develop Android applications as part of their projects (Obs[230:7],T_7). This workshop was intended to provide the students with some basic guidance and best practice advice for developing their applications.

Another way of getting on-demand learning is through impromptu teaching and flexible coaching provided by a member of staff. Flexible coaching was experienced frequently within studio workshops, as there were always teaching staff moving around the studio space observing progress, critiquing and offering suggestions.

Stand-Up Meetings

Observation Theme: Agile Methods

Framework Parameter: Mentoring/coaching

These meetings occur once a week, in between workshop sessions, and act as an opportunity for a member of staff to touch base with the students. They are used by the students to very briefly present their project, describe the goals and milestones, what the team members' responsibilities are, and general progress. Essentially, they are a project management activity, but they also keep up the pace of the project. In the meetings the students would briefly describe the group's activities, and whilst reflecting and discussing their burndown chart (displayed on the touch TV on the wall) they would go around and discuss each individual's contribution that week. The discussions often referred to particular requirements that the activities were affecting.

A typical stand-up meeting consisted of the students gathering around the wall-mounted TV, which would display that group's burn-down chart for that week's sprint. Each of the students took it in turns to describe their progress and any complications they are facing. Approximately half-way through the module, the member of staff attending the meetings announced that the stand-up meetings would no longer be thirty minutes per group, and instead would be fifteen minutes (Obs[330:5],T_4). This was a (successful) attempt to keep the meetings on topic, and

not deviating too far into development issues. However, staff members would stay after the stand-up meetings to discuss project specifics if desired – the shortened meeting times were to encourage a better project management focus.

6.3.6 Awareness

The first thing that becomes obvious when walking into Lancaster's software engineering studio is all of the work around the room, on the walls and on the whiteboards. Software engineering is inherently not a very visual field, but the students have made their ideation, diagrams, and processes very visible to all in the studio. The studio is also very social, so the students have been quite familiar with how the other groups were getting on.

Mobile Whiteboards

Observation Theme: Student Engagement

Framework Parameter: Visual work

The use of whiteboards, specifically the mobile whiteboards, has proven invaluable to the students. They were often used by all of the groups, and frequently form a centrepiece for discussion. It was observed that having something physical was easier to show off than a digital equivalent, and therefore easier to engage in critique over.

The studio had enough mobile whiteboards for at least one board per group, which removed any pressure to wipe content between sessions. Whiteboards frequently led to group discussion, which is conducive to reflection-in-action. Typically, they function as a central discussion point but were also used to house peripheral information, often physically relocated in close proximity to the students. The whiteboards, mixed with the use of the work on the walls, often led to frequent reflection: students would interrupt their current tasks to look and interact with the displayed work. Interestingly, the studio's digital tools – a large wall-mounted touchscreen and horizontal touch table – were rarely used by the

students, because they are smaller and not as simple to interact with. One of the students shared his thoughts on why he thinks they work well in group projects: *"I think that when I stand up to get in front of the whiteboard to write something down, you just focus. I'm just like 'right, need to think about what I'm writing and write something coherent' so that everyone else can understand it as well. It's not just a jumble of ideas in my head anymore"* (Fg,S_9).

The mobile whiteboards also link into the physical environment category as they provide some flexibility, as they were frequently moved around: they were moved into the group's area when used, put against the wall when not used (out of the way) and moved near the group when used as a point of reference (but not physically interacted with). Although the students had permission and the ability to be flexible with their environment and furniture, it was not used as expected. Fortunately, that flexibility manifested itself in another form: their use of the mobile whiteboards. These created on-the-fly changes to the dynamic of the space, which alleviated the fact that there were no dedicated individual or private spaces. There were also no dedicated social spaces, as space is a premium, but this was balanced by the social culture that the students adopted.

Visibility of Design Work

Observation Theme: Physical space

Framework Parameter: Visual work

Working in groups leads to design work being made more public, as individuals need to externalise their thinking for others to comprehend and contribute to the design process. This will inherently lead to design work being more visible, as knowledge is needed to be communicable and transferred amongst group members. If design work is done by individuals, the process design decisions may not need to be made explicit. For others to comprehend a design whilst working in a group, good communication skills are required.

The visibility of design work lends itself to the permanence of the work in the

space. Visible work, or work that is good at communicating its ideas, will inherently be a candidate for being left permanently up in a space. Students could leave their work up, with designs and prototypes remaining on walls or whiteboards. The benefit to other people in the studio was in seeing other groups' progress and how their thought processes emerged around a particular problem. Making work visible also allows for greater possibilities of reflection-on-action for students' past work and broader reflection-in-action for their current activity, primarily because it increases the accessibility to their work and encourages feedback from everyone else in the room. Because each group works on vastly different projects, there is no culture of hiding your work for fear of plagiarism, inadvertent or otherwise. In fact, looking at each other's work is encouraged.

6.3.7 Critique

This is helped most through the approach of coaching and 'constant questioning'. It can be used in various forms of critique (formal, informal, group and individual), all of which are utilised in Lancaster's software engineering studio, and encourages the students to reflect on their project, process and even the answers they are giving to the questions. The students show off their work and get in-depth feedback every week, with the intention of developing ideas and solutions throughout the project – allowing the students to make mistakes, but crucially, learn from them. They have 'permission to fail'.

Transition From Weaker To Strong Students

Observation Theme: Group Dynamics

Framework Parameter: Peer-coaching

At the start of the studio's first year there were a range of stronger and weaker students; they were easily identified by seeing who actively engaged in the course, and who was trying to attain the baseline of achievement in the course. Over the course of the studio modules, the students have demonstrated significant growth

in this regard. All students equally shared the load of group projects, and they all demonstrate and present the work to peers and staff. During the final weeks of observations, during a grading session, the staff explicitly realised the difference of how they were referring to the cohort of students, noting that they were calling them all *"strong students"* (T_2). In the interview with T_1, he stated that the third year students were now articulate and enthusiastic. At the end of the year, the students felt that they had learnt and practised a lot, with one of the students remarking that *"we've probably learnt quite a lot more from this one module"* (Fg, S_5).

Students that appeared to have lower confidence had numerous opportunities to discuss and present their work to others. Sometimes it would be presenting their contribution to the group's project, or presenting the group's project on their behalf. Strong students on the other hand had the opportunity to practice peer-coaching and peer-learning. Their projects required a broad range of skill sets, so many students found an opportunity to coach their peers in a particular skill set that they had.

Touch Screens

Observation Theme: Interacting with Digital Media

Framework Parameter: Continuous Critique

The studio was provided with two touch screens, a wall mounted-touch TV and a horizontal touch table. The wall-mounted touch TV was rarely used, with the exception of the for the presentations – which it was well positioned for in the centre of the room, facing a semi-circle of chairs. It was also used for social situations. For example, the students were frequently observed using it to play online videos or music through it. Therefore, this also links into other framework categories such as 'playful space' within 'Inspiration'.

The touch table also saw very little use during the studio's first year, with the exception of a few touch-based games. However, the students were given an

additional requirement for their SCC.330 projects: make a touch screen friendly interface as part of their project. As would be expected, this increased the use of the touch table significantly. There were no obvious interaction issues with the device, which suggests that it was rarely used last year for another reason, perhaps because it was not needed?

During a session observing the students in their third year, staff were discussing potential changes to the studio. It was mentioned that smart whiteboards are being considered for future years – it was discussed that this was in part to do with the success and prevalence of the mobile whiteboards.

Delivery Of Product At Each Iteration

Observation Theme: Agile Methods

Framework Parameter: Developing ideas & Continuous Critique

In the third year each project iteration ("sprint") lasted a week. The students planned, developed, and tested their project each week and demonstrated their work as a group and then as individuals on a one-to-one basis. This ensured that the students tested frequently, as they did not want a broken demonstration.

Naturalness of Design Iteration

Observation Theme: Agile Methods

Framework Parameter: Developing ideas

Designs were easily iterated through, especially in the first several weeks of the year (Obs[230:1-10]). Working closely together whilst being able to share and express ideas led to numerous design iterations. As the students progressed through their project, they would refine certain aspects to better suit the requirements or software engineering principles. The design iterations were public, as they were openly discussed and often drawn on the whiteboards.

Stressfulness Of Critique

Observation Theme: Critique

Framework Parameter: Continuous critique

Students looked uncomfortable at the start of their first year. This could be because they are not used to publicly presenting and receiving critique of their work. Though the students visibly looked a lot more comfortable giving and receiving critique as they progressed through their modules.

Constant Critiquing

Observation Theme: Critique

Framework Parameter: Continuous critique

This is made possible by the number of staff available during studio sessions, and the emphasis of critiquing and reflecting on their work. Workshop sessions were designed to allow for maximum critique and feedback.

In the students' second year (SCC.230), they received a lot of critique through coaching and the frequent, presentations. This changed towards the students' third year, where the workshop sessions are quite different to those of the last module in the previous year. The weekly sessions workshop sessions are primarily used for demonstrating and articulating the students' projects, receiving direct feedback from numerous staff members. In these sessions the students have their work assessed by those staff members (both group and individual grades). The weekly feedback and grades enabled a constant flux of work, with the students incorporating the feedback into their projects every week (something that may not occur in other courses where assignments are handed in at the end of the course – feedback received for work after it has received its final grade will unlikely be incorporated into the work). Even though the workshop sessions are the primary time that the students are exposed to teaching staff on this module, they only make up a small portion of the time that the students spend in the studio overall; this resonates with what numerous interview participants said in chapter 3, that

timetabled staff contact is by far not the only element that makes a studio.

All studio module sessions had a solid routine based around critique, feedback and assessment. Numerous staff members would attend, and the session would start with one of the groups demonstrating their project, with emphasis on the previous week's development and any software engineering principles involved, followed by the other group. The demonstration involves constant questioning over the project, progress, and principals applied, as well as feedback (a 'Design Jury'). Specifically for SCC.330, once both groups had demonstrated their projects, the staff would go around the studio as a group and talk to each individual student about their contributions to the group and the project, again questioning, providing critique and feedback (a 'Desk Critique').

Although the students were uncomfortable with the amount of presenting at the very start of the studio modules, the students have since expressed that they like the high frequency of feedback (Fg).

Peer Critique

Observation Theme: Critique

Framework Parameter: Peer critique

The studio environment nurtured feedback among students by encouraging critique from peers and fostering a culture that openly supports critique. Through critiques, people can reflect on their work to either find alternatives or understand why their solution is optimal. This forms a significant part of the studio culture (Kuhn, 1998).

The peer-testing task introduced the idea and benefits of getting the students' peers to test their code. Students had to reflect on their own testing practices and find better ways of identifying testable elements of code. During the focus group it was learnt that the students found this session to be the least useful and interesting, but they also admitted that this is likely because they did not have a significant codebase at the time that this session was planned, which prohibited

the session from functioning optimally.

The task related to code review had similar benefits to peer testing, but it looked more closely at code quality and design decisions. It did not have the same reaction as with the peer-testing session because it happened later in the year. This meant that the students had developed a substantial enough code base, which provided the students with adequate code to review and critique.

In some of the presentation sessions, students and staff wrote down positive points along with elements of concern on Post-It notes, which were shared and discussed after the presentations. One benefit to this is that the post-presentation sessions did not end up in lengthy Q&A exchanges, primarily because the thoughts and questions were documented and made visible on the walls over the rest of the course. This also helped overcome a few instances of awkwardness – regarding commenting on a peer's work. On several occasions throughout the year, the students reflected back on the positive comments and concerns. It is noted, however, that this is a possible avenue to explore for scaling the studio.

Frequency of Assessment

Observation Theme: Assessment

Framework Parameter: Continuous critique

The students felt the stress of twice-monthly assessment. They did however say that they preferred this approach instead of end of year written examinations. One reason for this is because they get to work on and improve their work throughout the year. This resonates with a previous interview participant (chapter 3):

> *"Most students just get a mark and then they do another project. They read the description but they probably won't make an effort to revise, simply because they have already got a mark."* (A_8).

During the weekly workshop sessions (SCC.330), the students demoed their project as a group, and then the three members of staff would navigate around the room, talking to the individuals of the group and assess their contributions to

the group. The students are given a group grade, which is then adjusted by their individual grade.

6.3.8 Culture

The students have stated that when working with other computer science students in one of their shared modules, they would work in the computer science lab (a traditional computer lab). But when working on their software engineering work, and often when working on individual computer science work, they worked on it in the software engineering studio. The students say that this is because it is "our space" (pointing to a sense of ownership of the space) and that it was often a better work environment. The students have referred to the studio as "homely".

Students Favour Tasks They Have Experience With

Observation Theme: Defining own projects

Framework Parameter: Peer-learning

Students often appeared to default to allocating certain tasks (wider project skills outside of core software engineering concepts, e.g. graphic design or database optimisation) to students who were most comfortable with the skills to fulfil those tasks. However, on several occasions the students would share the workload with others that did not have those specific skills to achieve the task. This led to instances of peer-learning, as described earlier in this section.

The students in the groups would play to their strengths, utilising and sharing the skills they knew, but also learning some new skills from the other students (e.g. database logic, Photoshop and design etc.).

No Plagiarism: Lack of Tension

Observation Theme: Defining own projects

Framework Parameter: Sharing

As each of the groups were working on different projects, which they defined them-

selves, there was no risk of plagiarism, accidental or otherwise. This led to countless observed instances throughout the course were students helped each other out.

Breaking Down Barriers Between Staff And Students

Observation Theme: Staff

Framework Parameter: Social

When the students started the studio course they were often quiet and rarely willing to engage unless they had to (for example, when a question is asked to everyone during an activity). With the high levels of interaction between the staff and students, the staff quickly built a rapport with the students. This had the added benefit of making the staff more easily approachable for the occasions when the students were deliberately given some space to work on a problem.

The approachability of staff is also helped by having numerous staff within a single session, and prevents against situations where only a single teaching staff member is available, and that person has their time consumed by a single student.

Enthusiasm Of Staff

Observation Theme: Staff

Framework Parameter: Good work ethic

Staff are enthusiastic for the course, as well as about teaching on the course. Students have picked up on this themselves, as one student described a typical studio session whereby *"you don't have to stick your hand up [to get their attention]. They just come over for a chat. They want to see what's going on. They seem kind of as enthusiastic about it as we do"* (Fg,S_9). It was also discussed in an interview with T_1 that members of staff are putting in time, not because they have to, but because they want to.

Procrastination and Breaks

Observation Theme: Student Engagement

Framework Parameter: Good work ethic

Throughout all of the observation, students were never observed procrastinating. Students often took small (bathroom and food) breaks when they wanted them. Although looking around the room there would be evidence of distractions (e.g. social network websites open on monitors), these were often open in the background and were used infrequently and only for short periods of time (Obs[230:5-20]). However, sometimes these social networking websites were used as collaborative tools, to transfer files and communicate with team members that may not be present.

Although they were not always interacting with these non-work related sites, they only infrequently turned to them in short bursts. This is summarised by one of the students: *"Usually you just got Facebook open, someone messages me, talk to them for a little bit, [then] go back to work"* (Fg,S_9). Whilst this is not strictly desired if it distracts from an activity, they can also be seen as brief breaks.

Learning From Failure

Observation Theme: Student Learning

Framework Parameter: Permission to fail

The year-long projects that the students embarked upon encouraged them to learn more than just the scope of the curriculum. Because students define their projects, they are forced to further reflect on their practice. One example from Lancaster's software engineering studio was a group of students who needed to set up a server for a website. They found issues with requesting server access and setting up the virtual machines; subsequently, they missed some deliverables. This taught them to identify and organize resources early, as well as to have mitigation plans in place. It was a valuable learning experience, and despite this problem they did well at the end of the year.

Interaction Across Groups

Observation Theme: Group Dynamics

Framework Parameter: Peer-learning & sharing

Once the students advanced to their third year (SCC.330), they were now sharing the studio with new second year students (SCC.230) – these students were not explicitly being observed. Despite the students working in groups, there were some instances of interaction and communication with others outside of their year group. However, the observations of cross-group interaction pertains to students within the same year group.

Students would rarely interact with other students outside of their group, with the exception of social interactions. Some of the students have helped other groups with bug fixes, and on one occasion assisted with a database schema design. However, students frequently worked together and helped each other within their groups.

Limited Lurkers

Observation Theme: Group Dynamics

Framework Parameter: Good work ethic

A lurker is someone who does not participate much in group work, relying on their team members as to perform the work. Although there were some instances of students showing tendencies towards lurking, they do not have much chance to hide in the studio. An example of this is the frequent presentations and demonstrations, and also the use of constant questioning. To avoid one student in a group answering most of the questions, teaching staff ask many questions directed towards each of the individuals of a group, as well as some questions directed to the entire group. Each student is expected to be able to coherently answer questions and show a broad understanding of all aspects of their project.

With the students working in groups, discussions of group dynamics during the focus group were inevitable; as with any group-based work, there will always be

the potential for good and bad group dynamics. During the focus group, one student described tension within his group due to reduced work ethic of other members (Fg,S_1). He talked about how he feels that this is because the studio has less structure than the computer science degree counterpart – it could be that, at this time, some students were placing studio work as a lower priority, or putting it off to the last minute. This student also went on to say that he knows others in his group *"really like work to be structured"*, being told what to do and when, and that *"it's too easy for some people to ride along, and not do much at all really"* (Fg,S_1). One troubling comment made during this conversation was that issues with other students not working well in the team may not get reported to their supervisor due to friendships within the group.

Competition Between Project Groups

Observation Theme: Group Dynamics

Framework Parameter: Social

Throughout the course, the students exhibited a friendly competitive nature towards other groups. This was particularly prominent in the students' third year. In SCC.330 they were working towards the same project brief. There was an obvious air of competition amongst the two groups, although it was never observed as a secret-keeping competitive nature; the students helped each other, regardless of group, if the opportunity arose. The competitive nature made itself most apparent during group demonstrations. One example of this was seen in the groups' competitive, but friendly, banter in pre and post-demonstration sessions (sometimes even during). Another example that was not an isolated incident was when one group was sending commands to the other group's system, whilst it was being demonstrated, using the publicly accessible API they developed. These interactions were not malicious, but simply showing the atmosphere of the group.

Sense Of Home

Observation Theme: Physical Space

Framework Parameter: Treated like a second home

The students were encouraged by the teaching staff in the first workshop to use the space like a second home (Obs[SCC.230:1],T_1). This was reinforced with physical elements, such as including a fridge; this encouraged an often contentious subject of being allowed to eat and drink in the studio. Whilst the students were explicitly told they were allowed to eat and drink in the studio, the same does not apply for other computer labs at the university.

The students admitted that they felt a sense of ownership over the space, with one student remarking: *"I almost feel like it's our little space"* (Fg,S_5). They admitted that they have frequently used the space and have even ordered pizza to the studio. At the same time, they added that the responsibilities in the studio worked both ways: *"equally, because there's that freedom, we respected it. We took all the [pizza] boxes, not even in the bins there, took them out to the skip because it is our lab innit? We don't want it to be scruffy"* (Fg,S_{10}).

The students have also discussed that they used the space to watch films, making it a social space too.

6.3.9 Individual's characteristics

There are no dedicated private and quiet spaces, although this did not present itself as an issue. The lack of these spaces is likely mitigated by the fact that the studio is large and is geared towards group work, and also that no sensitive projects or data are handled. There have also been a few times when a group segregated themselves from the rest of the room using the mobile whiteboards.

Sleeping In The Studio

Observation Theme: Physical space

Framework Parameter: Private and quiet spaces

The students would frequently organise late night sessions in the studio to work on their projects. Sometimes these would extend into all-night sessions, and to catch some rest the students would sleep in the studio. Some would go home, but others preferred the convenience of sleeping in the studio. The students would sleep on three or more chairs pulled together, or under the desks.

Sleeping in the studio is something that interview participants in chapter 3 said happened frequently, alluding to the fact that this is a symptom of highly collaborative collocated learning. Sleeping in the studio is neither a good or bad aspect of studio life; even though Lancaster's studio does not have dedicated private or quiet space, the observation is that quiet spaces were socially negotiated.

6.3.10 Inspiration

Outside of the workshop sessions, music is often played through the wall-mounted TV, but on occasion a student would show something that they found personally interesting. The music is not disruptive because everyone consents to it. During the focus group it became very clear that the students were inspired to work on their projects because they came up with the project idea, and therefore had a vested interest in it.

Although already mentioned above under Culture, the observation about 'competition between project groups' (Group Dynamics theme) also refers to Inspiration; friendly competition between students is enhanced by being in close proximity to them, and manifests itself in playful situations (for example, sending commands to the other teams public API during a demonstration).

6.3.11 Collaboration

With most work in Lancaster's software engineering studio being group-work oriented, supporting collaboration is very important. The tables laid out in 'U' shapes, seen in figure 5.8, as well as the centre of the room, provide a starting point for a variety of collaborative activities. The single most used element

for collaboration in the studio was the mobile whiteboards. Primarily a piece of equipment to support collaboration, but due to the fact that they are easily manoeuvrable, they also provided the ability to create impromptu collaborative spaces – they allow the students to change the dynamic of the room, as they see fit. These whiteboards, as well as two other wall-mounted whiteboards, saw a significant amount of use, and the students often mentioned how indispensable they were to them. They were used for many things, including informal notation as well as structured diagrams, both are great for externalising information and supporting face-to-face communication (Cherubini et al., 2007).

Learning How To Communicate

Observation Theme: Student Learning

Framework Parameter: Soft skills

The students are constantly presenting their work to the staff and other students. This enables the students to practice communicating their ideas and implementation details. This is also enhanced by a session explicitly for teaching presentation and speech skills (Obs[230:19],T_1). In this session they learnt how to effectively communicate, considering things like emphasis in speech, tone of voice, and communicating within time constraints. This is beneficial to improving confidence and for effectively conveying ideas.

Learning how to communicate can also benefit the students learning through their communication. Working in co-located groups allows the students to constantly communicate their ideas to each other. Critiquing each others' work also improves communication skills, because if an idea is not effectively communicated then the student will receive questions asking them to better explain the idea before any critique or feedback is given. Which leads to another benefit of students presenting their work, as it enables staff to more easily see if the student understands their work.

Importance Of A Group Leader

Observation Theme: Group Dynamics

Framework Parameter: Soft skills

The focus group's participants were the three group leaders (and the student representative for the module). One student expressed his displeasure of having to drag his group members along when the project started, as they were not enthusiastic about the project (Fg,S_1).

Each group had a nominated leader (which the students nominated themselves), but it was discussed that the team leads were not strictly leaders (Fg,S_5). Two of the groups described a flat leadership structure, whereby any actions taken as the leader were more akin to a moderator. This was in contrast to the other group, whereby many ideas and responsibility fell on S_1.

In the students' third year (whilst using SCRUM) a different SCRUM master was assigned each week. This gave all of the students the opportunity to experience leadership responsibilities.

Team Working

Observation Theme: Group Dynamics

Framework Parameter: Soft skills

Students learn to work as a team. Sharing workloads or goals (e.g. pair programming), and separating workloads across the team members (e.g. allocating tasks to individual members).

The group that self-described as having "good" group dynamics during the focus group (S_9 and S_{10}) said that *"most work gets done in the studio"* (Fg, S_{10}), whereas it is the opposite for the other two groups (S_1 and S_5) which self-described as having not as good group dynamics.

To mix up the team dynamics the students were placed into different groups during their third year modules. For SCC.330 the students were split into two groups (two teams of six), instead of three groups like the previous module (SCC.230).

Mixing up the groups like this gives all of the students the opportunity to further practice their soft skills, shakes up the group roles that have solidified over the past year.

The studio has experienced a very high attendance rate, with only a few students unable to attend studio sessions on rare occasions. On one of those occasions, the absent student was remotely working with the group, communicating via Facebook.

Collaborative Document Writing

Observation Theme: Group Dynamics

Framework Parameter: Soft skills & Equal effort

There were limited observations of students writing their reports during studio workshops, as the workshop sessions were generally reserved for practical work. This way the students were able to maximise the benefit of the teaching staff availability. Therefore, insight into some of the students' document writing was obtained during the focus group.

Some students opted to work on their group report all at the same time, so they could have real-time discussions: *"we're doing our group report, it was on a Google docs, and all four of us were on the same document"* (Fg,S_{10}). Other students described how the separated the report into chunks and delegated the work, in a process that resonated with how the students develop their software: *"when we get to the final report and we just need to do so much in a short space of time, we're like 'right, we need to split this up'. But otherwise it is like working together on a piece of code. Like me and (S_{11}) often sit and work together on the same function or something, just to see if we can get it working, because two heads are better than one [...] Either that, or two monitors both looking at the same file"* (Fg,S_9). Whilst talking about using Google Docs S_9 was asked if they often do report writing synchronously or asynchronously. He replied saying that *"it's all at the same time"* (Fg,S_9).

The students were asked about where they are when they are report writing. Two

groups performed most of their report writing at home, but the group that stated that they work on it at the same time added that they work *"in the studio because it's just a good meeting place for everyone"* (Fg,S_9). This was later reinforced by another student, adding that *"most work gets done in the studio, 100% to 80% of probably the whole group projects are done in the studio"* (Fg,S_{10}). Even when using tools that were made to enable remote collaborative working easier, this group chose to work collocated in the studio.

Another perspective was given by S_5 on his group: *"Yeah, ours is a bit different. I mean, we kinda got this division already because of the holidays, with the first increment [of the work] and all that. Yeah, we all decided what bits of the first increment we wanna do, and just did it ourselves, and when we got back together, after the holidays, um, yeah, we just tried to put it all together. Yeah. And the same with the group report. We just had a- we took about two days, or so, to do the whole report, and yeah the bullet points, we just split up to answer whatever parts we all did, and then two people literally just stayed together and put the whole thing together, the whole group report."* (Fg,S_5). S_5felt that the report writing was easier because the separation of report writing amongst the team members followed the division of labour of the code. S_1 described his group following a similar route to the S_5, having individual members assigned tasks to do in their own time. This approach was seen to change throughout the rest of the degree.

Using Social Networks as a File Sharing Tool

Observation Theme: Group Dynamics

Framework Parameter: Supporting equipment

Some of the student groups used unconventional techniques to support their group work. One of these was the use of Facebook as a file sharing tool. Although it is not billed as a software development collaboration tool, the convenience of using Facebook in this way outweighed other techniques – this is because the team members were already using this particular social network, and were connected with

their team members. Whilst this could indicate a lack of supporting equipment for collaborative work (the ability to quickly and easily transfer files), it also could suggest it is simply more convenient to use existing software (i.e. Facebook).

Continuous Integration/Deployment

Observation Theme: Agile methods

Framework Parameter: Soft skills

It was observed that this encouraged a high amount of communication amongst the students, benefiting from high levels of collaboration. Students would often merge their code into a master branch, which initially was done manually with no tool support.

Pair Programming

Observation Theme: Agile methods

Framework Parameter: Soft skills

Some agile processes, such as pair programming, occurred on their own without prompting or teaching the students about it. It was just a natural approach to working together in some situations.

Burn-Down Charts

Observation Theme: Agile methods

Framework Parameter: Soft skills

The burn-down charts were good at getting the students to translate project features into effort and person requirements (Obs[330:5]). Each student within a stand-up meeting was able to describe the contents of the burn-down charts and inform T_4 who was working on what that week. In this regard they are potentially useful for project management reflection. However, one student felt like it was just a formality of the course (a sentiment shared with a few others), and that it had little benefit to their project or team members (Obs[330:5],S_{11}).

Group Assessment

Observation Theme: Assessment

Framework Parameter: Equal effort

The pressure of group assessments affected the students differently, compared to individual assessments. With looming deadlines, if a feature was not working or polished enough for an upcoming demonstration, then other team members banded together to change their workload priorities (Obs[330:8]).

During a demonstration or presentation, each student describes and discusses some part of their project, often spread evenly amongst the group. The students support each other during these presentations, whereby if someone forgets to mention something, or if something opportunistically presents itself, other students will mention them – it will benefit the student who forgot to mention that aspect, and also the group as a whole.

6.4 Conclusions

This chapter shares insight into Lancaster University's studio implementation, shares observations from an extensive ethnographic study, and reflects and correlates these observations with the interviews presented earlier in this thesis – this is done by reflecting on the observations with reference to the 'studio framework', presented in chapter 3 and 4.

The analysis of the field notes from the participant observation suggest that the studio framework can be transferred to software engineering education. Lancaster's software engineering studio specifically reflects well in the studio framework, with all of the framework's categories represented by observations in the field notes, and supported by the students in the focus group.

The studio was initially only intended for smaller groups of students, which resonates with the studio framework, although ideas for future work include scaling up the course. The small group size has enabled numerous rich interactions

between the staff and students, but the staff have not necessarily put more time into a studio than a lecture-based counter-part. Whilst there are no concrete figures to support this, the staff do less preparation than lectures – studios focus more on on-the-fly coaching. This is also supported by conversations witnessed between studio and traditional teaching staff whilst observations were on-going.

Reflective practice is considered to be the essence of the studio concept (Schön, 1983; Tomayko, 1991), and is described in detail in section 2.2.2. Although reflective practice is not explicitly mentioned in the studio framework, it is implicitly covered by several of its categories; for example, 'Modes of education', 'Awareness', and 'Critique', all of which have reflective aspects to them. This resonates with the findings of this chapter, whereby many of the themes that surfaced contained overlapping elements of reflective practice. Several of the examples observed illustrate a variety of reflection-in-action and reflection-on-action practices.

The studio was made possible by being able to get hold of a crucial resource: a sizeable dedicated room. Space is an expensive commodity in universities, so it may not be feasible for all institutions. Further investigation is needed to explore the implications of limited resources on studios.

Based on the observations and student feedback, one significant aspect of the studio was the use of the mobile whiteboards. A seemingly simple asset, these provided a wide range of benefits, and were always being used: supporting collaboration, changing physical layout of the room through positioning and allowing greater awareness to other students, to name a few. Another prominent aspect was the approach of constant questioning (Tomayko, 1991), which is inherent in Lancaster's studio approach. It helps build a culture supportive of critique and encourages further reflective practice in the students.

A good metric to end these conclusions of Lancaster's studio is students' enjoyment. At the end of the students' second year, several of them indicated that the studio module is their favourite of all modules in their degree – not bad for the first year of any course. This opinion persisted into their third year studio modules,

with opinions openly shared about their preference to the studio modules over the rest of the degree. Beyond the students, the staff are also enjoying the studio, putting their time into the studio because they want to.

7 Conclusions

Studios are a very beneficial educational approach, replacing or sometimes complimenting traditional teaching methods. This has led to software studios increasing in popularity, although not many have been implemented (Hundhausen et al., 2008a). This thesis provides an opportunity to others to implement their own studio using the studio framework (Table 3.2) and the software studio framework (Table 4.3) as a foundation, and reflect on shared experiences of Lancaster University's software engineering studio.

7.1 Contributions

The purpose of this thesis is to ask: what are studios, what is the current state of studio education in software engineering, and can studio education be transferred to software engineering? These are answered by the creation of the studio framework through interviews with studio educators, exploring literature and interviewing software engineering studio educators, and analysing observations of Lancaster University's software engineering studio.

7.1.1 Studio Framework

A priority for this thesis was to provide a definition of studio education, presented in chapter 3, and later extended variant with additional parameters in chapter 4; The categories mostly stayed the same (though 'Inspiration' was removed), and the software studio framework shows some variability in the parameters discussed

between the two studies. This needed to be created due to the lack of a concrete and shared definition for studio education. It was ultimately decided that several studio-based educators and researchers should be interviewed, from various disciplines and institutions which studio education originated from. This allowed for the collection of a variety of perspectives on studios. The studio framework was subsequently created through the analysis of the interview transcripts. This framework sets forward a series of "categories" which broadly describe valuable aspects of studio education (e.g. 'Awareness' and 'Critique'), these categories contain "parameters" which specifically describe how to approach the implementation of it's housing categories (e.g. having a student display the "Visual history of progress" of a project). The framework was later extended (Table 4.3) in chapter 4 with additional parameters gleamed from software engineering studio educators, and shows how studios in software disciplines compare to the characterisation of what a studio is (chapter 3).

The 'studio framework' shares a definition and understanding of studio practices in place of a single over-simplified definition. An important take-away message for the studio framework is that, due to their complex nature, studios should not be considered a binary state (i.e. whether it is or is not a studio); there will be some spaces which are more 'studio-like' than others. Indeed, if one thing was certain from the interviews, it is that a studio may exist even when all of the categories are not satisfied; these spaces are then thought of as less studio-like than others.

The benefit to having the studio framework is that studios can now be validated, and are now more easily comparable. Although there were varying perspectives on studio education, a dominant theme surfaced throughout the majority of the interviews: there are many intertwined aspects that define studio education, but it is primarily the people and the culture that make a studio.

7.1.2 Current State of Studios in Software Engineering

The state of studio education is initially explored through the literature review, in chapter 2. Publications that describe their institutions implementation of software studios are summarised, which led to the conclusion that none of them follow a shared definition or understanding of studio education. However, there were a few common studio elements that many stated that they utilised (such as critiques).

To compliment these findings, a second batch of interviews were conducted – this time with software engineering educators with experience of studios. These interviews, and their analysis, followed a similar methodology as the previous set of interviews in chapter 3 that culminated in a variant of the studio framework: the software studio framework (Table 4.3). This framework contained most of the categories of the original framework (except for one, 'Inspiration'), with the differences being the the parameters that reside within each of the categories. This particular investigation concluded that software engineering educators' broad perspective of studio education largely conformed to that of the design discipline educators. However, individual software studio educators only discussed parts of the studio framework, so how well each of the software studios conforms to the studio framework overall is yet to be seen.

7.1.3 Analysis of Lancaster University's SE Studio

The final contribution is the analysis of Lancaster's software engineering studio. After an extensive ethnographic study, the field notes from the participant observation were analysed for themes. These themes, complemented by observations and a focus group, were then correlated with the studio framework to determine if the studio education could indeed be transferred to software engineering. The results of chapter 6 inform us that Lancaster's studio is a studio, as all of the studio framework's categories are significantly represented by observations and concrete examples.

Through these observations, insights into Lancaster University's studio imple-

mentation are shared, in chapters 5 and 6, with the intention of providing software engineering educators a grounded example of a studio.

The studio experienced numerous successes and some issues in its first year. One particularly interesting takeaway message is that the introduction of the studio at Lancaster is a new approach to teaching for the students, but also for the teachers. In the interview with Prof. Jon Whittle he stated that not all of the staff got it straight away; in the early weeks, some staff defaulted to lecturing during the workshop sessions. He clarified that this was not a criticism of the staff, but an ongoing process of learning for the teachers, as well as the students. This could be overcome by *"training the trainer"* – an approach adopted by at least one other institution teaching through studios: During a discussion with participant B_{12}, he stated that people are specifically trained to be able to coach students in their studio.

Software engineering is a creative process that is fundamentally about designing solutions to problems. Creative design inherently requires reflection (Schön, 1983), yet we know that *"naïve designers spend little to no time reflecting on how they are designing"* (Siegel and Stolterman, 2008, p.378/5). Design is also an inherently complex activity – *"design professionals [...] deal often with uncertainty, uniqueness, and conflict"* (Schön, 1987, p.157) – so bite-sized or limited scope activities should not make sense in courses with a focus on software design. The ability to design is learnable, but some claim that it cannot be taught via traditional classroom methods (Schön, 1987). In the design disciplines, it is a commonly held belief that designers *"learn how to design largely by doing it [...]. It seems almost impossible to learn design without actually doing it"* (Lawson, 2012, p.7). This gives credence to why it is important to practice and reflect in software engineering, something that was frequently observed in Lancaster's studio. The observations provided interesting perspectives on reflective practices. Sometimes reflection was planned or expected as part of an activity, but frequently it was due to the nature of the studio and its activities. Not all instances of reflection need

to occur within a studio, but the nature of studios and the culture they encourage naturally allow reflection to occur (Schön, 1987).

Something to consider is that a studio may have struggles with group dynamics, like any group-based course might. It was discussed in the focus group that a few students experienced such issues, but these diminished as the course progressed.

7.1.4 Impact

Research described in this thesis that is already published, presented in publications found on page 'ii', is already proving to have an impact in the wider software engineering academic community. One example, towards the end of writing this thesis, Prof. Jon Whittle and myself were contacted by Julia Prior about visiting Lancaster's studio. She and her colleagues have recently implemented a studio at the University of Technology in Sydney, Australia (Prior et al., 2014). In her email she stated that they implemented it after over two years of R&D which *"included drawing on your work"*.

7.2 Future Work

With the increasing popularity of software studio education there is plenty of opportunities to take this research area further. One avenue would be to perform comprehensive evaluations of other software studios against the studio framework. This would be interesting to see how current studios compare to the framework, but also could potentially uncover trends such as whether other software studios tend favour certain elements of the studio framework.

Another course of further research would be to revisit Lancaster University's software studio in five to ten years, ensuring that any growing pains that a new course may experience have smoothed out, and performing new observations. This could be useful to determine how the studio has changed over the years, and also to see how the course may have adapted to cope with the studio. This could be

enhanced by making observations of several year groups as they progress through the studio course.

A further potential avenue of research would be to look into how to efficiently transform a traditional course into a Studio-based course. In regards to this course of research, there are a couple of questions that come to mind: how seamlessly could a curriculum transfer across to a studio course? Will staff need training to ease this transition? What might they need training for? How will the curriculum need changing to accommodate a studio approach?

Finally, there are still some open questions in regards to the use of studio education. Specifically whether the benefits that studio education affords outweigh their downsides. Some of the perceived issues with studios that have surfaced regard a studios ability to scale for larger numbers of students, and the effort and resource requirements necessary to implement and run a studio course (e.g. teaching staff hours). Are studios cost effective? Do Studios require significantly more effort and resources than traditional lecture-based teaching? Inversely, a question could be: Does the benefit to student's collaborative and communicative skills outweigh benefits afforded by traditional teaching? And lastly, do studios that find techniques to scale for larger intakes of students retain their benefits of being a studio, if so, how well?

Bibliography

Baker, Alex and André van der Hoek (2009). 'An experience report on the design and delivery of two new software design courses'. In: *Proceedings of the 40th ACM technical symposium on Computer science education (SIGCSE '09)*. New York, NY, USA: ACM, pp. 519–523. ISBN: 9781605581835. DOI: 10.1145/1508865.1509045.

Billingsley, William and Jim Steel (2013). 'A comparison of two iterations of a software studio course based on continuous integration'. In: *Proceedings of the 18th ACM conference on Innovation and technology in computer science education (ITiCSE '13)*. New York, New York, USA: ACM Press, pp. 213–218. ISBN: 9781450320788. DOI: 10.1145/2462476.2465592.

Bligh, Donald A. (1998). *What's the use of lectures?* 5th. Intellect Books. ISBN: 9781871516791. DOI: 10.1080/03098268508708932.

Blumenfeld, Phyllis et al. (1991). 'Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning'. In: *Educational Psychologist* 26.3-4, pp. 369–398. ISSN: 0046-1520. DOI: 10.1080/00461520.1991.9653139.

Broadfoot, Ouita and Rick Bennett (2003). 'Design Studios: Online? Comparing traditional face-to-face design studio education with modern Internet-based design studios'. In: *Paper presented at the Apple University Consortium*, pp. 1–13.

Brockbank, Anne and Ian McGill (2007). *Facilitating Reflective Learning In Higher Education*. 2nd ed. McGraw-Hill International. ISBN: 9780335220915.

Brooks, Frederick Phillips (1995). *The Mythical Man-Month: Essays on Software Engineering*. 2nd ed. Boston, MA, USA: Addison-Wesley. ISBN: 9780201835953.

Buchanan, R. (1992). 'Wicked problems in design thinking'. In: *Design Issues* 8.2, pp. 5–21. ISSN: 0747-9360.

Bull, Christopher N. and Jon Whittle (2014a). 'Observations of a software engineering studio: Reflecting with the studio framework'. In: *Proceedings of the 27th Conference on Software Engineering Education and Training (CSEE&T'14)*. Klagenfurt, Austria: IEEE, pp. 74–83. DOI: 10.1109/CSEET.2014.6816784.

Bull, Christopher N. and Jon Whittle (2014b). 'Supporting Reflective Practice in Software Engineering Education through a Studio-Based Approach'. In: *IEEE Software* 31.4, pp. 44–50. DOI: 10.1109/MS.2014.52.

Bull, Christopher N., Jon Whittle and Leon Cruickshank (2013). 'Studios in Software Engineering Education: Towards an Evaluable Model'. In: *Proceedings of the 35th International Conference on Software Engineering (ICSE '13)*. San Francisco, CA, USA: IEEE Press, pp. 1063–1072.

Carbone, Angela and Judy Sheard (2002). 'A studio-based teaching and learning model in IT: what do first year students think?' In: *ACM SIGCSE Bulletin* 34.3, pp. 213–217. DOI: 10.1145/637610.544485.

Carlhian, Jean Paul (1979). 'The Ecole des Beaux-Arts: Modes and Manners'. In: *JAE* 33.2, pp. 7–17.

Carnegie Mellon University (2013). *'MSE Courses' [online]*. URL: http://mse.isri.cmu.edu/software-engineering/web3-programs/MSE/Courses.html (visited on 18/06/2013).

Carpenter, Shana K et al. (2013). 'Appearances can be deceiving: instructor fluency increases perceptions of learning without increasing actual learning.' In: *Psychonomic Bulletin and Review*, (in press). ISSN: 1531-5320. DOI: 10.3758/s13423-013-0442-z.

Carter, Adam S. and Christopher D. Hundhausen (2011). 'A review of studio-based learning in computer science'. In: *Journal of Computing Sciences in Colleges* 27.1, pp. 105–111.

Cherubini, Mauro et al. (2007). 'Let's go to the whiteboard: how and why software developers use drawings'. In: *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07)*. New York, NY, USA: ACM, pp. 557–566. DOI: 10.1145/1240624.1240714.

Corbin, Juliet and Anslem Strauss (2008). *Basics of Qualitative Research*. 3rd. SAGE Publications, Inc. ISBN: 9781412906449.

Cruickshank, Leon (2006). 'A Provocative Educational Application of a New Approach to Design Methodology'. In: *International Conference 'Wonderground' Design Research Society*. Lisbon, Porgtual.

Dannels, Deanna P. and Kelly Norris Martin (2008). 'Critiquing Critiques: A Genre Analysis of Feedback Across Novice to Expert Design Studios'. In: *Journal of Business and Technical Communication* 22.2, pp. 135–159. ISSN: 1050-6519. DOI: 10.1177/1050651907311923.

Dearstyne, Howard (1962). 'The Bauhaus Revisited'. In: *Journal of Architectural Education* 17.1, pp. 13–16.

DeWalt, K. M., B. R. DeWalt and C. B. Wayland (1998). 'Participant observation'. In: *Handbook of methods in cultural anthropology*. Ed. by H. R. Bernard. Walnut Creek, CA: AltaMira Press, pp. 259–299. ISBN: 978-0761991519.

DeWalt, Kathleen M. and Billie R. DeWalt (2011). 'Doing Participant Observation'. In: *Participant Observation: A Guide for Fieldworkers*. 2nd. Rowman Altamira. Chap. 3, pp. 41–66. ISBN: 0759119279.

DeWalt, Kathleen Musante and Billie R. DeWalt (2002). 'Observation and Participation'. In: *Participant Observation: A Guide for Fieldworkers*. Rowman Altamira, pp. 20–23. ISBN: 9780759100459.

Docherty, Michael et al. (2001). 'An innovative design and studio-based CS degree'. In: *ACM SIGCSE Bulletin* 33.1, pp. 233–237. DOI: 10.1145/366413.364591.

Droste, Magdalena (2002). *Bauhaus, 1919-1933*. Köln, Germany: Taschen America Llc. ISBN: 9783822821053.

Dubinsky, Yael and Orit Hazzan (2005). 'The role of a project-based capstone course'. In: *Proceedings. 27th International Conference on Software Engineering (ICSE'05)*. IEEE, pp. 645–646. ISBN: 1-59593-963-2. DOI: 10.1109/ICSE.2005.1553628.

Ehn, Pelle (1998). 'Manifesto for a Digital Bauhaus'. In: *Digital creativity* 9.4, pp. 207–217.

Estey, Anthony et al. (2010). 'Investigating studio-based learning in a course on game design'. In: *Proceedings of the Fifth International Conference on the Foundations of Digital Games (FDG '10)*. Monterey, CA, USA: ACM, pp. 64–71. ISBN: 9781605589374. DOI: 10.1145/1822348.1822357.

Falchikov, Nancy (2001). *Learning together: Peer tutoring in higher education*. 2nd ed. Psychology Press. ISBN: 9780415182614.

Fällman, Daniel (2007). 'Supporting studio culture in design research'. In: *Proceedings of International Association of Societies of Design Research*, pp. 1–12.

Fischer, Gerhard et al. (1998). 'Embedding Critics in Design Environments'. In: *Readings in Intelligent User Interfaces*. Ed. by M. Maybury and W. Wahlster. San Francisco: Morgan Kaufman Publishers, pp. 537–561.

Gannod, Gerald C., Janet E. Burge and Michael T. Helmick (2008). 'Using the inverted classroom to teach software engineering'. In: *Proceedings of the 30th International Conference on Software Engineering (ICSE '08)*. New York, NY, USA: ACM, pp. 777–786. ISBN: 9781605580791. DOI: 10.1145/1368088.1368198.

Garlan, David, James E. Tomayko and David Gluch (1997). 'Agents of Change: Educating Future Leaders in Software Engineering'. In: *IEEE Computer* 30.11, pp. 59–65. DOI: 10.1109/2.634865.

Gibbs, Graham (1981). 'Twenty terrible reasons for lecturing'. In: *SCED Occasional Paper No. 8*. Birmingham, UK.

Glaser, Barney G. and Anselm L. Strauss (1967). *The discovery of grounded theory: Strategies for qualitative research*. Chicago: Aldine. ISBN: 0202302601.

Glass, Robert L. and Tom DeMarco (2006). *Software Creativity 2.0*. 2nd ed. developer.* Books. ISBN: 0977213315.

Greenberg, Saul (2009). 'Embedding a design studio course in a conventional computer science program'. In: *Creativity and HCI: From Experience to Design in Education* 289, pp. 23–41. DOI: 10.1007/978-0-387-89022-7_3.

Hazzan, Orit (2002). 'The reflective practitioner perspective in software engineering education'. In: *Journal of Systems and Software* 63.3, pp. 161–171. ISSN: 01641212. DOI: 10.1016/S0164-1212(02)00012-2.

Hendrix, Dean et al. (2010). 'Implementing studio-based learning in CS2'. In: *Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE '10)*. Milwaukee, WI, USA: ACM, pp. 505–509. ISBN: 9781605588858. DOI: 10.1145/1734263.1734433.

Herrick, Dan R. (2011). 'This isn't your father's computer lab: computer labs redefined'. In: *Proceeding of the 39th ACM annual conference on SIGUCCS (SIGUCCS '11)*. San Diego, CA, USA: ACM, pp. 65–72. ISBN: 9781450310239. DOI: 10.1145/2070364.2070382.

Herrick, Dan R. (2012). 'Food and drink in computer labs: why not?' In: *Proceedings of the ACM SIGUCCS 40th annual conference on Special interest group on university and college computing services (SIGUCCS '12)*. April. Memphis, TN, USA: ACM, pp. 161–164. ISBN: 9781450314947. DOI: 10.1145/2382456.2382495.

Hokanson, Brad (2012). 'The Design Critique as a Model for Distributed Learning'. In: *The Next Generation of Distance Education*. Ed. by Leslie Moller and Jason B. Huett. Springer US. Chap. The Design, pp. 71–83. ISBN: 978-1-4614-1784-2. DOI: 10.1007/978-1-4614-1785-9_5.

Howell, Joseph T. (1972). *Hard Living on Clay Street: Portraits of Blue Collar Families*. Prospect Heights, Illinois: Waveland Press, pp. 392–403. ISBN: 0881335266.

Hundhausen, C. D., D. Fairbrother and M. Petre (2012). 'An Empirical Study of the "Prototype Walkthrough": A Studio-Based Activity for HCI Education'. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 19.4, 26:1–26:36. ISSN: 10730516. DOI: 10.1145/2395131.2395133.

Hundhausen, C.D. and J.L. Brown (2005). 'Personalizing and discussing algorithms within CS1 studio experiences: an observational study'. In: *Proceedings of the first international workshop on Computing education research*. New York, NY, USA: ACM, pp. 45–56. ISBN: 1595930434. DOI: 10.1145/1089786.1089791.

Hundhausen, Christopher (2011). *The Online Studio-Based Learning Environment: Studioize me*. URL: https://www.youtube.com/watch?v=7EFCx56zRCQ (visited on 12/06/2014).

Hundhausen, Christopher et al. (2009). 'Integrating Pedagogical Code Reviews into a CS 1 Course : An Empirical Study'. In: *Proceedings of the 40th ACM technical symposium on Computer science education (SIGCSE'09)*. New York, NY, USA: ACM, pp. 291–295. ISBN: 9781605581835. DOI: 10.1145/1508865.1508972.

Hundhausen, Christopher et al. (2010). 'Does studio-based instruction work in CS 1? An empirical comparison with a traditional approach'. In: *Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE '10)*. Milwaukee, WI, USA: ACM, pp. 500–504. ISBN: 9781605588858. DOI: 10.1145/1734263.1734432.

Hundhausen, Christopher D., N. Hari Narayanan and Martha E. Crosby (2008b). 'Exploring studio-based instructional models for computing education'. In: *Proceedings of the 39th SIGCSE technical symposium on Computer science education (SIGCSE '08)*. New York, NY, USA: ACM, pp. 392–396. ISBN: 9781595937995. DOI: 10.1145/1352135.1352271.

Hundhausen, Christopher D., N Hari Narayanan and Martha E. Crosby (2008a). 'Exploring studio-based instructional models for computing education'. In: *ACM SIGCSE Bulletin (SIGCSE '08)* 40.1, pp. 392–396. DOI: 10.1145/1352135.1352271.

Hundhausen, Christopher D., Dana Fairbrother and Marian Petre (2011). 'The "Prototype Walkthrough": A Studio-Based Learning Activity for Human-Computer Interaction Courses'. In: *Proceedings of the seventh international workshop on Computing education research (ICER '11)*. New York, NY, USA: ACM, pp. 117–124. ISBN: 9781450308298. DOI: 10.1145/2016911.2016935.

Hundhausen, Christopher D, Anukrati Agrawal and Pawan Agarwal (2013). 'Talking about Code: Integrating Pedagogical Code Reviews into Early Computing Courses'. In: *ACM Transactions on Computing Education (TOCE) - Special Issue on Alternatives to Lecture in the Computer Science* 13.3, 14:1–14:28. DOI: 10.1145/2499947.2499951.

Jazayeri, Mehdi (2004). 'The education of a software engineer'. In: *Proceedings of the 19th International Conference on Automated Software Engineering*. Linz, Austria: IEEE, pp. xviii–xxvii. DOI: 10.1109/ASE.2004.1342718.

Koch, Aaron et al. (2002). *The redesign of studio culture: a report of the AIAS Studio Culture Task Force*. Tech. rep. Washington, DC, USA: American Institute of Architecture Students (AIAS).

Kopczyńska, Sylwia, Jerzy Nawrocki and Mirosław Ochodek (2012). 'Software Development Studio – Bringing Industrial Environment to a Classroom'. In: *First International Workshop on Software Engineering Education based on Real-World Experiences (EduRex)*. Zurich, Switzerland: IEEE, pp. 13–16. ISBN: 9781467318051. DOI: 10.1109/EduRex.2012.6225698.

Kuhn, Sarah (1998). 'The software design studio: An exploration'. In: *IEEE Software* 15.2, pp. 65–71. ISSN: 07407459. DOI: 10.1109/52.663788.

Kuhn, Sarah (2001). 'Learning from the architecture studio: Implications for project-based pedagogy'. In: *International Journal of Engineering Education* 17.4/5, pp. 349–352.

Kvan, T and Y Jia (2005). 'Students' learning styles and their correlation with performance in architectural design studio'. In: *Design Studies* 26.1, pp. 19–34. ISSN: 0142694X. DOI: 10.1016/j.destud.2004.06.004.

Lackney, Jeffery A. (1999). *A History of the Studio-based Learning Model*. URL: http://www.edi.msstate.edu/work/pdf/history_studio_based_learning.pdf (visited on 27/10/2012).

Lawson, Bryan (2012). *How Designers Think*. 4th. Routledge. ISBN: 9781136398001.

Ledewitz, Stefani (1985). 'Models of design in studio teaching'. In: *Journal of Architectural Education* 38.2, pp. 2–8.

Lee, Jaejoon et al. (2015). 'Software Design Studio: A Practical Example'. In: *Proceedings of the 37th International Conference on Software Engineering (ICSE '15) [in press]*. Florence, Italy: IEEE.

Löwgren, Jonas and Erik Stolterman (2004). *Thoughtful Interaction Design: A Design Perspective on Information Technology*. Cambridge: MIT Press. ISBN: 9780262296922.

Lynch, Kathy et al. (2002). 'A studio-based approach to teaching information technology'. In: *Proceedings of the Seventh world conference on computers in education conference on Computers in education: Australian topics - Volume 8 (CRPIT '02)*. Ed. by Anne McDougall, John Murnane and Dianne Chambers. Darlinghurst, Australia: Australian Computer Society, Inc., pp. 75–79.

Mark, Gloria (2002). 'Extreme collaboration'. In: *Communications of the ACM* 45.6, pp. 89–93. DOI: 10.1145/508448.508453.

Mazur, Eric (1997). *Peer Instruction: A User's Manual*. 1st ed. Upper Saddle River, NJ, USA: Prentice Hall. ISBN: 978-0135654415.

Mazur, Eric (2009). 'Farewell, lecture?' In: *Science* 323.5910, pp. 50–51. DOI: 10.1126/science.1168927.

McCall, R. (2013). 'Critical Conversations: Feedback as a Stimulus to Creativity in Software Design'. In: *Creativity and Rationale* 6.May, p. 11.

Morkel, Jolanda (2011). 'The Social Dimension of Studio Space: Face-to-Face and Beyond - Exploring the Online Learner Experience'. In: *Sixth International Design Education Forum of Southern Africa Conference*. DEFSA, pp. 139–145.

Narayanan, N. Hari et al. (2012). 'Transforming the CS classroom with studio-based learning'. In: *Proceedings of the 43rd ACM technical symposium on Computer Science Education (SIGCSE '12)*. Raleigh, NC, USA: ACM, pp. 165–166. ISBN: 9781450310987. DOI: 10.1145/2157136.2157188.

Oh, Yeonjoo et al. (2013). 'A theoretical framework of design critiquing in architecture studios'. In: *Design Studies* 34.3, pp. 302–325. ISSN: 0142694X. DOI: 10.1016/j.destud.2012.08.004.

Petre, M. (2013). 'An online design studio'. In: *ACM Inroads* 4.3, pp. 44–45. ISSN: 10730516. DOI: 10.1145/2505990.2505998.

Prior, Julia, Andrea Connor and John Leaney (2014). 'Things coming together: learning experiences in a software studio'. In: *Proceedings of the 2014 conference on Innovation & technology in computer science education*. New York, NY, USA: ACM, pp. 129–134. ISBN: 9781450328333. DOI: 10.1145/2591708.2591720.

Ramakrishnan, Sita (2003). 'MUSE studio lab and innovative software engineering capstone project experience'. In: *ACM SIGCSE Bulletin* 35.3, pp. 21–25. DOI: 10.1145/961290.961521.

Reimer, Yolanda Jacobs and Sarah A. Douglas (2003). 'Teaching HCI Design with the Studio Approach'. In: *Computer Science Education* 13.3, pp. 191–205. DOI: 10.1076/csed.13.3.191.14945.

Salama, Ashraf (1995). *New Trends in Architectural Education: Designing the Design Studio*. Raleigh, NC, USA: Tailored Text. ISBN: 9780964795006.

Schön, Donald A. (1983). *The reflective practitioner: How professionals think in action*. London: Temple Smith. ISBN: 9780465068784.

Schön, Donald A. (1985). *The Design Studio: An Exploration of its Traditions and Potential*. London: RIBA Publications. ISBN: 9780947877453.

Schön, Donald A. (1987). *Educating the Reflective Practitioner*. First Edit. San Francisco, CA: Jossey-Bass. ISBN: 9781555422202.

Schön, Donald A. (1991). *The Reflective Turn: Cases Studies in and on Educational Practice*. Teachers' College Press. ISBN: 978-0807730454.

Shaffer, D.W. (1997). 'Design, collaboration, and computation: The design studio as a model for computer supported collaboration in mathematics'. In: *Proceedings of the 2nd international conference on Computer support for collaborative learning (CSCL '97)*. Ed. by Rogers Hall, Naomi Miyake and Noel Enyedy. International Society of the Learning Sciences, pp. 253–258.

Shaw, Mary et al. (2006). 'Deciding What to Design: Closing a Gap in Software Engineering Education'. In: *Software Engineering Education in the Modern Age* 4309, pp. 28–58. DOI: 10.1007/11949374_3.

Siegel, Martin A and Erik Stolterman (2008). 'Metamorphosis: Transforming non-designers into designers'. In: *Undisciplined! Design Research Society Conference 2008*. Sheffield, UK: Sheffield Hallam University, pp. 378/1–378/13.

Suddaby, Roy (2006). 'From the Editors: What Grounded Theory is Not'. In: *Academy of Management Journal* 49.4, pp. 633–642. ISSN: 1539-6924. DOI: 10.5465/AMJ.2006.22083020.

Süß, Jörn Guy and William Billingsley (2012). 'Using continuous integration of code and content to teach software engineering with limited resources'. In: *Proceedings of the 34th International Conference on Software Engineering (ICSE '12)*. Zurich, Switzerland: IEEE, pp. 1175–1184. ISBN: 978-1-4673-1067-3. DOI: 10.1109/ICSE.2012.6227025.

Thomas, John W (2000). *A review of research on project based learning*. URL: http://www.bobpearlman.org/BestPractices/PBL_Research.pdf (visited on 26/12/2013).

Tomayko, James E. (1991). 'Teaching software development in a studio environment'. In: *ACM SIGCSE Bulletin* 23.1, pp. 300–303. DOI: 10.1145/107005.107070.

Tomayko, James E. (1996). 'Carnegie Mellon's software development studio: a five year retrospective'. In: *Proceedings of the 9th Conference on Software Engineering Education*. Daytona Beach, FL, USA: IEEE Computer Society Press, pp. 119–129. DOI: 10.1109/CSEE.1996.491367.

Uluoğlu, Belkis (2000). 'Design knowledge communicated in studio critiques'. In: *Design Studies* 21.1, pp. 33–58. ISSN: 0142694X. DOI: 10.1016/S0142-694X(99)00002-2.

University of British Columbia (2014). *HCI Learning Studio*. URL: https://www.cs.ubc.ca/our-department/facilities/hci-learning-studio (visited on 13/06/2014).

University of Nebraska-Lincoln (2011). *'Design Studio' [online]*. URL: http://raikes.unl.edu/designstudio.shtml (visited on 19/09/2011).

Vygotskiĭ, L.S. (1978). *Mind in society: The development of higher psychological processes*. Ed. by Michael Cole. 14th. Harvard University Press. ISBN: 0674576-292.

Wallingford, Eugene (2011). *'Software Studio' [online]*. URL: http://www.cs.uni.edu/~wallingf/teaching/162/studio.html (visited on 19/09/2011).

Wang, Tsungjuang (2010). 'A New Paradigm for Design Studio Education'. In: *International Journal of Art & Design Education* 29.2, pp. 173–183.

Watkins, Jessica and Eric Mazur (2013). 'Retaining Students in Science, Technology, Engineering, and Mathematics (STEM) Majors'. In: *Journal of College Science Teaching* 42.5, pp. 36–41.

Whittle, Jon et al. (2014). 'Teaching in a Software Design Studio: Implications for Modeling Education'. In: *Proceedings of the 10th Educators' Symposium (EduSymp '14) at MODELS 2014*. Valencia, Spain: CEUR-WS.

Appendix A: Focus Group Questions and Discussion Points

Below are questions and discussion topics that were covered during the focus group with the studio's students.

Due to the semi-formal nature of the discussions, and the tangents that conversations took during the focus group, these questions were not asked in any particular order. The questions have been grouped roughly into categories that resemble the studio framework, as defined in chapter 3.

The students were not informed of the questions beforehand, and they were not informed of how the questions were grouped together – the grouping was intended as data management technique during the focus group, so I could quickly scan the question categories to see if a question was covered within the course of the discussions.

Defining the Studio

- Did you have any preconceptions of what the studio would be like?
- How would you describe the studio currently?

General Questions

- How well are you doing? Do you think the work is going well?
  - How is the module in general?
- How much do you feel you have improved this year
  - Which modules would you most attribute this to?
  - Has the studio been a positive or negative force regarding this?

- How much time and effort have you put into this module compared to others?
- Can you foresee any difficulties leading up to submission?
- Will your use of the room increase/decrease?
- Time spent in studio
  - How often did you think you'd spend
  - How often are you in the studio
  - Do you think this may increase/decrease?
- What issues you have encountered
  - File sharing, for example

Physical environment

- Nesting
  - Static seating/nesting patterns
- Congregating around computers, even though only using pen and paper
- Do you have enough space?
- Do you feel it is secure enough to leave things around? Would you?
- Storage?
- Is the space flexible enough for you
  - How have you reconfigured the space, if at all?
- Does the room offer suitable privacy?
- How will the studio be affected by next years intake?
- Proximity to walls?
  - Has this affected the style of work you do?
- What do you do when you need private or individual space/time?

Facilitation of studio

- Thoughts regarding rules/freedom (food and drink)?
- Is the room equipped enough for your needs?
- Teaching styles
  - Various styles used (different people)
  - Lecture vs doing
  - Thoughts on sessions that are heavy/light on PPT slides?

- Attitudes towards "gather round" & lecture style semi-circle.
- Length of session
- How often is the space used outside of timetabled slot?
- Do you feel the Lecturers/TAs are approachable?

Modes of education

- Do you think you are learning of each other?
- How do you feel towards the projects
  - . . . in comparison to other projects in other modules?
- Thoughts on learning additional skills to fulfil your project?
- Use of devices and whiteboards as reference material.
  - Not directly/constantly interacting with them.
- How well does the studio work for ideation vs implementation.
  - Is there a difference?
- Various stances (standing sitting)
- Mentor meetings
  - How helpful are they?

Awareness

- Visual nature of work
  - When it is visible, do you benefit/appreciate it more
- Work appears to stay on whiteboards permanently?
  - Do you deliberately display work
  - Can others see?
- How much do you know about the other groups projects?
  - Are you friends or just colleagues with the other groups
- Has an awareness of others work lead to any social interactions?

Critique

- Presenting your work to others
- Do you critique each other groups work
- Do you critique within your group (outside of presentations)
  - More or less than other types of module environment.

  - Do you do it often?
- Presenting to the room
  - How comfortable are you with this?
  - Do you benefit from this?
  - (optional) Do benefits outweigh the negatives?
- Thoughts on peer-testing

Culture

- Ownership of space
  - How do you perceive this space
  - Home, work, somewhere in between - discuss!
  - Fridge? etc
- Is the room used for other work/activities
  - Coursework, parties etc.
  - How about non-work related activities alongside work
- Is this space social? Should it be social?
- Absence of individuals
  - Any particular reason why (not naming names)?
  - Some will obviously be legitimate.
- Proximity to other groups may cause silence
  - Being aware not to disturb the other group?
- Procrastination: studio vs. other labs
  - Does the amount vary? Why?
- Does the room provide a good work ethic?
- Peer-learning

Individual's characteristics

- Do you work here because it is quiet?
- How personal does the space feel
  - Have you personalised it at all?

Inspiration

- Use of real projects – ones with potential to be properly deployed.

- Does being in the studio inspire you at all?
  - Proximity to others.
  - Anything else that helps inspiration.

Collaboration

- Use of whiteboards?
- Interaction between groups – how often do you work and interact?
  - Are you too focused on project work in these sessions
- How do you decide to divide work amongst group?
- Huddling around a single monitor
  - What is going on?
- Certain members being put on the spot during presentations
  - More talkative, nominated, etc.?
- You are always working in the same spots:
  - What are your thoughts to having impromptu break-out spaces?

Digital technology

- Use of computers/laptops
- Computers used as second monitor
  - Multiple computers/devices used
  - Is a single monitor too limiting? Why?
- Use of other tech in the room (TV, touch table etc.)
- Mix of digital and analogue mediums
- Observation: issue with sharing/working/presenting digital work?
- Computers used to "type up" or formalise work that is discussed or white-boarded etc.
- When you sit down to work on something, you gravitate towards a screen. Why?
  - How would things be different if we removed some of the computers?
  - Half the room
  - Or, Every other computer etc.
  - Do you need screens for everything?
- Your computer use changes based on activity
  - Is this ideal?

  - Is it comfortable?
  - How could this be changed?

- Do you think monitors reduce social interactions?
- In your opinion, how visible and accessible is digital work to others?