Successful Studio-Based Real-World Projects in IT Education

Matthew Simpson, Jay Burmeister, Alan Boykiw and Jihan Zhu

Information Environments Program,
School of Information Technology & Electrical Engineering
University of Queensland
Brisbane, 4072, Australia

{matts, jay, boykiw, jihan}@itee.uq.edu.au

Abstract

An important aspect of University teaching is to promote deeper learning by ensuring students have good exposure to problems they will encounter in their working life. While it is not possible to reproduce every situation that will occur in an area of employment, it is possible to provide students with experience from which they can utilise principles and approaches in order to gain the necessary skills to address whatever scenario may arise. This paper looks at the nature of studio-based teaching and the pedagogy that supports it, through examining two case studies as a catalyst for exploring real-world projects. It is through the discussion of aspects such as real-world clients, user involvement, sequencing and integration that the success of studio-based teaching in IT can be revealed.[1]

Keywords: Studio-based teaching, Real-world projects, Real-world clients, IT education, Extreme programming, Ethics.

1 Introduction

This paper aims to provide an insight into a studio-based approach to providing real-world experience through IT teaching. The studio model works by integrating aspects of real-life projects throughout the duration of the degree, gaining maximum exposure to the real-world principles.

Based upon a history of architecture, design and studio pedagogy, studio gradually exposes students to the issues, situations and scenarios they will potentially encounter in the work environment. The exposure occurs over the full three years of the degree addressing issues such as real-world client involvement, individual project work and team environments (both internal and external to a project team), and students as clients. These issues reveal the relevance of studio to provide three years of experience as opposed to three years of learning.

The nature of studio pedagogy is discussed in Section 2, with Sections 3 and 4 describing two real-world studio projects. Issues related to successful real-world studio-based projects and their relation to IT education are discussed in Section 5, followed by conclusions in Section 6.

Copyright 2002, Australian Computer Society, Inc. This paper appeared at the Australasian Computing Education Conference (ACE2003), Adelaide, Australia. Conferences in Research and Practice in Information Technology, Vol. 20. Tony Greening and Raymond Lister, Eds. Reproduction for academic, not-for profit purposes provided this text is included.

2 Studio Pedagogy

The Information Environments Program represents a radical shift in course design and pedagogy in the field of information technology education. The Bachelor of Information Environments (BInfEnv) degree is a three year degree (four years with the optional honours year) which consists of streams of courses in four main areas: Information Technology (IT), Design, Information Environments and the integrative Studio stream. At the time of writing there were no other studio-based IT programs in Australia. Project based learning is common but a sequenced, integrative and designed studio curriculum is a unique aspect of the Information Environments degree program.

At the core of the Bachelor of Information Environments is a studio-based approach to teaching and learning, modelled on the design or architectural studio, which encourages a community of learners to interact to solve problems. This is in response to research that shows that it is no longer sufficient to pursue inflexible, sequential and compartmentalised IT development methods or to focus on delivering content in relative isolation of the contexts in which it is to be applied. Brown, Collins and Duguid (1989) argue that this separation of content (what is to be learned) from context (how it is learned and used) will not result in deep learning. Students can "acquire", and even manipulate, algorithms, routines and de-contextualised definitions, but this does not mean that they will be able to apply them to new contexts. Lave (1996) also argues that learning is a process that takes place in a participation framework rather than in an individual's mind. Software engineering skills that have been taught in isolation result in little transfer to new contexts, and, an added complication is that, given the rapid developments in technology, the content and skills presented to students in this way are likely to have changed almost before graduates have gained their first jobs.

The studio-based teaching approach offers students an opportunity to solve real design problems in ways that mirror the work of professionals in the world of information technology: through teamwork, collaborative learning and the application of related knowledge to new contexts (Docherty and Brown 2000). Knowledge and skills are acquired in context rather than as separate segments "to be learned". This approach closely relates to social constructivist theories of learning (Jonassen, Davidson, Collins, Campbell, and Haag 1995), which argue that learning is necessarily a social dialogical

process in which communities of learners socially negotiate the meaning of the phenomena. Meaning is constructed through collaboration and conversation by the learners, rather than through the passive receipt of "uncontested knowledge" delivered in a one-way flow of information by the lecturers.

Brandsford, Sherwood, Hasselbring, Kinzere, and Williams (1990) found that many software engineering students face difficulties when it comes to problem solving – they often have no idea where to begin, despite their familiarity with the syntax of programming languages. They can memorise facts and procedures, but have difficulty in explaining observed phenomena, or solving real-world problems or analysing problems and thinking critically. Many of these students may pass examinations, but have trouble generalising their learning from one situation to another, leading to a skills gap every time the job, content or technology changes. Future employers expect information technology graduates to be able to meet the rapid changes of technological innovation, and to work in creative and responsive teams. Traditional teaching methods do little to prepare students for these demands.

Studio-based teaching aims to foster creativity, reflection, articulation, and reasoning, all of which are important lifelong learning skills and valuable graduate attributes. The purposeful use of technology and of group work is central to these aims. Through this approach to studio-based teaching, real-world projects can be applied with both the students and the client benefiting from a rich, industry-orientated format. The nature of these outcomes is explained in the following Section.

2.1 The Use of Real-World Projects in Studio

The integration of real-world projects into studio aims to gradually expose students to situations and skills, through a series of projects, which increase the level of real-life exposure.

In Studio 2 (first year), students work to a project brief developed for a real-life client. The level of exposure to the client is kept relatively limited, with the client providing their needs and the students not meeting with them again until the final presentation of the designed solutions.

In Studio 4 (second year), the client is internalised with the lecturer acting in this role. Through the course of the project, students are expected to constantly interact with the clients as the requirements are adapted and fine-tuned. This process enables students to learn how to address and respond to client needs, present work in progress and manage client expectations.

Once students reach third year, the real-life client is once again introduced into the process and the students are expected to manage projects and expectations alike. It is this gradual exposure that enables students to learn one step at a time, a theme that is described more fully in Section 5.2.

2.2 The Studio Curriculum in the Information Environments Degree

The Bachelor of Information Environments degree is much more tightly integrated than a traditional IT degree with a large number of compulsory courses. This allows the studio courses to assume particular knowledge and prerequisites on the part of the student, better supporting the consolidation and integration aspects of the studio stream. Studios in second semester are used to integrate knowledge and are much more focused on implementation than studios in first semester, which are more focused on design.

Studio is used in the Information Environments degree structure to prepare students for the outside world. One of these aspects is the gradual ramping of skills and principles over the three years as mentioned above. Another is the system of exposure and application. Predominantly, studio is structured in such a way that the first semester of any given year focuses upon theory and design. In first semester studios there tends to be a divergence, where exposure is gained to an assortment of methods, techniques and approaches relevant to that particular year. Second semesters studios are aimed at the application of these skills, that is, on convergence. Students work on projects and designs that are aimed at drawing together the knowledge, skills and practices they have learnt from earlier studios and associated courses in the degree. The approach integrates well with the model of gradual exposure to principles and situations, by enabling students to practice techniques, and then in the next semester, to integrate them into a larger design process.

The role of studio in a larger sense is to act as a medium for integrating and utilising skills obtained from courses offered previously or concurrently. The streams that make up the Information Environments degree are shown below.

Studio 2 and 4 were chosen as case studies, for this paper as they are implementation studios as shown in Table 1. Studio 6 is an implementation studio; its focus is singular and up to students' individual project directions rather than group orientated, and therefore, the outcomes are harder to correlate.

	Year 1	Year 2	Year 3
1st Semester Design	Studio 1	Studio 3	Studio 5
2nd Semester Implementation	Studio 2	Studio 4	Studio 6

Table 1: The nature of studios across the degree.

The gradual ramping of skills and exposure to real-world clients and situations is shown through the two case studies below. The first for these (Section 3) describes the first year course, Studio 2, and the second (Section 4) describes the second year course, Studio 4.

3 Case Study 1: E-News Studio 2 Project

The aim of the project was to provide an opportunity for education to assist in addressing multi-disciplinary issues and development of online collaboration, between all parties involved. Specifically this involved a project to develop a website for the creation and modification of online news stories. The sites were required to be highly usable and easily updateable. Another aspect of the project was to explore the impact that interactive technology can have on the presentation of information that has traditionally been displayed in printed form. This was to be achieved through the re-representation of a real-life design environment in a studio-teaching model. By teaching in such a manner, students are able to engage the client and gain experience in the nature of the design world. The nature of this pedagogy is effective not only in aiding the instruction and learning of students, but also in creating a realistic environment for testing new methods and approaches in industry.

3.1 The E-News Project

One of the aims of the E-News project was to facilitate the interaction of two sets of physically distributed students involved in real-life projects. This revealed the nature of online communication, and enabled them to maintain continuity and connection across considerable distance. E-News was a collaborative project involving students from Information Environments (IE) at the Ipswich campus of the University of Queensland (UQ) and E-Journalism students at the Rockhampton campus of the Central Queensland University (CQU) conducted in the second semester of 2000. The Information Environments students were required to function as a team of web designers in order to develop a fully functioning database-driven website for an external client (shown in Figure 1). The E-Journalism students acted as the client for this purpose, providing the requirements and the content for the site.

Each group formed by IE students consisted of four to five members. They were required to form and maintain a balance of appropriate roles in the team. These include design, coder, content editor and graphic designer. For each of the 12 Web design groups, three students from E-Journalism acted as clients, allowing for group interaction of a sustainable size to occur.

Figure 1: An example of a completed E-News database website.

3.2 Final Product

In accordance with the nature of the studio pedagogy, the Information Environments students were briefed about the nature of the project, technical attributes and the context of the material to be displayed. Over the eight-week duration of the project, the clients informed the development of a design brief for the project, creation of initial concept designs and the selection of several preferred designs, which were short-listed through an online presentation. From this point the IE students formed into web design production teams. The structure of the teams was similar to professional web design production teams, consisting of graphic designers, project managers, interface designers, programmers, and other relevant members of the team (Di Nucci, Giudice, and Stiles 1998).

Communication with the clients utilised a technological approach, with initial communication between the two parties occurring through video conferencing. This enabled initial introductions between group members, formation of conceptual ideas and project understanding between both parties. This level of discussion and the establishment of understanding was comparable to that achieved through a physical meeting. By "breaking the ice" through video feed, a familiarity between the clients and designers began to develop and continued through other less "physically constraining" forms of communication. From this point, most communication and discussion occurred through email and bulletin board discussion. During the group development process the clients reviewed the designs online via websites. The clients provided feedback to the individual website production teams through the medium of email.

The constructed websites went live at the end of the eight weeks. The end result was an online design presentation to the clients, involving the clients conducting user testing on the sites. For user testing purposes, the journalism students utilised the site as one would work with a real site. The journalists split into groups and uploaded an assortment of news stories over the course of the following weeks, including images, video footage and sound files.

The final sites consisted of two major sections: the public website displayed the articles, and the administration section enabled the upload of articles and associated files by journalists. Access to the administration section required a login and password (see Figure 2).

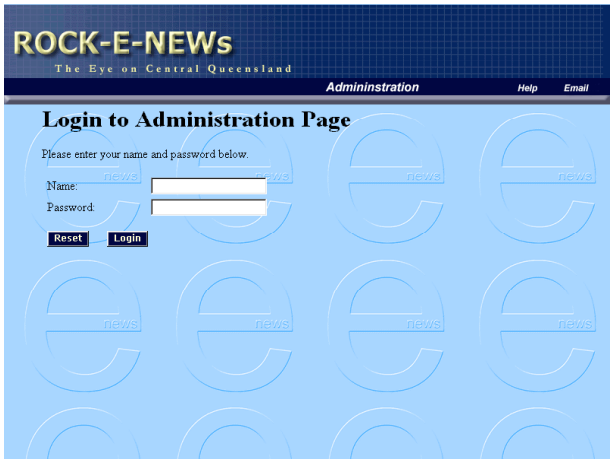

Figure 2: Administration login page.

The nature of the administration section enabled the journalists to be able to upload, edit and delete their stories with relative ease, through an easy-to-use web-based system (as shown in Figure 3). Journalists only needed to enter a title, summary and cut-and-paste the story into the text field. Adding the local file name and location on the computer through a series of simple online operations would upload all associated files. In some sites the ability to choose the location of the image was also provided, i.e. next to the header in the body of the article or at the end, depending on the context of the image.

When a story was uploaded, journalists could view the layout of the article with the ability to remove it, e.g. the layout was unsuitable for public viewing. This enabled the journalists to have editorial control over their own work. To assist the journalist with usability of the sites, an extensive help section was created to guide less experienced users through the process of uploading stories.

Figure 3: Adding news articles.

3.3 Outcomes

The final outcome of the Studio 2 E-News project was to provide a functional prototype of the system. The key underlying aim of producing this system was to enable users who only have basic computer skills and understanding to be able to use the website with competence, by providing usable layout, well structured pages, simple upload mechanisms and clear instructions. With current trends in computer usage in society, the basic skill set is increasing. A user who can run a computer and browse the web has all the skills required to utilise this web-based system.

As a studio outcome, the students gained exposure in the implementation of a conceptual design project through a "whole of process" exposure. Taking a client's needs and initial brief, through conceptual development, design, construction and prototyping of an artefact, enabled a level of exposure not unlike the process in a work environment.

The experience of implementing a design outcome was invaluable for providing students with an insight into the process of design. Students were able to experience how one must change a brief's constraints to include not just technical implementation, but also to maintain initial client outcomes. Students begin to appreciate the consideration necessary in the scope of a project and how to laterally produce an outcome for a technically complex task.

The process of testing and debugging a project is very closely associated with the consideration of project scope. The act of creating code and applying it to what is essentially a commercially viable artefact is far different to producing an isolated program formed based on a static set of project goals.

Ultimately, students saw a tangible outcome from their studio work for a real client, rather than just a set of grades and learning outcomes. Such an outcome is difficult, if not impossible, to achieve without a real-world project.

4 Case Study 2: The Virtual Café Studio 4 Project

Studio 4 occurred over a period of 13 weeks and was the fourth studio undertaken by second year students in the Information Environments degree. The emphasis in Studio 4 is on the production of working prototypes based on research and ideas.

Studio 4 was designed to explore information environments from the perspective of invisible or ambient computing technology. This is in contrast to the current genre of PC workstation applications that is the dominant focus of information technology education initiatives. The ambience domain was chosen to encourage students to examine new ways of communicating and interacting with information through unique technology solutions. The abstract nature of invisible computing prompted the need to provide a tangible and physical focus in order to achieve the learning outcomes mentioned above.

A virtual café was chosen to be the information environment of interest for the duration of the course. Human experience was emphasised within this context. Off-the-shelf hardware such as sensors, microphones, lights, speakers and smart circuit boards were to be combined with software applications to design and build physical demonstrations of a café concept.

4.1 Authentic Environment

The students proceeded through various phases requiring both individual and group contributions to the overall project. These phases were modelled on typical IT industry practice based on the lecturers' industry experience. This course was taught by two lecturers who took roles within the scenario as development manager and client, rather than traditional lecturing roles. This entailed managing the students as a development team, inspiring them to present solutions to the "client" as the project progressed. At first, the students could not grasp that the client was "always right" and that they had to be articulate and convincing of the value of their proposals rather than just handing in their project ideas for a grade. They very rapidly caught on to this idea and relished the challenge of impressing "the client". This is an example of encouraging the development of sought after graduate attributes through authentic environments.

The initial phases of the project were conducted in groups of four students. Their tasks included ethnographic observation and interview of café user groups, physical/virtual problem and opportunity analysis, and information flow evaluation. This research of the different types of users in the café scenario provided the students with a set of needs and problems for which they could design solutions. For example, owners were consistent in identifying that changing menus and specials was a problem that resulted in a significant cost to them. It was also the early stages of the students being proud that they had discovered a problem rather than being given one, encouraging deeper thought and reflection.

The next phase entailed making individual preliminary design proposals, and physical and virtual design mock-ups that would satisfy identified user needs. Hardware technology was also evaluated and the students presented ways of applying them to satisfying user needs (user-centred design). The designs were evaluated in terms of suitability for creating an overall café experience that demonstrated notions of ambient and invisible computing to a lay audience. The reality of constructing the designs within the allotted time for the program was also considered since it is typical of industry practice. Both students and lecturers participated in a review session to determine the most feasible alternatives.

The notion of representing the ambience of a café is abstract, so the idea of building a physical demonstration of this area of research provided a tangible focus for the work by the students. This scenario also provided a context for the application of theory and technology from other design and IT courses. For instance, the Java programming language that was used for all the software development was taught in another course. The detail of the network architecture that was used in the implementation of the final demonstration was taught as a separate workshop.

The development of the virtual café software followed a practice called eXtreme Programming (XP) involving user story cards, pair programming, task assignment, time estimates and test case development (Beck 1999, Beck 2000). This is a highly learner-centred and activity-focused approach and it was selected because of its emphasis on rapid, nimble, iterative development with a strong user focus. Students took intimate ownership of the outcomes resulting in a deeper appreciation of the rationale and application of effective design processes and methodology. XP is described more fully in Section 5.4, and in relation to studio-based teaching in the next Section.

4.2 eXtreme Programming (XP)

eXtreme Programming (XP) (Beck 1999, Beck and Fowler 2001) is a flexible software development methodology that encompasses many valuable user-centric design principles and usability values.

4.2.1 How does XP fit with Studio-Based Real-World IT Project

In a studio-based IT project course, as with any project-based IT course, a project management process must be adopted, upon which student projects can be organised and managed. The question is what kind of project management process is best suited for studio-based real-world IT projects. In Studio 4, XP was adopted as the software development methodology to support the project management process because of its flexibility, consistency with user-centric design principles, and usability values.

XP is an agile software engineering process for small to medium sized programming teams, and is based on a set of well-known software engineering practices and values. Unlike traditional software engineering processes, XP values people over process, encourages communication between clients and developers, and communication

between developers within a team. XP emphasises simplicity over hefty design and over-engineering. The driving principle is to ask, "What is the least we can do to build good software?". Thus, time is never wasted on over-engineering for future needs and hefty designs. XP embraces changes over insistence on following a plan. These attributes maps very well with conducting real-world IT projects in studio settings.

The XP project lifecycle involves iterative development. Unlike the traditional approaches, an XP project is divided into small releases and iterations. Each of these iterations contains all four activities of software engineering: analysis, design, implementation and testing. Each of the releases contains a fully functional product (though the functionalities may be limited depending on the scope chosen by the client in each release). Thus, in an XP setting, a student can be expected to experience a full range of activities in software engineering, and to have the chance to participate in all activities several time during the project. In a traditional model (such as the waterfall model), however, it is difficult for students to follow even a single iteration of software engineering activities in a semester-length project course, let alone practising them several times.

XP is implemented as follows in our studio settings. A feature set for the software is initially created with the on-site client and the developers then estimate the work needed to build each feature (here students exercise their judgement by trading project scope, qualities and budget, with the time required to implement the feature). Individual features are written on cards and called "user stories" and they are implemented in an order consistent with what features we are certain are going to be in the finished product. Such features are implemented using pair programming, a core principle of XP. The whole class can be divided into a set of four to six member teams. Different scenarios and implementations are encouraged, but all these could be based on the same set of user-stories. The role that students take is primarily as software developers or sometime as clients (see Section 5.4.1). The academic staff and tutors take on the role of project leaders and their main duties are mentoring and managing overall progress of multiple projects.

4.3 Outcomes

The final result of the students' efforts was a physical manifestation of the software code and hardware configurations that they had developed throughout the semester. The software that was developed enabled hardware such as touch and motion sensors, microphones, lights, interactive whiteboards (SmartBoards), projectors and speakers to create a unique café experience.

During the evening that the café was set up for inspection, it was very evident that student learning went far beyond the classroom learning outcomes. Students fielded questions about their designs and were engaged in discussions with IT professionals reinforcing the reality of what they had researched, designed and built. In most cases, the discussions centred on user interaction and value rather than on the technical achievements.

Figure 4 shows a prototype of an interactive menu that allowed guests to order by touching items. It also demonstrated that an owner could change and update the menu from a remote office PC. The ability for them to see that their research had led to a design concept, and that the invited guests appreciated the user value, was a convincing validation of the design process concept. This type of learning affirmation is not possible in a theoretical or simulated project scenario.

Figure 4: Demonstration of a prototype of an interactive menu.

Figure 5 shows an intelligent environment where lighting precedes a patron of the café to guide them to their table. The user is unaware that a computer is reacting to their needs and assisting them in their task. Demonstrating invisible computing is a challenge faced by researchers globally so the process of rapidly building a prototype was of genuine interest to researchers and industry guests. Again, the students revelled in the fact that they were representing "new" ideas grounded in an authentic project process environment rather than handing in an answer to an assignment that was relatively similar to their peers the previous year. Further evidence of enhanced learning outcomes is supported by course evaluations. Student comments consistently articulated, "… the studio is the place where we pull everything together like the real world and that's excellent".

Figure 5: Intelligent environment.

From a technical perspective, the achievement of having the students build the prototype should not be understated. The software that was developed by the students through this process enabled hardware such as touch and motion sensors, microphones, lights, interactive whiteboards (SmartBoards), projectors and speakers to communicate to create the unique café experience. The idea of building things in contrast to cursory code reviews appealed to the students and was reflected in their course evaluations.

The students did not need to be taught to configure technology in new ways to create the effects they had researched and designed. They had an inherent desire to see the results of their research come to fruition. Many late nights were spent on experimenting and pursuing ways of achieving desired outcomes. Other examples of technology application were: chairs signalled when they were occupied; microphones picked up the level of sound in the café and this "ambience" was represented as graphic images on a remote sign; live camera images of guests were displayed and could be streaked and altered with the stroke of a finger on a display board. Ensuing discussions of the ethics of collaborative technology concepts could be pursued as a result of people experiencing the idea rather than just discussing the concept.

The impact of the students' work would have been severely minimised without the overall context of the café and information environment focus afforded by the flexibility of the studio delivery process. The technology infrastructure associated with the campus also allowed students to pursue their passion for the project as many times they worked remotely into the night uploading and downloading code to and from the central repository.

5 Discussion

The nature of the two studios incorporates the various stages of the overall studio process, covering both limited exposure with clients through to continuous involvement with an in-house client, and individual through to integrated team-working environments. From these differing approaches many issues about both the nature of the studio-based teaching process and the advantages of student experience are exposed. These are discussed in detail below.

5.1 Real-World Clients

The involvement of clients in the process of design is paramount to achieving a successful outcome for all parties. Being able to discuss requirements, needs and wants, address issues and problems as they arise and produce a design outcome with which the client is satisfied, is integral to the process.

The studio approach works to integrate client interaction into its structure. The nature of involvement with a client can vary considerably between projects and there is a range of skills to be mastered. For this reason exposure to clients is a gradual process throughout the studios. These aspects can be split into several areas:

- Using students as clients.
- Moderation of client interaction.
- Managing client expectations.
- Transition from team to individually managed projects.

The nature of the aspects results in considerable cross over between the topic areas.

In first year studio, students gain exposure to all these issues. The level of exposure is minimal, serving as an introduction for the purposes of exposure and familiarisation. Studio 2 utilises the model of students as clients effectively. The students are able to gain exposure to issues of presenting progress, receiving feedback, interpreting feedback from the more familiar direction of their peers. The level of exposure to external clients is also limited in first year. Students are given a talk by the client, which entails, aims, objectives, needs and wants. Through lecturer guidance, students extrapolate the relevant information to produce a brief.

The role of managing client expectations is an integral part of the process. The art of interpreting a client's needs to explain how these aims are to be meet, and achieving those outcomes is the key to achieving client satisfaction.

Students are exposed to this process through an approach of gradually increasing responsibility. Issues such as timeframe, success criteria, and final outcomes are all addressed. Studio 2 works upon a system of having the client present during work presentations; Studio 4 (in second year) has more involvement with the teams of students accountable to the project managers and consultants (the role of lecturers); Studio 6 in the final year involves students directly communicating with external clients, with the role of the lecturers reduced to an advisory position.

The level of responsibility is also managed through a similar gradual process. Teamwork is conducted in conjunction (with individuals) in earlier years with projects of a shorter and less complex nature. A single project in Studio 4 becomes the focus, with individuals contributing the various aspects of the project goals.

Third year tends to see students completely responsible for all levels and parts of their projects.

Where external clients are involved in a third year studio project, their involvement is similar to traditional IT Honours projects. Students work individually on a project and liase with their external client periodically.

5.2 Sequencing

As described above in Section 5.1, issues related to studios are sequenced across the degree and also within a year. Students are involved in real-world projects in the second semester of each of the three years of the degree. The students' engagement in various aspects of the projects such as user and client involvement, technical knowledge and ethics is sequenced across each successive year. Each year, students become more involved with users and clients, utilise their increasing technical knowledge, and are required to go through a more rigorous ethical process.

5.2.1 User Involvement

User involvement is an important aspect of IT education since systems that are developed without adequately considering or involving users are at a higher risk of failure.

In first year studio projects, end-users (as opposed to clients) are involved in a minor or peripheral manner, if at all. The client, or academic staff, will often answer questions about what the users want etc. Thus, although not actively involved in person, the users are represented by "user advocates", and are therefore not absent from the students' consciousness.

In second year studio projects, students interact with end users to gain information that will help refine the design. Methods of interacting with users include questionnaires, surveys, interviews, and ethnographic observations.

In third year studio projects, students have to identify and secure the involvement of their own users.

Real-world projects provide an ideal setting for involving users in development. Sequencing the user involvement across the three years of the degree, that is, increasing user involvement each year, enables students to acquire skills in dealing with user issues gradually, rather than just at the end of their degree.

5.2.2 Technical Knowledge

The level of technical knowledge attained by students naturally increases across a degree. Thus, the technical challenge presented to students in successive real-world studio projects needs to be matched with their current technical abilities.

Students have a relatively low level of technical abilities in first year. The studio projects are constructed in a way that reduces the technical difficulty to a level that is sufficient to challenge the students, but not to discourage them by being too difficult. For example, the database backend may be implemented so that the students may concentrate their efforts on the interface frontend, and the functionality that should be delivered to the client and users. Students are encouraged to work in teams and to engage in peer teaching and learning, for example, students competent and interested in Java programming work with students who are less able Java programmers.

Second year studio students are more technically proficient, and are therefore expected to build on their base of technical knowledge, which is gained through a mixture of staff-directed, self-directed, and peer learning.

By third year, students are expected to be fully capable with respect to technical issues. Academic advisors are typically involved to give advice on domain and research issues, although they may give advice on technical issues, particularly on more technically focused projects. However, a high level of technical competence is assumed.

While making studio projects technically challenging in order to provide opportunity for students to consolidate and extend their technical capabilities, it is important to not provide so much challenge that the students do not have the capacity to address other non-technical learning outcomes.

5.2.3 Individual and Teamwork

Teamwork is a vital part of a student's education, and real-world studio project provide an ideal opportunity to teach students how to successfully engage in teamwork.

In general, students work both individually and in teams on the same project in first year studio projects. Initially, all students design solutions to the project. After students presented their designs, several designs are selected for teamwork. Each student continues with his or her own individual design, but also contribute to a team design. Thus, a many to one (teams to project) approach is taken, with the benefit that a single project may be used to meet the needs of the entire class. A similar approach to the mixture of individual and teamwork is taken towards design in second year studio projects. However, students do not pursue their individual designs. Rather, the clients (external and internal) select certain designs as having high value. Teams are formed around the students' interests, initiative and ownership of project proposals. The designs are then analysed with respect to technical and resource issues to eventually narrow the potential projects down to the final set of designs that are further developed. The teams are re-formed to support the chosen projects. The students then work in these teams to develop the designs through to completion. Team size varies form three members to as many as nine depending on the complexity of the project. Any given design may be re-scoped or cancelled due to unanticipated difficulties such as technical challenges. Such a cancellation is done in consultation with the clients (external and internal), and the project managers. At this point students are reassigned to existing projects.

Third year studio projects are almost exclusively individual projects similar in scope and nature to traditional IT Honours projects. It is possible for two or three students to work on different, relatively independent

aspects of a larger project, but in a weakly-coupled manner so that no student is disadvantaged by the lack of progress of any other student.

Working as an individual within a team environment is also an important skill for student's to learn. This issue is particularly addressed in Studios 1 and 3.

5.2.4 Presentations

The ability to effectively communicate to project stakeholders is an advantage in the workplace. Students are provided with many opportunities to present their work in a variety of ways in studios and other courses within the degree.

Starting in first year, students present their designs in individual and team presentations to academic staff and clients.

In second year Studio projects, students present their designs in individual and team presentations to academic staff and clients, but also present their final work in and end-of-year show. During the end-of-year show, they interact with members of the broader university community, IT, and business communities, as well as members of the general public. They are required to explain their work in a manner appropriate to the audience they are talking to.

As early as the second week of semester, third year students present design proposals, and continue to present their design work as frequently as every two weeks throughout the semester. As for second year students, third year students are involved in demonstrating and explaining their work to a wider community in the end-of-year show.

A major advantage of real-world projects is that it is possible to interact with stakeholders and the broader community with interesting and relevant content.

5.2.5 Ethics

In any project (or research) involving participants (e.g., user testing), it is important to treat people ethically. In general, this involves adequately informing them about what they will be doing, obtaining informed consent, and respecting their privacy and confidentiality.

In first year studio projects, students are introduced to ethical issues, but do not engage in any ethical applications since contact with users is minimal.

Through studio projects in second year, students are exposed to ethical issues at a higher level, although they do not have to engage in a formal ethical process. Academic staff handle ethical issues such as negotiating ethical behaviour with external clients and securing gatekeeper letters. Students are made aware of the ethical conduct that has been agreed with the external clients, and have to seek feedback from an academic before conducting research involving user contact, including approval of information sheets and questionnaires.

In third year studio projects, students go through a formal ethics application process, and are not permitted to conduct research involving user contact until their ethics application has gained approval. The ethical process is a peer review process rather than an adversarial process, and involves a review of their methodology, questionnaires, gatekeepers, consent forms, and information sheets.

Besides treating participants ethically, an ethical clearance process that reviews methodological aspects of a project before it begins results in fewer projects being compromised by unforseen difficulties.

5.3 Studio Integration

Studio courses are taught across all six semesters of the program as shown in Table 1. The structure and content of each studio course is based on the concept that design must be taught as a hands-on, project-focused, problem-based approach. Within the general structure and aims, staff devise suitable projects for each new semester that provide the learning vehicle for the pursuit of those aims. Skills that have been introduced in other courses are exercised and applied in the studios.

The studio stream has been designed to enable students, industry and staff to participate at appropriate levels over the duration of the degree. The learning outcomes in the six studios build on each other, providing students with the opportunity to develop and achieve the desired graduate attributes. This integrated and sequenced arrangement is unlike other approaches that tend to rely on students developing on their own and making progress toward acquiring the skills expected of graduates in an ad hoc manner, for example, acquiring the attribute of contributing in a team environment.

In the Studio 4 case study students worked as a development team constructing a project outcome, rather than focusing on completing a common assignment for assessment. The students learned about being accountable to their peers as a result of the interdependent nature of significant software development projects.

5.4 XP and its Benefits

The XP practice of pair programming was particularly crucial to the team-learning aspects of the course. Pair programming involves two developers working together on the same piece of code and swapping between being the coder and reviewer. The associated "pair-pressure" and "pair-learning" experienced demonstrates that pair programming is more efficient and accurate than coding solo. Weaker students learned from stronger peers while those who were advanced in their level of skills relished the challenge of friendly competition with their partners to improve their jointly owned design. Some students were at risk of not being able to complete the course successfully but were encouraged to become part of the project teams. The stronger students assumed leadership roles that were nurturing rather than the typical egotistical behaviour seen in tertiary learning environments. The competition was generative, and innovation occurred in short time frames without supervision. This practice is relatively new, but similar results are being achieved in industry.

One myth against studio-based IT project is the associated cost of implementing the studio-based teaching environments. The cost is manifested primarily in the personnel requirements (i.e., the number of academic staff and tutors needed to supervise student group works), and the difficulty of scaling up the teaching model with increased class size. Our experience and other research works (Cockburn and Williams 2001, Williams and Kessler 2001) shows that adopting an XP project management methodology actually reduces the teaching load and hence reduces the cost of personnel required involved for supervising the student projects. The results observed are:

- reduced consultation time needed with the students,
- reduced meeting time about team management, and
- reduction of the number of student questions.

The chores of lecturing staff involved in studio-based teaching have shifted from a normal cyclic routine of "lecture preparation – knowledge delivery - student assessment – feedback" to more managerial and advisory type of duties. Special lectures may be sourced from other lecturers in the program that have expertise and research interests in related areas. XP project management methodology fosters strong communications: student assessments are administrated on spot during project meetings; feedback is given instantly. Student problems are also solved on spot.

As a result, staff members are liberated from the burdens of studio project management duties and can spend more time on other aspects of their roles. Lecturers also get involved in other courses in delivering a module of learning from their research interests. This distribution of load and focus results in a levelling of workload across the program.

5.5 Students as Clients

XP methodology mandates that the client must be on-site, full-time, working closely with software developing teams. The client's duties are to answer questions from developers, set priorities, test and accept software releases, and steer iterations of the project so that developers can work as fast as possible. In an educational environment, however, this requirement is not always practical and sometimes can be cost-prohibitive. We have adopted a strategy to turn this difficulty to our advantage – "use students as clients". The types of graduate we endeavour to produce are project leaders or information architects who can contract projects and lead a team to build them. The ability to communicate cleanly and effectively the client's needs and requirements are essential activities for them in their future employment. Student skills in this area clearly need to be developed.

Selected students (usually the group managers and trackers initially) are asked to act as clients for their teams. Their tasks are liaising between the "real" clients and fellow group members. They conduct interviews, observations and ethnographic studies with the real clients. They take costumer needs, priorities and their understanding for the project back to the group. They take on the role of on-site clients to converse, represent the needs of clients and steer the progress of the project. By adopting this strategy, not only do we reduce the cost of having an on-site client, but also the students' aptitudes in client communication and project management are developed. The role reversal results in the following learning outcomes: students understand the role and duties of being a client, and develop skills in communicating client needs, prioritising them and accepting deliveries.

6 Conclusions

Through studio-based teaching, students have been exposed to the intricacies of real-world projects. The two case studies have revealed the effectiveness of the studio pedagogy when applied to the learning process. The process of studio as a whole-of-degree strategy has enabled students to be gradually exposed to skills, techniques, and situations, providing a transition through an appropriate juxtapose of learning and experiences. Client management and teamwork facilitation has been applied along with XP principles, to prepare students for their potential work environments. Through these issues the suitability of studio application in Information Environments to produce experienced rather than just learned students is evident.

7 References

BECK, K. (1999): Embracing change with extreme programming. *Computer.* **32**(10): 70-77.

BECK K. (2000): *Extreme Programming explained: embrace change*. Reading, MA, Addison-Wesley.

BECK K. and FOWLER, M. (2001): *Planning extreme programming*, Addison-Wesley.

BRANSFORD, J., SHERWOOD, R., HASSELBRING, T., KINZERE, C. and WILLIAMS, S (1990): Anchored instruction: Why we need it and how technology can help. In *Cognition, Education and Multimedia: Exploring Ideas in High Technology.* 115-141. HIX, D. and SPIRO, R. (eds). Hillside, NJ: Lawrence Erlbaum Associates.

BROWN, J., COLLINS, A. and DUGUID, P. (1989): Situated cognition and the culture of learning. *Educational Researcher*. **18**(1): 32-42.

COCKBURN, A. and WILLIAMS, L. (2001): *The costs and benefits of pair programming*. 223-245. In *Extreme Programming Examined.* SUCCI, G. and MARCHESI, M. (eds). Addison-Wesley,

DI NUCCI, D., GIUDICE, M. and STILES, L. (1998): *Elements of web design.* Peachpit Press. Berkley, CA.

DOCHERTY, M. and BROWN, A. (2000): Studio-based teaching in information technology. *Proc. ASET-HERDSA Conference 2000: Flexible Learning for a Flexible Society*, Toowoomba, 213-219, ASET-HERDSA.

JONASSEN, D., DAVIDSON, M., COLLINS, M., CAMPBELL, J. and HAAG, B. (1995): Constructivism and computer-mediated communication in distance education. *The Journal of Distance Education*. **9**(2): 7-26.

LAVE, J., (1996): Teaching, as learning, in practice. *Mind, Culture and Activity*. **3** (3): 149-64.

WILLIAMS, L. and KESSLER, R. (2001): Experimenting with industry's "pair-programming" model in the Computer Science classroom. *Journal on Computer Science Education*. **11**(1): 7-20.