Things Coming Together: Learning Experiences in a Software Studio

Julia Prior, Andrea Connor, John Leaney
Faculty of Engineering and Information Technology
University of Technology, Sydney
Sydney, Australia
julia.prior@uts.edu.au

ABSTRACT

We have evidence that the software studio provides learning that genuinely prepares students for professional practice. Learning that entails dealing with complex technical problems and tools. Learning that involves working effectively in groups. Learning that results in the building of students' self-confidence and the conviction that they can successfully deal with the challenges of modern software system development. Learning that allows the accomplishment of the more elusive professional competencies. In order for students to achieve this type of deep learning, they need time to immerse themselves in complex problems within a rich environment – such as the software studio. The studio also enables each student group to develop and succeed according to their needs, and in different ways.

The conclusions above arise from an ethnographic study in an undergraduate software studio prototype with two student groups and their mentors.

Categories and Subject Descriptors

K.3.2 [Computer and Information Science Education]: *Computer science education, Curriculum.*

General Terms

Design, Human Factors

Keywords

software engineering education; design practice; studio-based learning and instruction; ethnography; groups; graduate attributes; professional competencies.

1. INTRODUCTION

Challenges in contemporary Information Technology higher education include: a significant mismatch between what employers perceive as important abilities and how universities prepare graduates for employment, particularly with regard to non-technical skills [16]; and the changing expectations and learning styles of students [1] [16].

One response to these challenges is to use studio-based learning, the classic approach in the creative arts. Over the last twenty years, the studio approach has gained traction in the ICT and Engineering disciplines because it offers a superior learning experience, especially in achieving practical skills [12] [6].

In the undergraduate ICT degrees at our university, one of our aims is to provide an industry-collaborative, reflective software learning and development environment for students [13]. This is in order to re-invigorate software education and make it more effective, vibrant and acceptable to students and industry. This entails moving a substantial portion of the current teaching objectives in some majors of the undergraduate IT and Engineering degrees into a Software Development Studio (SDS). The SDS will start mid-2014.

The SDS will be a component of every year of the degrees, with experiences and responsibilities increasing with each year. Institutionally, the studio will be a component of a course (subject), integrated with lectures, tutorials and laboratories. Educationally, the studio will be based on reflective practice, and developing software as design practice. It will incorporate the ideas of an iterative/agile approach to software development, using industrial tools and with students working in development roles appropriate to their stages of learning and experience. Grounding in professional practice will be provided by industry partners and mentors and industry projects.

Before our new studio-based approach becomes part of the way that software engineering and development is taught in our faculty, it was, and is, important that we explore, design, prototype and evaluate different aspects of the approach. This paper reports upon our prototyping.

1.1 Related Work

Empirical research to date demonstrates that students appreciate studios, and, that the studio's rich environment contributes to graduate attributes/professional competencies. The research methods used have included surveys, diary examination, structured interviews, action research and reflection.

With regard to students' appreciation of studios (where studios are compared to lectures, etc.), Carbone [5], Armarego [1], Hundhausen [15] [14] and Williams [19] all reported that students prefer the studio experience to the lecture/tutorial/laboratory format.

As for graduate attributes/professional competencies, Williams [19] found that "students overwhelmingly enjoy learning from their peers and believe that a collaborative environment better prepares them for the 'real world'."

Carbone [5] states that "Students found the studio precinct an inviting and comfortable place to learn despite some frustrations with IT-related problems. Students' comments indicate the teaching environment facilitated collaboration, and by the end of the year they began to see the course as being better integrated."

Cennamo [7] asserts that "students need to learn to iteratively generate and refine possible solutions to a design problem. Collaboration with others is essential to seeing the design problem in new and different ways, serving to both broaden solution possibilities and assist in idea refinement. Students need learn to communicate clearly by using the conventions of their discipline, in order to convey their design ideas and gain meaningful input from others."

Hundhausen et al's work [15] is notable in that they used what may be called a control group (which stayed in the lecture stream). Their results indicated that students' self-efficacy (empowerment, ability to control their situation), which may be extended to imply life long learning, increased with the studio group, and, *decreased* in the traditional stream group.

Daniels and Cajander have explored the issue of groups and collaboration over a number of years. In [9], they explore collaboration within and without groups, with the outcome of greater understanding between subgroups, and, a realisation by subgroups that greater collaboration would improve their work.

With regard to professional competencies, or graduate attributes, empirical work includes issues for students in evaluating competencies [4]; the issue of staff engagement [10] and the issues of success in developing competencies in students and assessing them [8].

2. RESEARCH APPROACH

Ethnography is seen as a method that allows a broad landscape to be developed, to find the (previously) unseen and unobserved, and may shed light upon the confusion mentioned. "The irony is that good ethnography requires the researcher to pursue the detours, and to become lost in the culture in order to learn the terrain" [11]. Previous empirical research has not used ethnography, and it seems appropriate to add ethnography to the research viewpoints from which to view studios.

Beyond the broad landscape, one of the things that we wanted to explore in the prototype was how the people worked together in a studio-type environment, as it was happening during the time that they were experiencing it. Bull et al, discovered that the studio is not as well defined as those who use the term may hope. "Our 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" [3].

As an experienced ethnographer, the second author was employed to lead the study. She was required to participate in all the weekly studio sessions, by 'being there', the most important aspect of fieldwork [11]. Her non-technical background gave her some distance from the participants (both students and mentors) and the process. This distance facilitated observations and the development of insights that someone more familiar with software development may have missed.

3. FIELDSITE AND SETUP

In the second semester of 2013, we ran a software studio prototype. We did this in an undergraduate second year core project subject, in which students work in assigned groups of 10, each with a project tutor, to design and develop a complete software system from scratch. The subject is regarded as being very challenging, by both students and staff, as it is the students' first full system development experience. The Agile Scrum approach is used and students have to practice version control, configuration management, different levels of testing etc. Prior to this semester, students have only done subjects that each focus on one, discrete aspect of software development, covering programming, requirements modelling, algorithms and data structures, interface design, database design and web systems.

All of the groups were given the same requirements: to develop a system to track feral animals for a state Wildlife and Parks department. The core of the system was a database to store details about sighted feral animals and registered users, with a web-based system to record and search for feral animals in various state parks and display analytics of the sightings. The web system needed to provide for user entry of sightings of particular feral animals, a search function for feral animals and an analytics function to visually display statistical and summary information on feral animals, and user account registration and management, The groups were allowed to choose whichever technologies they preferred – the studio prototype groups both chose to use MySQL and PHP for the development and Atlassian's Jira to manage the project.

Two groups enthusiastically volunteered to participate in the studio prototype, and so we had 20 students and 2 academic mentors (project tutors), one of whom is the first author of this paper.

There was also an industry mentor in the weekly studio sessions. His was a consultative role, and he answered students' queries on development issues as these came up and advised on contemporary development, particularly issues of architecture, scalability and usability. The two groups met together in same room with both academic mentors and the industry mentor once a week for two to three hours.

4. WHAT HAPPENED WAS …

In this section, a narrative of the studio prototype is presented. It is not a complete account of what happened in the studio over the semester; it is a distillation of the ethnographic record that focuses on particular aspects of the story. The events are given in chronological order to give the reader a sense of the temporal nature of the study and, especially, the changes that occurred over the semester. The names used in the narrative are pseudonyms.

4.1 Starting Off

The studio is a large trapezoid-shaped room, with tables with computers upon them, around the outside and in the centre of the room. There are no partitions, but the room shape allows a degree of separation. The two studio prototype groups are referred to as G1 and G2. Their first action was to elect leaders, Stephan (G1) and Neal (G2).

In the first studio session, the students performed an exercise called the Lego Scrum game, which is frequently used to introduce the Agile software development methodology. The students were given minimal instructions on how to go about the task, but they were told what was required: build a mini-town, with various components such as a house, a garage, roads, a sports stadium. There were also some constraints given such as colours, building functions and relative size of components. They were also told that the task had to be completed within a limited time. Each group was given a box of Lego blocks to use.

One of the groups, G1, immediately all sat down in a circle on the floor, and spread all their Lego blocks out in front of them to start. The group was very methodical and cooperative– they broke the tasks up, planned how to go about the building and allocated specific building tasks to each group member. This group finished in good time, with extra components built.

The other group, G2, stood around a table, with the box of Lego on top of it. They seemed to be rather overwhelmed by the assignment and went about the task in some confusion. There was not much discussion, and no planning or task allocation occurred. Two people tried to build the components while another two dug around in the Lego box and passed blocks to the two builders. The other members of the group just stood around and watched. This group was not finished by the end of the allocated time, and seemed very disheartened by their performance and the expectations imposed by the task.

The Lego Game was indicative of the first Sprint planning that the groups are required to do later in the session. G1 talked about the Scrum methodology and how to use it and implement it, as several group members were familiar with the approach and so they could get going with it immediately. This group were very confident in themselves, as well as with their understanding of the development process and how to use it. Most of the members of this group seem to be extrovert, self-confident personalities, and so the discussion was lively and they required little direction or prompting from the mentors.

G2 was a much quieter group whose members appeared more introverted than G1, and not at all confident about what they were required to do or their capacity to tackle the work. The Agile approach was totally new to all of them. This group focused on the technology that they wanted to use for the database, the programming language etc., rather than the Scrum methodology and its use as a process to understand the requirements and develop the software.

In the second week, G1 was still very focused and apparently organised, getting on with the work without much guidance. G2 needed significant direction, obtaining it from both their academic mentor and the industry mentor. Neal, the G2 group leader, was more confident than in the first week and, encouraged by the mentors, the group orientation started to change from listening to the mentors to listening to each other. At one point, Neal said to the group that collaboration is necessary and beneficial, and they should not be concerned… "those who are not strong can get better in the process."

A week later, G1 did not seem to have as much energy and focus as previously – one member, the communications leader, was unusually late, and the other members engaged and disengaged with the discussion at various times, looking at their own screens or moving out of the circle. Later in the session, the group worked together with some concentration on designing the database schema on the whiteboards, in two sub-groups, each on a different part of the schema. Their academic mentor was very pleased with their progress, stating to the ethnographer that they were ''exceeding expectations'' at that stage.

G2 had started to understand, implement and become comfortable with the Scrum process. They spent some time going over the approach again with their academic mentor, and the discussion seemed to be a breakthrough for the group. They started being methodical, with the whole group working out together which tasks they needed to do in the next Sprint and allocating them to specific group members. They used the class time to plan and organise the work, rather than to design or work on their software system They spent time outside class once or twice each week, to work together for a couple of hours.

Apart from the obvious use of technology in class, social media, especially Facebook, was used extensively by both groups to communicate about their work and the project.

4.2 Sprints and Leadership

A month into the semester, G1 still seemed more cohesive than G2, and indeed somewhat competitive with regard to G2. Both groups seemed to be in a similar place and doing much the same tasks in the studio session.

G2 could see "results" in terms of completing their allocated tasks and working as a group. There was a sense in the group that they might finally be on track and were able to envision the project in its various stages. Completing their first Sprint had given them momentum and a project plan through which to consider the next set of tasks.

The organisational tool of the Scrum Sprint seemed to helped both groups create a project plan. It also seemed to help Neal find his feet as the G2 group leader and as the coordinator of the Sprints and Backlogs week to week. His style was more understated than that of the other group leader, but he seemed to be more comfortable and grew into the leader's role with each passing week.

G1, which originally appeared the more cohesive, had its leader changed (to Hardeep) and a week later appeared to be undergoing another leader change. Wayne, one of the leadership contenders, acted as 'discussant' for the group, posing problems and suggesting possible directions for them to go in. [The ethnographer] was no longer sure if Hardeep was still the group leader as Wayne had assumed a central role.

4.3 The Hare and the Tortoise

It was the week before the mid-semester presentations, in which each group in the subject had to present their work to another group and their tutor for a peer review. G1 and G2 practiced their presentations in the studio session.

G1 was not ready at the beginning of the session, so G2 offered to present first. It was obvious that their efforts over the past few weeks had paid off in terms of group cohesion and understanding the Agile process – it had provided them with a structure to work within, and they had become deeply embedded in it and made significant progress. The group appeared fully prepared and acquitted themselves well, even if their presentation was a little unadventurous. As this was a simulation of the formal presentations, G1 left the room for a few minutes to put together some questions to ask the presenters, although 2 of them remained in the back of the studio and worked on their presentation. Neal, the G2 group leader, answered most of the questions. He came across as pleased and confident in answering the questions, and spoke about his group's work in a relaxed manner. He did not appear to be at all phased by any questions and one answer was that, yes, they are following the Scrum methodology very closely.

Overall, G1 seemed somewhat surprised by the quality of the presentation and work they had just seen.

G1 started their presentation in some disarray; it was clear that they were underprepared. Although their presentation perhaps better demonstrated the system requirements and how they responded to these, their slides were not complete, one of their presenters was missing and they finished in a bit of a muddle.

What struck [the ethnographer] most about the response from both groups was a level of generosity – no one was gloating or trying to point score off the other group. There was genuine curiosity and camaraderie on display, even if a sense of competition still hung in the atmosphere.

In discussing their presentation and progress later in the session, G1 decided to appoint a single ScrumMaster (Wayne) for the rest of the semester. Further, their mentor suggested that they needed to improve the level of communications within the group. One of the members, who was very tired, said to the ethnographer just before the session that working in groups was demanding, especially when most of their interactions were on Skype and via GoogleDocs. This was also when G1 found out that G2 had been meeting face-to-face every week outside class and decided to do the same.

4.4 The Invisible Boundary

It was mid-October. Both groups appeared relaxed. They knew they were nearing the end of the semester and had achieved goals in relation to setting up their systems. At this session they reported on the past weeks' achievements and plan for the final Sprint.

The mentors gave feedback to all the students together on the formal mid-semester presentations and discussion ensued. The G2 leader raised a problem that his group had with the presentation peer review. The reviewing group (non-studio group) marked very hard, whilst his group took a more measured and generous approach with the group that it reviewed. Perhaps this reflects the co-operative atmosphere and ethos that had been established in the studio setting over many weeks.

After more feedback and discussion in each group separately with their mentors about their presentations, and, the mentors' mid-semester assessment of their respective groups' work and progress, the groups worked on closing their previous Sprints and planning the final Sprints. At some point, Neal crossed over an invisible boundary and sat talking with members of G1, on G1's side of the room. This was the first time [the ethnographer had] seen him do this and it seemed like breaking a habitual, spatial division between the two groups, and, evidence of inter-group camaraderie, collaboration, and cooperation.

4.5 Doing Time

In the very last studio session for the semester, in the week after the final presentations to other groups in the subject, each studio group gave a detailed demonstration of their system to the rest of the studio - the first time everyone had seen 'step through' demonstrations of the entire system. Although the mentors asked probing questions about the system design, particularly with regard to the user interface and data management, the atmosphere was informal and relaxed, with some gentle teasing and laughter amongst all the students and the presenters from both groups.

After the two demonstrations, there was a class discussion about the subject as a whole, and more specifically, about the studio prototype – what the students found particularly helpful about the environment and approach and what was not very useful. One of the themes that came through very clearly was that the students felt strongly that they wanted more time working together in the studio, not just the 2-3 hours per week timetabled for the subject. Both groups of students had gradually learnt that working together face-to-face regularly and for extended periods was far more effective and productive than trying to get tasks done individually in their own time and then integrating these in class time.

5. FINDINGS

Two findings from this study are highlighted here. The first is the insight into group relations, both intra- and inter-group, and the changes in behaviour within and between the groups over the semester. The second is the holistic nature of the learning experience in the studio, which we call 'things coming together'.

5.1 Group Relations

Looking over the semester as a whole, some of the most significant changes occurred at a group level. We discuss these developments from two perspectives: within each of the groups, and between the groups.

5.1.1 Intra-Group Relations

Initial impressions suggested that G1 was cohesive, competent and collaborative from the beginning of the semester. In the Lego exercise, G2 showed little sense of how to organise, delegate, decision-make – in short, collaborate – in order to complete a task within a reasonable time. These early group impressions frame the perception, by the students themselves and the mentors, of each group in the coming weeks.

To start with, G2 had very little confidence, in their knowledge, skills or capacity to successfully deal with the challenges given by the project. Their inclination to focus on tools and technical details in the first two weeks may have been because they were unsure about the Scrum process and how to use it effectively. For the first couple of weeks, there was not much interaction amongst the group members in G2. Discussions were often oriented towards the mentors instead of each other. This changed as their knowledge, skills and confidence grew. Regular group meetings, in the studio and outside class to work together seems to have enabled the group to build relationships between the members, work things out more as a group, and to become familiar and comfortable with use of process. By the end of the semester they seemed to have achieved a harmonious yet determined ethos as a group. G2 cohered slowly and steadily over time – with the group leader growing into his leadership role. There are a few confident personalities in this group but they were not dominating, and none of them seemed interested in being the group leader at any stage. Once they started to gain an understanding of the methodology, this group stuck very closely to the Scrum approach and this helped them allocate roles, clarify workloads and establish group interactions that appeared to be equitable and inclusive.

G1 came across as super-confident, with a clear idea from the beginning of how to tackle the development challenges as a group. G1 had several members who were forthright, confident and willing to take on leadership positions. One individual worked as a de facto leader at various times in his role as communications leader. The group focused on system development, rather than the intricacies of implementing the process. From the start, they developed solutions quickly, individual tasks were allocated without much debate or coercion and these were completed relatively quickly. It seems, however, that their self-assurance and fast-paced approach meant that they were not as thorough as they might have been in understanding the requirements or considering alternate design solutions. Their mentor spent much of the studio time playing 'devil's advocate' and asking 'what if' questions, subtly reining them in to encourage them not to rush into decisions.

Further, ambivalence about and changes in the group leader role in the first half of the semester proved disruptive to their group functioning. It was also not clear how less self-assured members may have coped initially in a group where 5 (half) of the members had leadership personalities. The group had difficulty allocating roles, as well as ongoing concern over a group member who was not pulling his weight.

The wake-up call about their relative progress at the mock mid-semester presentations encouraged them to reflect on the way they were going about the work as a group. As a direct result of this, the group reorganised itself, allocated a ScrumMaster role to one specific person for the rest of the semester, separate to that of group leader, and started meeting and working together face-to-face regularly outside of class time. These strategies helped the group stabilise and smoothed their progress over the second part of the semester. They still appeared driven to succeed, but were less sure of themselves and perhaps more realistic.

Collaborative learning within each group

One of the most significant characteristics of a studio environment is collaborative learning, with students working out how to do things and to develop their own skills by learning together and from each other. This was very evident in the prototype.

The willingness of G2 to work slowly, methodically through the Scrum methodology and acquire confidence and competence in its use appeared to give them a solid foundation and enabled steady progress in system development. A sound, thorough understanding of the requirements, the Scrum process and various development tools were gained by a 'learning by doing' approach, More than this, together, the group had to figure out what they needed to know at each stage and how to use it effectively.

In G1, the group learnt about the Scrum approach from the three group members who already had experience in it, which meant the group got going quickly implementing a development process, and the other group members built up their own knowledge about Scrum from others' experience. Whatever else happened, it is clear that each group learnt how to operate effectively, but in very different ways; neither group's learning was better, just different.

5.1.2 Inter-Group Relations

The perception of how each group responded to the Lego exercise may have been a distraction. It set the groups up in a way that confirmed G1's view of itself as having the capacity, and then some, to cope with the semester project, which encouraged them to be somewhat gung-ho in their approach. G2 was left feeling overwhelmed and convinced that they did not know enough to deal with the challenges posed in the subject.

Initially there was some sense of rivalry and competition – particularly from G1, whose own self-perception was one of "having the edge" on G2.

A turning point in terms of shifting perceptions of both groups and the dynamic between them emerged during the mid-semester mock presentations. G1 appeared surprised by the standard of work done by and confidence of G2, leading to a re-assessment of their own work practices. This event changed the dynamic and the "assumed order of things" between both groups.

So, the interactions between the groups within the studio sessions effected changes in the way G1 went about their work and managed their intra-group relations, learning from what strategies they deemed successful for G2.

Collaborative learning across/between the groups

The co-operative atmosphere during the mock presentations was impressive. Each group appeared to be genuinely interested in the work of the other and afterwards this continued to permeate the environment.

Attitudes between the two groups changed, over the first half of the semester particularly, from competitive to a strong sense of camaraderie between the groups and across the studio as a whole, where the groups supported and encouraged one another.

Moreover, there was clear evidence of one group learning from the other when G1 made changes to their strategy as a direct consequence of finding out what had been effective for G2.

This section is about how the 2 groups worked and changed over the semester, but it is very important to recognise that this is not a comparison, or value judgement of the two groups' characters, behaviour and performance. It is simply a statement about how different groups function and flourish in different ways. Both groups ended up in a similar place – working together competently and effectively as a group, having designed and produced a good quality software system. But, the learning experiences and the changes that occurred in each group were very different.

This is a very significant finding in terms of the studio approach to learning software development. The studio environment allows each group to evolve, as they need to, given their specific member mix, with various backgrounds and experiences and preferred ways of working. This is in contrast to traditional teaching according to a syllabus, covering each topic regardless of where students are or what is their prior knowledge. Learning both technical skills and more on an as-needed basis throughout the project development process was made possible by the studio environment.

Much of what we learnt about the group relations, the behaviour within each of the groups and their interactions across groups can be considered to lead to the second finding. The changes in the group relations did not happen in a vacuum, but as part of the studio prototype experience as a whole. The import of this holistic experience is addressed in the following section.

5.2 Things Coming Together

One way of looking at the studio prototype is as a network which incorporates people, software tools, subject policies and procedures, a development methodology, processes, techniques, documents, practices and products [17][18]. This network is not static, nor is it pre-configured or already there. Rather, the relational and emerging nature of this type of network means that it is continuously and dynamically reconfigured over time. Star calls this network a web; she values "the ways in which knowledge is co-created by a web of people, symbols, machines and things" ([18] p.405).

So, we could consider the 'tangible' elements of this (studio) network to be the two groups' members, the three mentors, the system specification/requirements, the Scrum methodology, the development tools, the project management tools and communication tools. The 'intangible' elements include the intra-group relations and interactions, the inter-group relations, the relations and interactions of the mentors with the groups and individual members, and the relations of the groups and their members with the technical tangibles. All of these elements interconnect; dynamically providing a network or web in which software development knowledge and skills are co-created.

Perhaps the most challenging aspect of the project for the students was integrating their technical know-how, decision-making, and communications–verbal and written–at a systems level. Instead of being able to focus on a discrete thing such as a single program or a relatively simple data model, the students were required to deal with all the different aspects of the development experience at once. Although at specific times they could focus on a particular task or element, much of the time, their efforts were aimed at integrating things and working in a complex context.

The Lego exercise was something that needed to be done immediately, within a very limited time, and it was a contained problem, with very clear requirements and tasks. Very much like an assignment in a 'normal' subject. Using a methodology such as Scrum over a significant period of time to develop the effective use of a process by the whole group to design and build a particular software product of good quality was an entirely different proposition. Group functioning and relations had to be managed more thoughtfully, strategically and contingently. Ultimately, both groups were successful, but they went about the project in different ways and, notably, from different starting contexts.

Another significant factor is time – time for students to engage with a complex problem, time to build the group, time to build an effective solution. Time is the catalyst within a rich learning environment that allows students to immerse themselves and to deeply learn [2] as things come together.

6. CONCLUSION

In conclusion, we see evidence in this study that the studio provides a learning experience that genuinely prepares students for professional practice. Learning that entails dealing with complex technical problems and tools. Learning that involves working effectively in groups. Learning that results in the building of students' self-confidence and -conviction that they can successfully deal with the challenges of modern software system development. Learning that allows the accomplishment of the more elusive professional competencies

7. ACKNOWLEDGMENTS

Our sincere thanks to the studio prototype mentors and students and the subject co-ordinator. This research was made possible by Faculty and University teaching and learning grants.

8. REFERENCES

[1] J. Armarego and L. Fowler, Orienting students to studio learning, in *Proceedings of the 2005 ASEE/AAEE 4th Global Colloquium on Engineering Education*. Australasian Association for Engineering Education, 2005.

[2] J. Biggs and C. Tang, *Teaching for Quality Learning at University*, Fourth Edition. McGrawHill. 2011.

[3] C. Bull, J. Whittle and L. Cruickshank, Studios in software engineering education: towards an evaluable model, in *Proceedings of the 35th International Conference on Software Engineering (ICSE13),* IEEE, 2013.

[4] Å Cajander, M Daniels and B von Konsky, Development of professional competencies in engineering education, in *Proceedings of the 41st ASEE/IEEE Frontiers in Education Conference*, IEEE, 2011.

[5] A. Carbone and J. Sheard, A studio-based teaching and learning model in IT: what do first year students think?, in *Proceedings of the 7th annual conference on Innovation and Technology in Computer Science Education*, 2002, pp. 213-217.

[6] A. Carter and C. Hundhausen, A review of studio-based learning in computer science, *Journal of Circuits, Systems, and Computer* (JCSC 27), October, 2011.

[7] K Cennamo, C Brandt, B Scott, S Douglas and M McGrath, Managing the complexity of design problems through studio-based learning, *Interdisciplinary Journal of Problem-based Learning*, vol. 5, Issue 2, 2011.

[8] M.Daniels, Developing and assessing professional competencies, *Acta Universitatis Upsaliensis, Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 808*, 2011.

[9] M.Daniels and Å. Cajander, Experiences from using constructive controversy in an open-ended group project, in *Proceedings of the 40th ASEE/IEEE Frontiers in Education Conference,* IEEE, 2010.

[10] B. de la Harpe and A. Radloff, Developing graduate attributes for lifelong learning – how far have we got?, *Lifelong Learning Conference*, 2008.

[11] D. M. Fetterman, *Ethnography: Step-by-Step*. Third ed. Applied social research methods series, Vol. 17. Sage, 2010.

[12] D. Garlan, D. P. Gluch and J. E. Tomayko, Agents of change: educating software engineering leaders, *IEEE Computer*, November, 1997.

[13] O. Hazzan, The reflective practitioner perspective in software engineering education, *The Journal of Systems and Software*, 63, 2002.

[14] C. Hundhausen, A. Agrawal, D. Fairbrother and M. Trevisan, Integrating pedagogical code reviews into a CS 1 course: an empirical study, *SIGCSE'09*, Chattanooga, Tennessee, USA, 2009.

[15] C. Hundhausen, A. Agrawal, D. Fairbrother and M. Trevisan, Does studio-based instruction work in CS 1? An empirical comparison with a traditional approach, *SIGCSE '10*, Milwaukee, Wisconsin, USA, 2010.

[16] T. Koppi, P. Ogunbona, J. Armarego, P. Bailes, P. Hyland, T. McGill, F. Naghdy, G. Naghdy, C. Pilgrim and M. Roberts, Addressing ICT curriculum recommendations from surveys of academics, workplace graduates and employers, *OLT Final Report*. Wollongong: University of Wollongong, 2013.

[17] L. Suchman. *Human-Machine Reconfigurations: Plans and Situated Actions*, 2nd edition. Cambridge University Press, 2007.

[18] S. L. Star. The trojan door: organizations, work and the "open black box". *Systems Practice*, 5(4), 1992.

[19] L. Williams, L. Layman, K. M. Slaten, S. B. Berenson and C. Seaman, On the impact of a collaborative pedagogy on African American millennial students in software engineering, in *Proceedings of 29th International Conference on Software Engineering (ICSE'07)*, 2007.