DOI: 10.33965/celda2019_201911L037

SOFTWARE FACTORY PROJECT FOR ENHANCEMENT OF STUDENT EXPERIENTIAL LEARNING

Muhammad Ovais Ahmad[1] and Kari Liukkunen[2]
[1]*M3S Research Unit, University of Oulu, Finland*
Department of Mathematics and Computer Science, Karlstad University, Sweden
Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Poland
[2]*M3S Research Unit, University of Oulu, Finland*

ABSTRACT

Providing opportunities for students to work on real-world software development projects for real customers is critical to prepare students for the IT industry. Such projects help students to understand what they will face in the industry and experience real customer interaction and challenges in collaborative work. To provide this opportunity in an academic environment and enhance the learning and multicultural teamwork experience, the University of Oulu, Finland offers the software factory (SWF) project. This paper presents the design of the SWF course and the learning environment and assessment techniques, and it discusses the importance of reflective learning diaries and serious games. Additionally, this paper examines factors in the SWF learning environment that affect student learning in the SWF course. Survey data were collected from the last six years of SWF projects. The results show that students consider the SWF to be a good collaborative learning environment that helps them achieve academic triumphs and enhances various professional skills. The learning diaries are effective for increasing students' learning experiences as well as providing an opportunity for teaching staff to monitor students' progress and offer better facilitation. These results are helpful for academic institutions and industry when developing such a learning environment.

KEYWORDS

Collaborative Learning, Project-Based Learning, Software Engineering Curricula, Teamwork, Software Engineering

1. INTRODUCTION

In the last two decades, there has been a significant shift in the information and communication technology (ICT) industry, as highlighted by Marc Andreessen in his essay *Why Software is Eating the World* (Andreessen, 2011). This shift has also pushed educational institutions to train graduates with various skills and competencies. This challenges universities to create software engineering (SE) technology agnostic courses that prepare graduates for the real-world software industry and fulfill industry needs and demand. One way to train SE graduates is to provide project-based learning through capstone courses (Howe, 2010; Walker, 2015). Educational institutions must train students in real-life practical projects where students engage in a collaborative teamwork environment and develop project management skills (Palacin-Silva et al. 2017). Universities around the world already include such capstone courses based on ACM curriculum guidelines (ACM Joint Task Force, 2014). However, various transversal capabilities—leadership, decision making, negotiation, self-reflection, and the infusion of design thinking—receive little or no attention in these courses. To provide such competencies, the University of Oulu established the software factory (SWF) learning environment/laboratory in 2012. The Oulu SWF is an infrastructure platform that serves multiple purposes to support SE research, education, and entrepreneurship.

The SWF is a test bed for SE ideas and a source for original research on the development of basic scientific software (Ahmad et al., 2014). The SWF is an educational vehicle for the university where the artifacts produced in the factory serve to improve learning and provide teaching materials in close collaboration with industry (Ahmad et al., 2014). The Oulu SWF is part of a European Union SWF network (Taibi et al., 2016). The aim of the SWF is for students to share their experiences, learn in a collaborative environment, and grow to compete in the fast-growing ICT domain. The Oulu SWF laboratory is equipped with 17 latest computers, software development tools, a wide range of gadgets for students' i.e. smart phones, tablets, cameras etc.)

This paper describes the SWF course design, assessment techniques, student perception, documentation of teaching experiences, and it discusses the importance of reflective learning diaries. The aim is to identify factors in the SWF laboratory that affect learning, in terms of exploring i) student achievements in term of skills gained, ii) students' perceptions of the SWF learning environment, as measured with the computer laboratory environment inventory (CLEI) (Newby & Fisher, 1997), and iii) students' attitudes toward the SWF project course, as measured with the attitude toward computers and computing courses questionnaire (ACCC) (Newby & Fisher, 1997).

The paper is structured as follows. Section 2 sheds light on related work, and Section 3 discusses the SWF course under the pedagogic lens and presents the SWF course learning objectives, mode of delivery, overall structure, student team formation, mentoring, and student assessment. Section 4 reports students' perceptions, and Section 5 discusses lessons learned from the teacher perspective in the context of teaching and managing such courses. Finally, Section V concludes the paper and sheds light on future research work and improvement to the SWF and similar courses.

2. RELATED WORK

The ACM curriculum guidelines recommend offering project- and problem-based courses to students that prepare them for real work in the industry (ACM Joint Task Force, 2014; Palacin-Silva et al., 2017). It is essential for SE students to have hands-on experience and a glimpse of real software industry work during their education. To prepare students for a profession, universities around the world offer various capstone courses. The concept of the SWF project is also based on capstone course concept.

Since 2010, the SWF and SWF-based courses have been offered at various universities around the world, i.e. University of Helsinki, the University of Oulu, the University of Eastern Finland, the Free University of Bozen-Bolzano, Tampere University of Technology, the Free University of Cagliari, the Technical University of Madrid, Montana State University, the Catholic University of America, and Bowling Green State University (Ahmad et al., 2014; Taibi et al., 2016; Tvedt, et al. 2002). These SWFs aim to provide students with practical experience in software development projects and help the students to gain business experience in a collaborative environment, as well as polish their technical expertise. However, research on SWF projects and course curricula is scarce. Most studies report success stories, students experimenting with processes, and positive experiences of students' motivation in such courses or projects. We did not find any study that reported on SWF course design, mode of delivery nor detailed of students' assessment techniques.

3. SOFTWARE FACTORY PROJECT DESIGN AND EXECUTION

Since 2012, the SWF laboratory has offered a 10 ECTS (290 h of work) advanced-level course for Information Processing Science master's degree program students at the University of Oulu, Finland. The purpose is to expose students to real-life software development projects in a multicultural collaborative environment. The focus is learning by doing—that is, managing authentic, resource-limited project work and integrating the practices of an academic expert in a unique project assigned by a software company. The SWF course is offered every year in the spring semester to 15–20 students.

3.1 SWF under a Pedagogic Lens

When designing a SWF course, various theories or approaches to learning—behaviorism, cognitivism, and constructivism—are taken into consideration (Anderson, 2008). The SWF adopted a blended learning approach to maximize students' learning outcomes as follows:

- From the behaviorism school, we observed how teaching staff behavior affects students' learning, e.g., teacher approval of certain items required by the course. In such cases, the teacher acts proactively to respond quickly. In this way, we avoid unnecessary wait times from the students' perspective.
- Adopting cognitivism approach, the students are encouraged to have a mental map of their project and processes. Such encouragement is important, especially in the context of software development. The students need to have a map for a specific goal, which boosts motivation and reduces stress.
- As constructivists, we do not push students to memorize the concepts taught during the lectures. Various serious games, learning diaries, and discussions enable students to develop their knowledge.

In general, the SWF course is more inclined toward constructivism due to a student-centered model that focuses on learning by doing in a collaborative environment and problem-based learning (Gokhale, 1995). Such a collaborative course and environment have a significant impact on learning (Ahmad et al., 2014; Gokhale, 1995; Khine & Lourdusamy, 2003; Taibi et al., 2016).

3.2 Project Course Learning Outcomes

After completing the course, the students should demonstrate their ability to work on a challenging ICT project. Students learn to acquire and apply professional expertise in the topic of the project. One example of a project is a path finder using the Robot Operating System for an autonomous electric car. Students should be able to:

- Act as independent professional members in an ICT project; a team member collectively produces, monitors, and updates the project plan (a project with a fixed deadline and human resources).
- Search up-to-date scientific literature on the subject matter of the project to build professional expertise in the topic and apply this to the project work.
- Develop analytical and creative skills for successful completion of the project and monitor and communicate the status (time and human resources used) of the project in real time within the team.
- Develop skills to communicate with the customer in a professional context and manage a successful project review with the steering group/project team.
- Report and explain the status (progress, results, and future estimations of the project) to the steering group to support decision making and problem resolution concerning the project's future.
- Work as a project team member with people from different technical and/or cultural backgrounds, produce a realistic outcome in relation to the project deadline and human resources (ok, good, or excellent), and reflect on the relationship between the process model(s) selected for the project (evolutionary, agile, lean, etc.) and the management practices followed in the project.

3.3 Mode of Delivery

The SWF course adopted blended teaching or mixed-mode instruction to boost collaborative learning (Martyn, 2003). This approach has become popular in SE because it helps develop critical thinking and improves understanding of various concepts (Gokhale, 1995; Palacin-Silva et al., 2017). There are four major components of the SWF course: classroom lectures, serious games workshops, weekly customer meetings and monthly progress reports, and individual project work. All SWF project–related communication, materials, and deliverables are stored in the Optima workspace[1].

The course provides traditional introductory lectures (4 × 2 h) and two workshops (2 × 8 h), where the steps for carrying out the course are described with relevant information. The students already have a background in project management and software development tools, processes, and techniques. However, in the first three introductory lectures, these concepts are briefly reviewed. The interesting aspect of these lectures is the participation of software professionals from the local software industry. Oulu is a smart city, and many big ICT companies, such as Nokia, Ericsson, and Bittium, have offices in the city. The invited professionals share their experiences and provide insights into running projects efficiently and adopting a software development and management method or technique based on project needs.

Students are divided into multiple groups during the fourth lecture; each team has three to four members. Additionally, interested software companies are invited to present the project ideas to the student teams.

[1] Optima is a learning management system used at the University of Oulu, Finland. https://www.discendum.com/references/?q=optima

These software companies and entrepreneurs are the real customers of the SWF projects. Such real customer engagement helps students learn more about the dynamics of software projects and enhance their soft skills. The teams select a project based on their interests and present the first draft to teachers and other teams.

The main goal of the workshop is for students to present their initial understanding of the project, problems expected, and possible solutions. The students present their project plan, development process, and management practices. The students break down the customer requirements and needs, as well as discussing the delivery of the project to the customer. In this phase, the teaching staff act as facilitators and guide students to successfully implement the projects. The aim of the workshop is to facilitate students through experiential learning to understand group dynamics and the software development methods, tools, and techniques used in the software industry. Various serious games are played, such as the Marshmallow Challenge, Draw Toast and Scrum Simulation with LEGO. For example, the Marshmallow Challenge is an instructive design exercise that engages students to work in teams. Such activities help students experience the dynamics of collaborative teamwork, the importance of analyzing each other's perspectives, and iteration planning.

The actual software development starts at the end of the first workshop. The students work on their projects for three months. The project team(s) experiment and select development methods, tools, and techniques based on the project requirements and customer demand. Frequent communication inside and outside the team is important. The project team meets the customer based on need, but the course recommends organizing a weekly meeting. Additionally, there are monthly management group meetings, where students present their overall progress, project and team challenges, and the status of the work hours. The students must also write a learning diary once a month. The teaching staff provide feedback in management group meetings.

3.4 Team Formation and Mentoring

We used a team formation tool called CATME (Hrivnak, 2013), which restricts each team to four or five students. The students' characteristics, such as ethnicity, gender, leadership preferences, specific technology skill level, and relevant knowledge, play a critical role in the formation of diverse teams. The teaching staff separately mentor each team and monitor their performance and dynamics. The teaching staff act as facilitators, help students prioritize their tasks, provide feedback on the development process, and discuss the reflective learning diaries to enhance student learning. To track students' progress and facilitate efficiently, teachers use various techniques, such as daily stand-up meetings, agile retrospectives, burn-down charts and Kanban boards (Ahmad et al. 2014; Ahmad et al. 2018).

3.5 Assessment Methods and Criteria

Assessing project work and then grading individual team members is always challenging. It is the teacher's duty to fairly assess each team member and present the criteria clearly and understandably. In the SWF course, we used a rubric-based assessment. It is mandatory for students to attend all the lectures and workshops. The distribution of the SWF course assessment is summarized in Table 1.

Table 1. Software factory (SWF) course evaluation and grading

Criteria	Points	Criteria
Final Software Product (Group Evaluation)	25	Final evaluation by the customer. The product must fulfil the customer requirements and meet functional and non-functional requirements.
Supporting Documents (Group Evaluation)	25	Supporting documents and evidence regarding the entire development process, including planning, management, and the implementation process.
Supervisor Evaluation (Group Evaluation)	10	Supervisor's evaluation of group performance.
Reflective Biweekly Learning Diary (Individual Evaluation)	20	Three reflective reports (3 × 5 points = 15). Final lesson learned report (5 points).
Peer Evaluation (Individual Evaluation)	20	Peer evaluation of individual performance by other members of the team.
Total	100	Final grade.

The customer evaluates the student project based on the group's communication with the customer, meeting deadlines, the final product, and whether customer got what he or she requested from each team. The customer also considers how each student team member approached them throughout the project.

The students are required to prepare a project plan, which is a live document until the end of the project. The project plan is updated with each sprint, which lasts for two weeks. The students prepare a report at the end of the project, in which the project team describes planning, managing, and implementing the process and the planned and actual work hours. The project report describes the product features and the resources used.

The teaching staff observe each team's progress from the beginning until the final presentation of the project. The teacher provides feedback after every weekly customer meeting and guides the students to improve their preparation for the next steps'. The teachers also consider how the teams prepare their presentations, handle technology during the meetings, and work together internally.

It is mandatory for every student to write four diary entries during the project. Reflective diaries are core elements of self-regulated learning that promote the development of metacognitive strategies (Fulwiler, 1986). Fulwiler described the rationale for introducing reflective diaries: "In the academic world, where we teach students to gain most of their information from reading and listening, we spend too much time telling our students how to see or doing it for them. That's not how I would encourage critical, creative, or independent thinking. Our students have good eyes; let's give them new tools for seeing better: journal writing is, of course, one of those tools."

Peer assessment is a powerful meta-cognitive tool, which has been advocated in various studies (Kaufman, Felder, & Fuller, 2000; Layton & Ohland, 2001; McGourty, Dominick, & Reilly, 1998). According to McGourty et al. (1998): "[i]n a cooperative learning environment, students themselves are often in the best position to provide one another with meaningful feedback regarding both their technical and interpersonal performance." A number of peer assessment tools and advised reducing the possibility of a student intentionally "damaging" his or her peers' scores and ensuring that students do not feel that they are "ratting" on their peers (McGourty et al., 1998; Nicole, Pamela, & Rebecca, 2005). In the SWF course, students are required to fill out a form to report aspects of their team members' contribution and behavior characteristics. The Oulu SWF project peer assessment is based on Sanders (Sanders, Dean, Sanders, & Dean, 1984). Examples of teamwork characteristic statements include attending team meetings, contributing to the discussion at the meetings, completing tasks on time, and the team member's ability to work with the other team members. The students were asked to score the characteristics using the following scale: Always (2 points), Usually (1), Sometimes (0), Rarely (-1), and Never (-2). Furthermore, students have the opportunity to express their feedback in answer to an open-ended question and report their concerns in detail.

4. STUDENT PERCEPTION SURVEY RESEARCH STRATEGY

The students' perceptions of the SWF course and facilities are based on the CLEI and the ACCC (Newby & Fisher, 1997)(see Table 2). Studying the learning environment (i.e. the SWF laboratory) is one way to explore student perception (Kolb & Kolb, 2005; Newby & Fisher, 1997). Newby and Fisher (Newby & Fisher, 1997) developed the CLEI to measure students' perceptions of their learning environment. The CLEI has five constructs: *student cohesion, open-endedness, integration, technology adequacy, and laboratory availability*. In the Oulu SWF context, the laboratory availability construct is not relevant, as the SWF laboratory is assigned to SWF project students 24/7. Furthermore, with the help of the ACCC (Newby & Fisher, 1997), students' attitudes toward computers and computing courses are assessed. The ACCC consists of four constructs: *anxiety, enjoyment, usefulness of computers, and usefulness of course*. In the Oulu SWF context, we removed the usefulness of computers construct because the targeted students are in the final year of the Information Processing Science master's degree program and considered competent in the use of computers.

The teaching staff conduct a voluntary online survey annually at the end of the SWF course from last six years (2012–2018). In six years, 50 of 90 students participated in the survey. The survey first part collect student background information. The second part of survey has variables from CLEI and ACCC which are measured using a five-point Likert-type scale, ranging from 1 (strongly disagree) to 5 (strongly agree). Third part of survey cover competencies gained in the SWF project. Further, in open-ended questions option is provided so that student freely express their SWF experience. The data analysis was conducted through descriptive statistics. Before the analysis, the reliability of the factor construct measurements were analyzed with Cronbach's alpha.

The research strategy also include the students' reflection diaries analysis and teachers' perspective. It is mandatory for each student to write minimum four reflection diaries during SWF project journey. The teaching staff explained their own perspective in light of six years of running SWF projects.

5. STUDENT PERCEPTION SURVEY RESULTS

The Table 2 shows that the reliability of the factor measurement is high; Cronbach's alpha varies between 0.597 and 0.951. These values show that the CLEI and ACCC constructs are internally consistent and reliably measured. Students' perceptions of the SWF laboratory learning environment are measured using the CLEI; their perception is quite positive. Table 2, shows that the highest mean scores are for "student cohesion" and "open-endedness" (mean = 4.24 and 3.84, respectively). "Technology adequacy" had the lowest mean score (mean = 3.28). The students feel confident and support each other in their project work. The students also have a positive response for "open-endedness" (average mean = 3.84), which might be due to the close industrial collaboration. The SWF project allows students to choose and experiment with software development processes and obtain feedback directly from the real software industry. The students collectively work toward the same goal and seek a solution for the given problem. The students are encouraged to use and put their theoretical knowledge into practice in the SWF project. The teaching staff facilitate during the project development. Additionally, the SWF laboratory is fully equipped with the latest technology, which is very important for executing such student projects. For example, the students have access to the latest computers, various types of tablets, smartphones, and smartwatches. This access is why students provide positive responses to the "technology adequacy" construct (mean = 3.28). This positive perception shows that CLEI constructs play an important role in student learning. The existing literature argues that there is a positive association between environmental and such attitudinal constructs (Al-Qahtani, 2012; Saadon & Liong, 2012).

Table 2. Constructs of the student perception survey

Measures	Constructs	M	Std. Deviation	Alpha
Attitude Toward Computers and Computing Courses Questionnaire (ACCC)	Anxiety	2.64	.83	.674
	Enjoyment	3.21	.55	.597
	Usefulness of Course	3.53	.72	.812
Computer Laboratory Environment Inventory (CLEI)	Student Cohesion	4.24	.63	.846
	Open-endedness	3.84	.75	.694
	Integration	3.41	.61	.527
	Technology Adequacy	3.28	1.07	.951

Survey questions use a five-point Likert scale, ranging from 1 (strongly disagree) to 5 (strongly agree).

Student attitudes toward the SWF course are quite positive, as shown in Table 2, the ACCC constructs. The constructs "usefulness of course" and "enjoyment" score the highest (mean = 3.53 and 3.21, respectively). The students found the course useful because they work with a real software company project, develop the product or services based on customers' requirements, and manage and monitor their activities. Furthermore, the usefulness construct shows that the students experience first-hand encounters with a real customer, which helps the students to learn negotiation skills and develop for future jobs. The "anxiety" construct received the lowest mean score (mean = 2.64), which indicates that the SWF course did not bore them, make them nervous, or create bad experiences from the beginning to the end of the project work. Such positive student perception is also reflected in the "enjoyment" construct, which received a mean score of 3.21 on the ACCC scale. Student anxiety can be explained with the students' own words in written responses to an open-ended question:

- *"Groups* [one project assigned to one team] *have huge gaps and differences because some students are very modest, and they felt shy to say that they are good at programming, and it is also hard to say and to evaluate if one person is good at programming. While the course really works, and I learned so many things from this course, and this is more like a practical course."*

- *"Confusion and sense of competition kept me on my toes for the whole length of the project. I like the course concept, but the variety of cultures among students brings variation in multiple aspects of how the projects flow. After understanding how human the students all are, I was able to forgive and work in a way I felt comfortable with."*
- *"Working in a group where people do not listen or understand what you are talking about when discussing web architectures, object-oriented programming makes working in the project really depressing."*
- *"When dividing the team, it is better to do it based on the required technical skills distinction, rather than culture differences. It could involve more interaction of the other teams and customers."*

In a nutshell, the students work collaboratively on the project, which helps them gain or improve various competencies. Students help each other and discuss ideas throughout the project work, which results in building a network that aims for better outcomes of their project. The project also helps students to make new friends and form supportive networks in pursuit of improved outcomes. These results correspond to previous findings (Saadon & Liong, 2012).

5.1 Competencies Gained in SWF

Table 3 shows the competencies gained in the SWF project. The students rate themselves highly against the achievement of various competencies gained during the SWF project. Building positive relationships with multicultural team members received the highest value (M = 4.00), which contributes to developing a shared mental model, managing tasks effectively, solving complex problem, better negotiating inside & outside group.

Table 3. Competencies Gained During the SWF Project Course

Competencies	Strongly disagree	Disagree	Neutral	Agree	Strongly agree	Mean
Effective task management	1	3	7	30	9	3.86
Solving complex problems	1	2	9	29	9	3.86
Usefulness of course	1	5	13	17	14	3.76
Sharing responsibilities	1	5	13	17	14	3.80
Developing a shared group vision	1	2	15	20	12	3.80
Building positive relationships	3	1	6	21	19	4.00
Negotiating with other groups	2	1	10	21	16	3.96
Use of rational argument to persuade others	4	2	13	21	16	3.62
Resolving conflict	4	2	16	17	11	3.58

These positive experiences and competencies gained during the project are also expressed in various students' responses to an open-ended question, as follows:

- *"The* [SWF] *project allowed me to develop my project management skills as a whole, from planning to scheduling, task allocation to having formal meetings, and collaborating with a steering group. I gained a good insight into what duties a project manager has and what kind of personal traits are needed for the successful management of a project."*
- *"I learned about interesting technologies and software tools in usability activities in practice."*
- *"Management has also increased my ability to deal with the different nature of people, how to motivate them and drive or steer them to do the work. Everyone has his own working style to do the work."*
- *"From this course, I found my direction for future work, and I found things that I like to do. Thanks to the teachers, customers, and also my group members!"*

An open SWF environment that emphasizes collaborative work enables students to help each other and generate and discuss ideas throughout the project work. Such a positive environment and experiences helps students to build a network that aims for better outcomes of their project journey. These results are also supported by previous studies. Overall, with these competencies, the students clearly worked collaboratively, which helped them to generate ideas, share their views with each other, and solve complex problems (65%). The benefits of students working collaboratively include opportunities to make new friends and form supportive networks in pursuit of improved outcomes (Saadon & Liong, 2012). These results correspond to previous findings (Baloche, 1994; Jane Burdett, 2003).

6. REFLECTIVE LEARNING DIARY

The reflective diary is a facilitation method, as well as an opportunity for students to reflect on their experiences and practices. Such reflective diaries enable students to see how they can better prepare themselves for the challenging SE profession. Writing reflective diaries is the core element in many medical, nursing and teacher education programs (Tang, 2002). The students write biweekly learning diary entries at the completion of each sprint that answer the main questions (What tasks did you do in this sprint? As a learner, what did you improve or learn compared to the last sprint? What were the issues and challenges you faced in this sprint?).

We expected students to reflect on their individual and group experiences for each sprint. In this way, they can identify their own learning, polish existing skills, and seek improvement opportunities. The teaching staff can also see in which direction the student is heading and make necessary adjustments to assist them. These learning diaries also help students to adapt to individual project needs by understanding software development methods, practices, and tools and their application during software development projects and experiments. We, as teaching staff, observed that in learning diaries, it is evident that students reflect on each sprint practice, seek improvement, and adapt development practices based on their experiences. The following are example statements from the students' learning diaries:

- *"We use Trello* [online tool; www.trello.com] *for task management like Kanban board; our understanding of the Kanban process was minimal. Moving our process to the white board in the Software Factory Lab, we had a deeper course to interact with one another, gather feedback from the supervising teacher, etc. Petty issues, such as missing work-in-progress limits on our board, were quickly raised by the supervising teacher. We thus had to move our work process from Trello to the white board in the Software Factory Lab just to not to repeat ourselves but use a common board and approach."*
- *"Being a multicultural team, it was hard understanding other people, which affected our work output to an extent. We tried to do the best, however, working schedule of team members was quite problematic. While there is no one to blame, we need to work together, find more consensus, interact on Slack more, be as productive as possible during the few minutes we have together, and persevere in the given task."*
- *"We come from different cultures, different languages, and possess different accents; it was quite difficult for team members to understand each other sometimes during conversations. However, when communicating on Slack platform, these issues were not present. This was the reasons why Slack communication was frequent"*.

In summary, such reflective diaries are recommended to provide opportunities for students to think critically, look back on the learning activities, help identify what they have learned throughout the practical software development project, and learn how to polish the required skills.

7. TEACHERS' PERSPECTIVE

The teaching staff implement the SWF project in relation to the results and adapt teaching techniques to optimize learning outcomes. In this regard, the following lessons were learned:

- The SWF laboratory involves the local software industry for real projects and customers for the university's SE students. However, non-disclosure agreements must usually be signed. This requirement must be communicated very clearly to students to avoid breaches. Similarly, the message should be communicated to the companies that the students might not be aware of the seriousness of confidentiality, and the companies should be careful when assigning confidential tasks.
- The company and real customers are invited to the final project presentation. However, their evaluation should not focus only on the final product. The evaluation criteria should be clearly communicated to these external stakeholders to avoid confusion and promote fair evaluations.
- The SWF project course design is very effective for motivating students and plays an active role throughout the project. For example, the SWF project has an almost zero dropout rate, despite requiring intense work over one semester. This is why it is important to include serious games and consider the gamification approach to improve students' motivation and active participation and increase collaboration (Glover & Glover, 2013; Sheth, Bell, & Kaiser, 2012).

- The SWF project course is also demanding from the teaching staff perspective due to the frequent communication and mentoring. Each SWF project team requires a teaching assistant to provide technical support, monitor their progress continuously, and facilitate throughout the SWF journey. This technique is very effective in a SE project-based environment (Palacin-Silva et al., 2017; Walker, 2015).
- Creating balanced teams is a challenge with multicultural and heterogenous students. An unbalanced team with inadequate skills or cultural conflicts can create difficult situations during the long and intensive work period. The teaching staff must proactively oversee the teams' work and communication.
- Continues student work assessment and providing frequent feedback improve overall student learning and their ability to execute project efficiently.
- Almost all universities around the world have strong policies regarding the installation of computer laboratory software. Such policies affect students when they need to urgently install software packages. The solution is to install a virtual machine on all students' laboratory computers. This enables students to freely install and update the required software, applications, and tools.

8. CONCLUSION

This paper describes the design of a graduate SE course that uses real projects from software companies with real customers in an SWF laboratory. We documented six teaching experiences, students' perceptions of the SWF laboratory, and the SWF course. Furthermore, this paper provides insight into course delivery, course assessment, peer evaluation, and the use of tool support for forming teams.

The SWF laboratory is an innovative learning environment that offers a graduate-level project-based course. The aim is to bring balance to two aspects: SE processes and building final software development products or solutions. It is very important to maintain a balance because students find it easy to focus on the coding and ignore development processes. Student experimentation with the SE process should be encouraged. Learning and mastering these processes are essential to compete in the competitive job market. The student survey results show that the majority expressed a positive view in two ways: (i) The SWF course is appreciated as an important course in their master's degree curriculum. The SWF project is a good blend of theoretical and practical training that enhances students' enjoyment, and they find the course useful for achieving the required competencies for future jobs; (ii) The SWF laboratory makes it possible to provide opportunities for students to interact with real software industry customers and work collaboratively in a multicultural environment.

Fair assessment is a very important and complex activity for teaching staff in such courses. We developed a matrix that considers various aspects of teamwork. In this way, free riders and hardworking students in the project can be identified easily and treated fairly. In this regard, peer assessment and individual reflective learning diaries play an important role in encouraging and motivating students in this collaborate course. The learning diaries are a good way to promote student engagement and assess their epistemological beliefs and conceptions of learning. This also helps the teaching staff create and update strategies for monitoring and regulating learning. In future work, the SWF course could be run in a geographically distributed context in cooperation with other SWFs and/or other universities. However, this might require much greater technical competency among teaching support staff and strong collaboration between universities. It would be interesting to investigate such courses in different academic and cultural settings.

ACKNOWLEDGEMENT

This research is supported by ITEA3 project VISDOM funded by Business Finland.

REFERENCES

ACM Joint Task Force. (2014). Software Engineering 2014 - Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Retrieved from https://www.acm.org/binaries/content/assets/education/se2014.pdf

Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. Journal of Systems and Software, 137, 96-113.

Ahmad, M. O., Liukkunen, K., & Markkula, J. (2014). Student perceptions and attitudes towards the software factory as a learning environment. In IEEE Global Engineering Education Conference, EDUCON.

Ahmad, M. O., Markkula, J., & Oivo, M. (2013). Kanban in software development: A systematic literature review. In 2013 39th Euromicro conference on software engineering and advanced applications. 9-16.

Al-Qahtani, M. F. (2012). Students' perception and attitude towards computer laboratory learning environment. Educational Research, 3(4) pp. 402-411.

Anderson, T. (2008). Theory and practice of online learning. Athabasca University Press.

Andreessen, M. (2011). Why Software Is Eating the World. The Wall Street Journal. Retrieved from http://www.djreprints.com.

Anuradha A. Gokhale. (1995). Collaborative Learning Enhances Critical Thinking, Journal of Technology Education. 7(1)

Baloche, L. (1994). Breaking Down the Walls. The Social Studies, 85(1), 25–30.

Fulwiler, T. (1986). Seeing with journals. The English Record, 32(3), 6–9.

Glover, I., & Glover, I. (2013). Proceedings of EdMedia 2013--World Conference on Educational Media and Technology. EdMedia + Innovate Learning. Vol. 2013. AACE.

Howe, S. (2010). Advances in Engineering Education Where Are We Now? Statistics on Capstone Courses Nationwide. Advances in Engineering Education.

Hrivnak, G. A. (2013). CATME Smarter TeamworkCATME Smarter Teamwork (www.CATME.org). Academy of Management Learning & Education, 12(4), 679–681.

Jane Burdett. (2003). Making Groups Work: University Students' Perceptions. International Education Journal, 4(3), 177–191.

Kaufman, D. B., Felder, R. M., & Fuller, H. (2000). Accounting for Individual Effort in Cooperative Learning Teams. Journal of Engineering Education, 89(2), 133–140.

Khine, M. S., & Lourdusamy, A. (2003). Blended learning approach in teacher education: combining face-to-face instruction, multimedia viewing and online discussion. British Journal of Educational Technology, 34(5), 671–675.

Kolb, A. Y., & Kolb, D. A. (2005). Learning Styles and Learning Spaces: Enhancing Experiential Learning in Higher Education. Source: Academy of Management Learning & Education (Vol. 4).

Layton, R. A., & Ohland, M. W. (2001). Peer Ratings Revisited: Focus on Teamwork, Not Ability. In Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition.

Martyn, M. (2003). The Hybrid Online Model: Good Practice. Educause Quarterly, 26(1), 18–23.

McGourty, J., Dominick, P., & Reilly, R. R. (1998). Incorporating student peer review and feedback into the assessment process. Frontiers in Education Conference. Moving from "Teacher-Centered" to "Learner-Centered" Education.

Newby, M., & Fisher, D. (1997). An Instrument for Assessing the Learning Environment of a Computer Laboratory. Journal of Educational Computing Research, 16(2), 179–190.

Nicole, C., Pamela, D., & Rebecca, S. (2005). Self and peer assessment in software engineering projects. In Proceedings of the 7th Australasian conference on Computing education - Volume 42 (pp. 91–100). John Garratt Publishing.

Palacin-Silva, M., Khakurel, J., Happonen, A., Hynninen, T., & Porras, J. (2017). Infusing Design Thinking into a Software Engineering Capstone Course. 30th Conference on Software Engineering Education and Training. 212–221.

Saadon, S., & Liong, C.-Y. (2012). Perception of Students on Services at the Computer Laboratory: A Case Study at the School of Mathematical Sciences, Universiti Kebangsaan Malaysia. Procedia Social and Behavioral Sciences, 59, 117–124.

Sanders, D., Dean, Sanders, & Dean. (1984). Managing and evaluating students in a directed project course. ACM SIGCSE Bulletin, 16(1), 15–25.

Sheth, S., Bell, J., & Kaiser, G. (2012). Increasing Student Engagement in Software Engineering with Gamification.

Taibi, D., et al (2016). "Free" Innovation Environments: Lessons learned from the Software Factory Initiatives. ICSEA 2016.

Tang, C. (2002). Reflective diaries as a means of facilitating and assessing reflection. In Proceedings of the 29th HERDSA Annual Conference. pp. 7–10.

Tvedt, J. D., Tesoriero, R., & Gary, K. A. (2002). The Software Factory: An Undergraduate Computer Science Curriculum. Computer Science Education, 12(1–2), 91–117.

Walker, A. Andrew E. (2015). Essential readings in problem-based learning: exploring and extending the legacy of Howard S. Barrows.