Agile Development in Software Engineering Instruction

Jaana Holvikivi, Peter Hjort

Agile development in software engineering instruction

Jaana Holvikivi, Peter Hjort

Metropolia University of Applied Sciences, Helsinki, Finland
jaana.holvikivi@metropolia.fi, peter.hjort@metropolia.fi

Abstract. Agile methods are replacing former, highly systematic project management practices in software development. Many studies have shown that agile methods are already mainstream in the software industry. Academia has incorporated these changes in development practices into education rather reluctantly. Much of higher education still depends on very traditional teaching practices and conventional curricula. In this paper, we describe a series of efforts to bring the agile world fully to ICT education, and discuss results for students and teachers alike. Agile methods can be taught, and moreover, they can also be part of the teaching toolkit. Teachers of agile courses face certain personality requirements: they need to be able to tolerate uncertainty and to be professionally proficient because of demands for flexibility and quick adjustment. The results of using agile methods as course structure, as well as agile planning of course content in small instructor teams have been successful.

Keywords: agile methods, ICT education, collaboration practices, project based learning, scrum

1. Introduction

Agile software development has replaced former, highly systematic project management practices in many areas of the software industry. Many studies have shown that agile methods are already mainstream in the industry [1,2]. However, the incorporation of the tremendous changes to development practices have entered academic education rather slowly. Much of higher education still depends on very traditional teaching practices and conventional curricula. First efforts to include agile methods to higher education were made more than a decade ago [3], and currently, courses on agile development are widely offered as part of software engineering curricula [4,5,6]. However, actual use of agile methods as educational practices is less common but a widespread use is probably on the brink of breakthrough [7]. Agile development can be applied in many kinds of project-based learning courses by replacing traditional project management with flexible practices and by replacing formal meetings with scrum meetings [8].

Collaborative problem solving and project-based learning are considered central methods to educate present day engineering students, because they simulate challenges that the students will face in professional work, such as open ended assignments, uncertainty and coordination of collaborative efforts [9]. Numerous implementations of project-based learning have been reported in various countries in recent years [10,11, 12].

Currently, education at the Metropolia University of Applied Sciences in Helsinki aims at developing the knowledge, skills, ethics, communication, and emotional components of the professional expertise required to meet the need for highly integrated competence in present day working environments. Dialogue with companies has revealed that ICT education had failed to fully respond to the new requirements in the software industry [13]. Demands from companies increasingly stress capabilities for collaboration, efficient team work and professional communication. Therefore, a curriculum reform was implemented in 2014. Project-based methods were included into most modules in the new curriculum. Additionally, the concepts of progressive inquiry [14] and problem-based learning were applied in course design. The aim was to change the studies in a way that makes entering the information technology profession a natural and exciting process regardless of student background.

In this paper, we describe a series of efforts to bring the agile world fully to the education, and discuss the results in culturally diverse groups. First, we explain trials of new practices, and then how they were extended to the entire software engineering curriculum. We discuss the results based on large amount of feedback and interview data among students and teaching staff.

2. Background

2.1 A preliminary case: Swengi - A new interface for a mobile version of a daily newspaper

The initial trial was organized as a simulated work placement for a group of students. A large Finnish newspaper wanted to attract young audiences to their tablet version. The development of tablet and mobile versions of the newspaper had thus far been centered around iPhone and iPad versions, which were quite popular. However, the paper was mainly read by middle-aged and older, well-educated urban citizens. As the use of Android devices is more common among the younger target audience, they were assumed to need a specifically designed user interface. The client was looking for fresh ideas for the user interface, and how to customize the offer of content especially for students. The task was organized as an experimental project, no monetary transactions were involved, and no functional product was expected.

The work was organized as an agile project, which emulated real workplace conditions. 14 students participated by working full time as interns in a designated office space inside the school building during three months. The project group held regular scrum meetings every morning where teachers participated as needed. In a scrum meeting, all participants stand up and explain briefly what they have done since the last meeting, what they intend to do next, and what kinds of problems they face. The students were requested to create user studies in an early phase of the project and after the first prototype was completed.

The idea was to offer a working life experience that teaches project management and team working skills in a simulated workplace setting, in addition to technical skills. Moreover, students started understanding what making a commercial product entails, and how user needs are incorporated in design. The scrum development

project was a new method for the participants. In the beginning, the group held scrum meetings every day in the morning. Teachers were involved in the beginning to show the method, set up time sheets for work, and comment on student achievements and plans. Three weeks later, teachers let the team divide into technical and user interface groups that were self-regulating. Students were allowed to decide how often they need scrum meetings, and stopped holding them daily. Soon they noticed that having fewer meetings did not facilitate the process. A similar result has been detected with other inexperienced developer teams [15].

The main result was a prototype of the application. Its design went through a couple of changes based on the reassessment by the team and client comments. In fact, the final design was unconventional enough to surprise the client. Later, some of those ideas were even implemented in the actual product.

Student self-evaluation was done continuously in scrum-meetings. Very soon it turned out that students reported rather mechanically their progress but actual self-reflection was superficial. This finding is not surprising in the light of our current university procedures where self-reflection is not systematically required or practiced. Another finding was the lack of detail in reflection and reporting. These are typical shortcomings in a technical university. They need to be addressed in curriculum development, so that self-evaluation becomes an inherent part of the study process [16].

In the closing session of the project, students were asked to answer the question "What did I learn?" The answers included many technical skills, such as Android, cursor control, and JSON. Most importantly, team work skills were seen as a major result of the internship. Students also noted their improvement in multicultural skills as the team consisted of Indian, Nepalese, Italian, Finnish, Pakistani and Vietnamese students. Additionally, they mentioned communication skills, scrum and project skills, presentation skills and English, and finally, stress management. What was not mentioned in student answers but could be observed by the teachers, was enhanced user experience understanding, user research methods, and actual design skills. Interestingly, they went unnoticed by students, even though the user survey that they conducted as well as user observations had a strong impact on the final solution. Probably this happened because the setup of the project was product-centered, which directed the concentration towards technology.

2.2 Students

The student groups in the international undergraduate information technology degree programme at the Metropolia University of Applied Sciences consist of 48-60 engineering students from a variety of nationalities, the majority of students in recent years being young Asian men from Vietnam and Nepal, but including a varied mix of other nationalities as well, particularly from Eastern Europe. In the first year, international students study separately from Finnish students, but in the second year, they begin to study together with Finnish nationals. The students are divided into smaller study groups (30-40 students) according to their major, which usually has for each study module five teachers from different professional disciplines such as mathematics, software, media engineering, and communication skills. Even though the majority of students come from a high school or secondary school background, some of them also have previous university studies in their own country. One of the

main problems with international study groups has been the slow integration to the university environment and adoption of appropriate study habits [16]. It was anticipated that participation in team work and projects would bring them closer and create a cohesive study community [17].

2.3 Curriculum reform and agile study modules

The university remodelled the entire curriculum for the academic year 2014-15. In the information technology education, the first study year is now divided into four 15 ECTS modules. Each module has a theme that introduces the different subjects students can major in: networks, programming and web-development, electronic devices, and object-oriented programming. The project in each module is supported by a varying amount of basic and theoretical studies such as mathematics and physics. The first module, a project-based orientation course enables students to acquire basic knowledge of the study environment together. The course also helps students build a social network. Consequently, they have other modules that consist of individual work and projects of different sizes.

Even though the main aim of the curriculum reform was to create large modules of 15 ECTS credits that consisted of projects, the practice has not been uniform within the modules. Actually, the modules implemented according to the new curriculum structure have various designs and arrangements for teaching. Each theme has an instructor team of 5 or 6 teachers who have a considerable degree of freedom when planning the implementation. Therefore, the ways that subjects are integrated varies a lot. The actual implementations could be classified as follows:

1. *Separated parts.* The implementations of some courses actually consist of almost separate parts. Some instructor teams simply decided to continue their earlier courses under a new umbrella, and the 15 ECTS module is divided into three disconnected parts worth of 5 ECTS each that are assessed separately.
2. *Partially integrated module.* Many implementations have a separate unit for mathematics and/or physics, but the professional content is mainly unified, even though media and programming tools or laboratory measurements are taught separately. Usually, however, there is a common project for students. The evaluation consists of several components that are summarized together.
3. *Integrated module.* Apart from the separate science classes, all professional and language content (communication skills) is integrated, and teachers collaborate in theoretical subjects and project work. Deliverables such as presentations and project documentation are assessed both on substance and communication aspects. Some types of lessons usually have more than one teacher present. Also during student team presentations most teachers attend, give feedback and evaluate together.

The integrated modules that are described in this paper and that applied agile approach, are Orientation and Games (programming) in the first year (2014 and 2015 implementations), Application Development Methods in the second year (2016 and 2017 implementations), and Software Business Start-up in the third year (2016). They were largely similar in design, lead by nearly the same teacher teams who applied agile practices in the planning of instruction.

The Orientation module had weekly assignments, most of which were completed in small teams. There were no permanent teams for the whole module.

The Games module was actually an introduction to Java programming. In the beginning of the module, students attended some lectures in programming, and completed a large number of programming assignments in a MOOC setting. The MOOC (massive open online course) was provided by the University of Helsinki. Additionally, students completed a game project during those 8 weeks. After the setup of teams in the second week, teams held weekly scrum meetings, and they were required to use kanban (Trello) for task management. In this module, the composition of teams was constrained in a way that single-nationality teams were not allowed. Earlier experiences have shown that single-nationality teams slow down the cultural adaptation process as well as leading to less successful projects [16]. On the other hand, multicultural teams teach important skills for the future of international students.

The Application Development Methods module in the second year contained a mix of Finnish and international software engineering students. As the name implies, the module concentrated on software project management skills. The lectures covered some conventional project management, a number of development tools, and user-centered design. The technical skills included setting up a Java server, creating a responsive client-side, using github and REST API, and usability testing tools. Other methods that the students were already familiar with included Trello and scrum meetings. As regular scrum meetings were not compulsory, some teams skipped most of them.

The Software Business Start-up module was conducted in the beginning of the third study year. The module built on the skills that had been acquired earlier including git version control, weekly scrum meetings, voluntary use of kanban and other agile tools. Some new technical skills were introduced, namely the so-called MEAN stack that contains node.js and noSQL databases, and use of Heroku server. Student teams were allowed to assemble freely, but the team size was limited to four.

3. Research and methods

The main results that have been proven until now are the study results. These students have not graduated yet, and therefore no data on their competences in the industry are available. For analysis of the outcomes, we collected student writings, conducted student and teacher interviews, conducted several surveys, and used feedback questionnaires. Data about the modules was also collected through field ethnography and participant observation [18], [19]. One team of educators (including some of the authors of this paper) took field notes of classroom practices, held numerous discussions on the successes and failures of pedagogical interventions, and videotaped a couple of classroom and planning sessions [20]. The study can be characterized as an explanatory building case study [21] where qualitative and descriptive methods are applied in data collection and analysis.

4. Results

4.1 Findings

The ECTS credit accumulation was investigated by examining how many students stayed on track with their studies, completing the expected 30 credits in the two study periods of the first semester. As can be seen from the figures, the number of students attaining at least 30 ECTS increased from 17% in 2013 to over 80% in the years 2014 and 2015 in the international groups (Table 1). This indicates a very successful start of the studies compared to earlier years.

Table 1. Course completions of international study groups in IT

	2012 n=74	**2013 n=76**	**2014 n=45**	**2015 n=48**
1-14 ECTS credits	23%	14%	0%	2%
15-29 ECTS credits	46%	68%	20%	8%
30->	31%	17%	80%	90%

Moreover, the completion rate of modules in successive studies was also very satisfactory. The composition of student groups varied between modules depending on their choice of major, and some Finnish students joined the groups, which explains the variation of student numbers in table 2.

Table 2. The success rate of SE students in the agile modules

Module	**Students**	**Passed**	**%**	**Average grade**
Games 1	24	22	92%	4,05
Games 2	27	26	96%	3,59
Application Development methods	41	34	83%	3,42
Software business startup	35	32	91%	3,50

The grades are on scale of 1-5, which means that an average above 4 is very good. The latest module, Software Business Start-up, showed an increasing ability of the students to apply agile development methods. Students needed no more instruction in the basic practices, and their mode of work was geared towards agile work. In the first module with scrum, namely Games, students experienced considerable anxiety, which they reported in the feedback. Even though the Software start-up module had the least distinctly defined goals, the uncertainty of the outcome was taken as granted by the

students. In that module, students were asked to brainstorm their business ideas, and teams were built around the most inspiring ideas. As a surprise to teachers, some Asian students actually appeared to have earlier start-up experience from their own countries. The cross-cultural comparison of Nepalese and Finnish experiences brought up certain similarities and some differences between business cultures, which transpired as an additional learning outcome for all. Very obviously, student ability to tolerate uncertainty increased from one module to the next.

4.2 Results of collaboration

In order to measure collaboration in the classroom, a small survey was conducted with 40 first year students after the second study module called Games. The module contained an introduction to programming using a MOOC on Java language [22], after which the students undertook a team project to develop a text-based adventure game. In addition, the course included 20% mathematics and 10% practice in web development. The amount of team work was designed to be about 50% of the total. Students were asked "What helped you most to learn in this course?" with options "Teachers", "Team-mates", "Other students" and "MOOC". The results are shown in Table 3.

Table 3. Contributors to study success in one module (%)

	Teachers	**Team**	**Others**	**MOOC**	***Share of Individual study***
Average	19	34	12	34	*50*
Standard Deviation	13	20	13	24	*20*
Max	50	60	40	85	*80*
Min	5	1	0	0	*18*

This result quite clearly confirms the observation that teacher contribution to actual work was conceived as small by the students. The role of teachers appeared more as facilitators and enablers of the study, even though actual teacher presence in the classroom was high and lectures were given regularly. The informal setting in the classroom where students sit in round tables probably made lectures appear more like discussion sessions than actual lectures. Nevertheless, the schedule of the course and the materials and assignments were prepared in advance.

The answers to the second question "What was the share (%) of your own study to your achievements (asides from team work)?" are shown in the same Table 3 in the column "Share of individual study". The variation in student perception of their own efforts was here large, as well. We can conclude that the course was not teacher-centred despite the continuous teacher presence in the classroom. The aim of students taking charge of their own learning was unarguably achieved. Other questions in this small survey were open. Many commented on the team work, which was deemed important and useful though they found it difficult to reconcile different opinions and control the process. Team work divided opinions as it caused small conflicts and frustrations, even though students agreed that it is an important skill for working life.

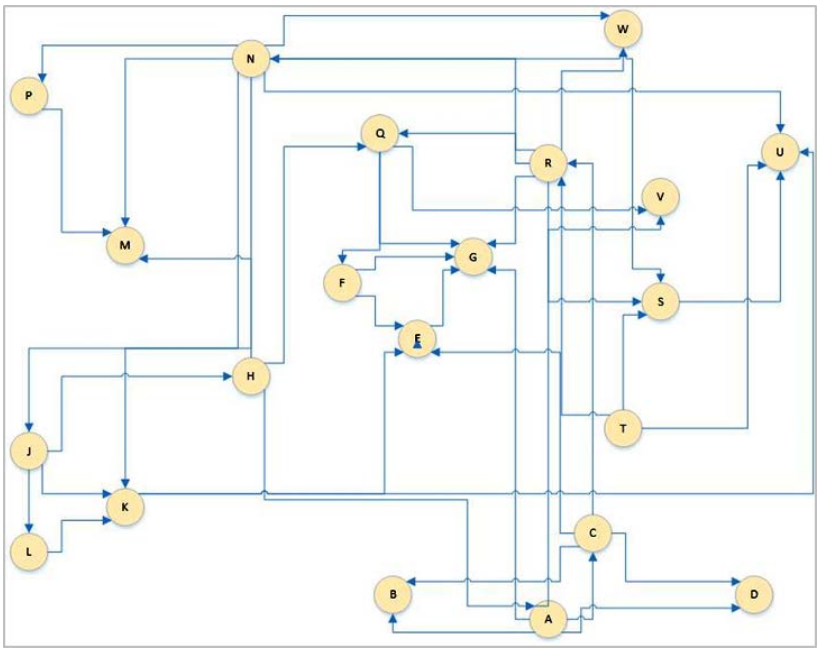

Fig. 1. Collaboration networks in one Games module group

The students were also asked to indicate who helped them and who were the students that received help form them. As can be detected from Figure 1, the mutual helping networks extended beyond team members. Teams can be observed as triads, but many nodes such as A, C, and G, have multiple connections. They are the best programmers whose advice was often sought by students outside of their teams. The average number of helping relations was 4,7, which exceeds the team size of 3.

5. Conclusion

Based on earlier studies and literature reporting teaching of agile courses, the promising outcomes of this study are not surprising. However, it remains unclear whether negative cases have been reported to the same extent as positive outcomes. As far as agile working methods have been studied in multicultural groups [4], or in courses on software business [23], our results are compatible. Nevertheless, the results of this study extend beyond agile courses, as in the present case, agile work has been expanded to the entire curriculum. Not only the subject of study, but the methods of teachers in course design and implementation, have applied agile ideas. The modules were not defined in much detail in advance, instead, the planning was flexible and done in small increments during the implementation [24]. In case of heterogeneous student groups, this allowed more freedom in the realization.

The method requires that teachers have an open mind and are ready to face uncertainty. Teachers need to have enough professional experience and confidence when they start collaboration in this format. Moreover, in our case the presence of one teacher from a different cultural background (India, as opposed to other Finnish teachers) increased credibility of the method among third world students. The results would encourage other universities to consider experimenting with agile teaching.

References

1. Kropp, M., Meier, A., Biddle, R.: Teaching Agile Collaboration Skills in the Classroom, IEEE 29th International Conference on Software Engineering Education and Training (2016)
2. Laanti, M., Salo, O., Abrahamsson, P.: Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation, Information and Software Technology vol. 53, pp. 276-290 (2011).
3. Rico, D.F, Sayani, H.H.: Use of agile methods in software engineering education, Agile Conference, AGILE'09. (2009)
4. Paasivaara, M., Blincoe, K., Lassenius, C., Damian, D., Sheoran, J., Harrison, F., Chhabra, P., Yussuf, A., Isotalo, V.: Learning Global Agile Software Engineering Using Same-Site and Cross-Site Teams. 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. 285-294 (2015)
5. Mahnic, V.: A Capstone Course on Agile Software Development Using Scrum, IEEE Trans on Education, Vol 55, 1 (2012)
6. Deved˘zic´, V., Milenkovic, S.R.: Teaching Agile Software Development: A Case Study, IEEE Trans on Education, Vol 54, 2 (2011)
7. Gannod, G., Troy, D. A., Luczaj, J.E., Rover, D. T., Agile Way of Educating; IEEE workshop proposal 2015. (2015)
8. Longmuß, J., Höhne, B.P., Bräutigam, S., Oberländer, A., Schindler, F.: Agile learning: Bridging the gap between industry and university, A model approach to embedded learning and competence development for the future workforce. In the Proceedings of 44th Annual SEFI Conference (2016)
9. Crawley, E.F., Malmqvist, J., Östlund, S., Brodeur, D. R.: Rethinking engineering education. The CDIO Approach, Springer, New York (2007)
10. Marjoram, T.: An introduction to the organization of the profession. In: Engineering: Issues, Challenges and Opportunities for Development. Paris: UNESCO. pp. 135-136. (2010)
11. Dalsgaard, F., Du, X., Kolmos, A.: Innovative application of a new PBL model to interdisciplinary and intercultural projects, International Journal of Electrical Engineering Education 47, 174-188 (2010)
12. Kolmos, A., Dahms, M., Du, X.: Problem-based Learning. In: Engineering: Issues, Challenges and Opportunities for Development. Paris: UNESCO. 334-340 (2010)
13. Holvikivi J., Lakkala M., Muukkonen, H.: Introducing collaborative practices to undergraduate studies. In: Brinda T., Mavengere N., Haukijärvi I., Lewin C., Passey D. (eds) Stakeholders and Information Technology in Education. SaITE 2016. IFIP Advances in Information and Communication Technology, vol 493. Springer, Cham (2017)
14. Paavola, S., Hakkarainen, K.: Trialogical approach for knowledge creation. In S. C. Tan, H. J. So & J. Yeo (eds.), Knowledge creation in education, 53–73. Singapore: Springer. (2012)

15. Kropp, M., Meier, A., Perellano, G.: Experience Report of Teaching Agile Collaboration and Values: Agile Software Development in Large Student Teams, IEEE 29th International Conference on Software Engineering Education and Training (2016)
16. Holvikivi, J.: Culture and cognition in information technology education, SimLab publications, Dissertation series 5, Espoo, Finland (2009)
17. Hjort, P., Holvikivi, J., Vesikivi, P., Lukkarinen, S.: Student Collaboration and Independence from Day One in Higher Education. In The Proceedings of the 43rd Annual SEFI Conference, Orléans, France (2015)
18. Spradley, J.P.: Participant Observation. USA: Thomson Learning. (1980)
19. Green, J., Bloome, D.: Ethnography and ethnographers of and in education: A situated perspective. In Flood J, Heath SB and Lapp D (Eds.), Handbook of research on teaching literacy through the communicative and visual arts. Macmillan Publishers, NY. 181-202 (1997)
20. Vesikivi, P., Hjort, P., Lakkala, M., Holvikivi, J., Lukkarinen, S.: Adoption of a New Project - Based Learning (PBL) Curriculum in Information Technology. In The Proceedings of the 43rd Annual SEFI Conference, Orléans, France (2015)
21. Yin, R.K.: Case study research. Design and methods (5th ed.). USA: Sage Publications (2014)
22. Eckerdal, A., Kinnunen , P., Thota, N., Nylén, A., Sheard, J., Malmi, L.: Teaching and learning with MOOCs: computing academics' perspectives and engagement. In: Proceedings of the 2014 conference on Innovation & technology in computer science education: Uppsala, Sweden, ACM Press. (2014)
23. Read, A., Derrick, D.C., Ligon, G.S.: Developing Entrepreneurial Skills in IT Courses: The Role of Agile Software Development Practices in Producing Successful Student Initiated Products, 47th Hawaii International Conference on System Science, pp. 201-209 (2014)
24. Vesikivi, P., Lakkala, M., Holvikivi, J., Lukkarinen, S.: Teacher collaboration in IT project courses: resistance and success. SEFI 2016 Annual conference. Tampere (2016)