

Agile method in education

Enikő Ilyés

Abstract: Nowadays agile software development has become popular. This fact allows software development programs in higher education institutes to refresh their curriculum so that students can be prepared for the use of agile methods. One approach to this is to apply Scrum in software development courses. But this constitutes a considerable challenge too. Since students only work together 1,5 hours a week, they don't have the possibility to practice Scrum daily. They don't have enough time for detailed sprint planning, assigning team members individually since that's the requirement of the educational system is nonagile, etc. Based on international research and case studies from Eötvös Loránd University, this article presents some critical questions and good practices regarding using Scrum in a software development course.

Introduction

Using Scrum in a traditional software development course (working 1,5 hours together and 6,5 hours alone, at home per week) raises many questions.

An important argument against using traditional Scrum is that students meet only once per week, so they don't have the possibility to practice Scrum every day. What length should a sprint have in this situation? This decision influences the time needed for sprint planning, the frequency of demos and reviews. These are important from the education point of view because of the lack of experience the students have in software development and management. The special roles of Scrum must be carefully chosen too. How can the appropriate student be chosen for a role if the teachers don't know the students ahead of time? Should the teachers fulfill all special roles? How can the students be motivated to function as a self-organized, independent team, having common responsibilities but still be evaluated individually since that is the requirement of the educational system? The boundaries of a software development course control the size, complexity of the software to be realized during the semester hence choosing the right project is very important. Finally, the biggest challenge is to transmit the agile attitude that runs along Scrum roles, events, products, especially because of the time constraints that an education course has

International research

Teachers across the globe highlight a few aspects to be considered while using agile methods in education.

According to researchers from Texas University, it is very important to choose the proper project to be completed during a software engineering course[1]. This assumes that the project has the right size and complexity, so that the students can experience the necessity of using a software engineering method and succeed while using it too. A professor from Iowa University considers fulfilling the role of project owner the critical factor, because he/she seriously affects the success of the project by writing the user stories in a proper way, prioritizing them, keeping in contact with the customer and the project team as well[2]. Instead of one meeting per week, longer periods (for example: one-two weeks) assigned for team project work would be much more beneficial for students. To archive this, restructuring the semesters would probably be necessary which could result in a lot of pushback[3]. Teachers from Maryland University identify the need for initial training, where students learn about the agile values, develop their soft skills, so that they can apply them during the course [4]. Only this could assure that the power of agile methods would be manifested. All teachers agree, that using agile methods during software development courses is a current research problem and many aspects of it still need to be explored further.

Case studies

In 2016 six teachers from Eötvös Loránd University's Faculty of Informatics decided to use Scrum for a student-team project during a software engineering course.

There were three different courses, each of them adopted Scrum in a different way. There was a course, where the teacher fulfilled the role of Scrum master, product owner and customer as well. In another course the roles were assigned to several students. There were 2-week and 8-week long sprints, teams with 10 and teams with 3 members. As the assessment method, an interview-like test was used in one course and a complex evaluation schema was applied in another where students received points from one another too.

Teams have been analyzed along the Scrum roles, events and products. Another important group constitutes the solutions regarding education related questions like: How to choose the adequate software project to be realized? How to gain the most from the little time the students can work together? How to evaluate the students individually? etc.

Conclusions

I monitored three different courses two semesters long. At the end of the semesters I composed a questionnaire (based on international research papers, expanded with local specialties), and analyzed the results. Interviews were also taken (from teachers and students). International research conclusions were contracted with local conclusions.

This section presents the result of my qualitative empirical study, which is a list of good practices concerning all the important decisions regarding the use of Scrum in an academic course. The suggestions trace out a concrete method of applying Scrum at a software engineering course.

Scrum roles

It can give extra motivation to the students and make the simulation of software development more authentic if the project has a real customer. (This can be for example a teacher from another faculty of the university). The Scrum master and the project owner role can be fulfilled by students – this can assure that the respective ones get some valuable experience regarding Scrum roles. This is at the same time risky for the project team because the success of the software development process highly depends on these key roles. Therefore, it is recommended to use the right methods like interviews or tests, while choosing the appropriate students. It might be necessary to build a stronger relationship with the product owner student, because usually it is strange for them to stand for the interests of the customer rather than for interest of the (student) project team, to be strict about the acceptance criteria of user stories. The Scrum master student must get more attention from the teacher too. A separate short training regarding his/her role is recommended at the beginning of the semester. His/her attitude to the project can easily influence the attitude of all team members.

All students should be part of one Scrum team and the teams should be encouraged to get self-organized, independent, to pay attention to communication. Communication can be supported, by establishing an effective communication channel at the very beginning of the course; for example, beginning the course with a longer introduction cycle, ice breaker games, or team events. If the students meet once per week, I suggest creating maximum 5-member Scrum teams.

Scrum events

Three-week long sprints could be ideal considering that semesters are 13 weeks long. Choosing a longer sprint would not be profitable because the rare occurrence of demos and reviews. Using planning poker while planning the sprint can be very exciting for students. It is more like a game for them because they don't have enough experience in software development to archive a relevant estimation. But they can profit a lot from discussing the user stories repeatedly till a common estimation comes off. It is a time-consuming procedure, so it must be well prepared.

The students don't have the possibility to practice the daily stand up every day, so these must be realized during weekly meetings. It is important to keep the characteristics of the daily Scrum event so keep them short and compact. The original concept of the demo must be accomplished too: fluent, accurate, careful presentation of the software to the customer. The students enjoy learning different review techniques, diagrams and statistics. The teachers can benefit from the reviews even more: they get feedback from the students, can collect information about the teams, their needs, their concept about Scrum and agile values. It is very important to emphasize the team's strengths too, not only the weaknesses.

Scrum products

It is important to start the project with a good product vision. This can be written by the teacher or the students can practice writing it too. As far as the product backlog, the details are very important, especially the acceptance criteria. If we miss this at the very beginning it can cause a big disappointment for students later. The teachers must assist the project owner student because he/she may have little experience with user stories and this can cause trouble for the whole project team. The students have the tendency to forget about the Scrum table. As teachers, we may have to come up with different techniques to remind them of the Scrum table.

Education

In most cases the teachers cannot find out ahead of time what kind of technologies the students of the course are familiar with and at what level they are. Regardless, the teachers must choose a software to be assigned which presents a challenge, but a solvable one. One good practice is to opt for a software that can be realized in both a basic and a complex way to. Even if the focus of the course is the software product, it is a good idea to assign a period at the beginning of the course to get ready, to get to know the software development method. The positive effect of software development methods can only be expected if the team members know and apply the method correctly. This is especially true for agile methods.

The evaluation of the students is a critical factor too. If we put agile project management in the focus of a software development course, then we must evaluate students during the course to observe how they contrive the agile values too. This is hard to measure. Personal observation and the feedback of other team members (using anonymous questionnaires) can be used for it. We can reflect on different aspects in regard to a team member: communication, collaboration, adaptation to change, etc.

All in all, teachers and students agree that a 1,5 hours long class per week is too short for both team meetings and software development actions. The daily Scrum, communication, Scrum roles could be realized more effectively if the time dedicated for the class would be longer. At the same time, I consider that if students follow the good practices collected in this article in regard to Scrum that will significantly contributes to the success of software development projects. Considering other advantages as well [5], I recommend the use of agile methods in software engineering education.

Acknowledgements

The project has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

References

- [1] K. M. C. C. S. L. Tucker Smith, "Software Engineering Senior Design Course: Experiences with Agile Game Development in a Capstone Project", ACM, 2011
- [2] J. G. Kuhl, "Incorporation of Agile Development Methodology a Capstone Software Engineering Project Course", ASEE, 2014.
- [3] F. M. Craig Anslow, "An Experience Report at Teaching a Group Based Agile Software Development Project Course", ACM, 2015.
- [4] D. D. F. R. a. D. H. H. Sayani, "Use of Agile methods in software engineering education", IEEE, 2009.
- [5] E. Ilyés, „Agilis szoftverfejlesztési gyakorlat megvalósításának lehetősége az ELTE Informatikai Karán”, 2017.