ATLANTA
120th ASEE Annual Conference & Exposition
FRANKLY, WE DO GIVE A D*MN
June 23-26, 2013

Paper ID #7108

Increasing student responsibility in design projects with agile methods

Dr. Martin Grimheden, Royal Institute of Technology (KTH)

Increasing student responsibility in design projects with agile methods

Abstract

This paper attempts to investigate the potential of merging agile methods with student projects in higher engineering education. The context of this study consists of a number of capstone projects within two comparable courses in the subjects of mechatronics and embedded systems given at KTH Royal Institute of Technology, Stockholm, Sweden. In the capstone project described here, students work in teams of about 10 students, over a period of about nine months.

Six capstone projects are studied; three of these were organized according to scrum [1] and three according to more formal methods. The six projects involved in total 54 students. The six capstone projects were divided into two groups, following two different courses. Students of both courses took a course in project management, either prior to the capstone course or in parallel. One of the two project management courses emphasized agile methods, the other more formal methods. The student teams who followed the project management course in agile methods were encouraged to organize their teams and projects according to scrum.

At the core of scrum is the notion of empowering the team to organize the tasks independently together with the idea of quick prototyping for fast customer feedback. Formal methods, in context, rely more heavily on documentation, planning and preparation. The hypothesis for this study is that delegating the responsibility of project organization to the student team would motivate the students to take a greater responsibility for both the project and their own learning, and, that this would promote increased student learning by way of motivating student responsibility.

Students of the three scrum-teams took a large responsibility for organizational aspects; more focus on organizational issues, larger responsibility for activities related to the course, and in aspects and actions related to achieving the learning goals of the course.

In comparison between the two categories of projects, it can be seen that the scrum-teams showed more signs of taking responsibility for achieving learning goals than in the non-scrum-teams. While the non-scrum-teams showed a great dedication toward finalizing project results, the scrum-teams also showed dedication toward performing activities with the purpose of reaching a learning goal not directly necessary for the project results. Even if the learning achievements are hard to measure, the anecdotal evidence of increased responsibility for the learning process shows signs of increased learning related to the course goals.

Introduction

This paper presents a comparison between two capstone courses given at KTH Royal Institute of Technology, Stockholm, Sweden. The first capstone course has been given since 1984, while the second was given for the first time in 2012. The purpose of this comparison has been to better understand and evaluate changes made to the first capstone course.

The first capstone course, Mechatronics Advanced Course (referred to as the Mechatronics capstone course) is given over a period of two semesters, covers 24 ECTS credits in total, and typically attracts around 30-40 students per year. These students are enrolled either in a MSc program in Engineering Design or in Industrial Engineering and Management with a specialization in Mechatronics.

The second capstone course, Embedded Systems Project Course (referred to as the Embedded Systems capstone course) is given over one semester, covers 9 ECTS credits and attracted 21 students for the first course round. These students are all enrolled in a MSc program in Embedded Systems. The 9 ECTS does not include a project management module of 6 ECTS that relates strongly to the capstone course.

Both courses are started with the presentation of a design brief, with a project provided by an industrial sponsor, and conclude with the presentation of a functional prototype at a design fair. The courses differs mainly in terms of coaching and supervision: the embedded systems capstone course involves one coach per project and the mechatronics capstone course one teaching team that coaches all teams in parallel.

Increasing student responsibility using agile methods

Starting in 2011, agile methods were introduced into the mechatronics capstone course. This has been described and evaluated in previous publications [2], [3], [4]. The main driver behind this move is that agile methods provides methods for, or enables, motivation for delegating responsibilities to a team. Agile methods emerged in the software industry in sharp contrast to more formal methods, such as the V-model and the Waterfall method. The Agile Manifesto [5], [6] intended to increase the responsibility of the team, of empowering a team of peers to perform the task given to the team. Responsibility and the tools necessary to take this responsibility were crucial for success.

The hypotheses for previous studies in this field have been that the team empowerment encouraged by agile methods is suitable for an educational context and provides a favorable basis for learning. See for example [2]. In this paper, we wish to extend the hypothesis one step further, that agile methods enhance opportunities for learning by increasing student responsibility.

Few educators would object to the statement that increased student responsibility is beneficial for learning. Many articles have been written about this relation, for example studies of how students participating in the decision-making process of a course can motivate students to take more responsibility for their learning [7], [8], [9], [10]. Other examples have shown that students enjoy learning more when given the opportunity to direct their own learning [10], [11].

At the core of agile methods is the empowerment of the development team to get, and take responsibility for the organization of the tasks at hand. The team is responsible for delivery, and given the freedom (and responsibility) of the daily planning. References such as [2], [12], [13] promotes these aspects of agile methods.

Case study

The study was undertaken from January to December 2012. The study involved two capstone courses, with the total of six capstone design projects and 54 students.

Key factor/capstone course	**Mechatronics**	**Embedded Systems**
Time period	March-December 2012	August-December 2012
Number of students	33	21
Size	24 ECTS including a 6 ECTS-module in project management	9 ECTS excluding a 6 ECTS-module in project management
Actual size, excluding project management module	18 ECTS	9 ECTS
Number of projects	3	3
Number of students per project	10/11/12	7/7/7
Typical project budget	€ 5.000	€ 2.500
Team supervision	A teaching team met regularly with the teams, performing design reviews.	One coach/faculty member per team. The coach met bi-weekly with the student team.
Individual supervision and feedback	Individual meetings between teaching team member and student, three times, 15 minutes per meeting. During this meeting, the student was given feedback on the individual learning progress.	Individual meetings between coach and student, two times, 15 minutes per meeting.
Grading	Individual grade set by teaching team	Individual grade set by project coach

Table 1: Comparing the two courses

Agile methods vs formal methods

The main difference between the two courses is mainly that the mechatronics students were encouraged to use agile methods and the embedded systems students were required to submit a detailed project plan in the beginning of the capstone course. In reality, these differences were subtle and not that well communicated. The mechatronics students were heavily influenced by comments and experiences from previous students. The teaching team gave a lecture on Scrum and agile methods, individual members of the teaching team strongly influenced the teams to adopt agile methods, and Scrum in particular.

For the mechatronics capstone course, the teaching team gave a 6 ECTS project management module, where agile methods were encouraged. The embedded systems capstone course did not include this module, these students instead took a separate module which did not encourage agile methods. This other module on the other hand strongly encouraged formal methods and required the students to submit an extensive project plan and regular status reports. The requirements of the project plan included submitting detailed time plans, resource allocation, list of responsibilities, deliverables etc.

Project organization	**Mechatronics**	**Embedded Systems**
Encouragement from teaching team	"Don't plan – prototype instead"	"Plan your project thoroughly"
Formal weekly requirements	Weekly logbook required, with burn-down chart	Weekly status-reports required
Formal project requirements	Three presentations for company representatives and teaching team (May 2012, October 2012 and December 2012) One final prototype and report	Two project presentations for company representatives and teaching team (October 2012 and December 2012) One final prototype and report
Other requirements, relevant for project organization	None	Required to submit a detailed project plan in October 2012

Table 2: Comparison of course deliverables

Description of the two processes

Extremely simplified, the embedded systems teams were asked to provide a project plan half-time in the project and to follow that until the end of the course. The mechatronics teams were asked to provide a final prototype and a written report at the end of the course, and work agile to get there. The embedded systems teams all planned according to a waterfall- or V-model, while the mechatronics teams all used Scrum. This is a simplified description since the mechatronics teams also engaged in heavy planning in the beginning of each period, semester etc to define a "backlog" of tasks to do during that particular period. The main difference is though that the embedded systems team had a stage-gate in the middle of the process, with a formal acceptance from the teaching team of their project plan, with extensive feedback.

In previous studies [2], [3], [4], the combination of agile methods and mechatronics capstone design courses have been presented. It has been showed that the use of Scrum in the mechatronics projects changed how the student teams defined their projects and the relation between the companies and the student teams. In [2], Scrum replaces the previously used waterfall model in a mechatronics capstone design project. As an example, the use of Scrum encouraged the student teams to discuss and question the presented design brief, and in a few cases made the companies re-think the project specifications and requirements. It was also argued that the use of Scrum encouraged the teams to "prototype earlier", by promoting a more iterative product development process. For this paper, the main differences between these two approaches is that the formal approach, as performed by the embedded systems teams implied that the students should plan the entire project extensively and, ultimately, get this project plan approved by the faculty. The mechatronics students' teams did not have any check-point or stage-gate in this sense, but rather encouraged to seek feedback instead of approval.

Empirical data supporting the study

At the end of the course, a week after project delivery, the six student teams met for reflections. During this day, a number of exercises were held. Each student met individually for 15 minutes with a faculty member to discuss final grade and the student teams gave feedback to each other and to the faculty. In one exercise, the students answered a number of questions related to the entire course and projects, with the intention of capturing important reflections upon the product development process. The students spent a considerable amount of time to create these responses, and they can be considered reflecting the majority of the students within the respective teams (since submitted by the teams).

The team's responses are presented below, arranged per team. Some responses are given without context, as they were also presented orally by the student teams, some comments are therefore given. In the following, they are therefore presented with the authors' explanation and analysis.

The first team comments refer to the process perspective of product development. The teams all responded to the question: "What are the top-three challenges when conducting a project like this, from a process perspective?"

Team	**Team comments (all quotes from student teams)**	**Comments, authors analysis**
Mechatronics team A	• No team leaders • Workload balancing, a lot of work needed in periods • Not here on the same time, due to other courses etc	The top-three challenges relate to team organization and the division of tasks, to conflicting schedules. The top-three challenges therefore mainly relates to coordination within the team.
Mechatronics team B	• Having an overview of the whole project • Following Scrum • Estimating time spent on tasks	The top-three challenges relate both to project management but also to agile methods. It is hard to estimate time and it's hard to follow a method.
Mechatronics team C	• Reach consensus of what to develop, goals within group and within product owner can be widely scattered • Establishing a suitable project working method when working within multiple disciplines such as software, mechanics etc	The top-three challenges relate both to conflicts between individual goals and project goals, and to methods that could be identified as part of Scrum.

	• Create a reasonable time plan and follow it accordingly	
Embedded systems team D	• To keep the project together from a project management perspective and to keep track of the project progress • Give an estimation of the required time and energy is hard without enough experience and proper background • True development of the project was heavily dependent on having a solid concept beforehand (typical R&D projects)	The top-three challenges relates to overall project management, to estimate the tasks and to developing (or having) a “solid concept beforehand”, meaning knowing what to develop.
Embedded systems team E	• To get the client’s requirements and make out something that correspond to their initial will • Arrange the work evenly and make everyone do the part that they like • To hear everyone’s voice and adopt everyone’s idea. There will always be compromises.	The top-three challenges relate to understanding the design brief, knowing what to develop, and to overall project management and team motivation.
Embedded systems team F	• Keeping the team together, so that every member knows what is going on with each part of the project • Dealing with dependencies etc. A component is needed by other groups and the component is unavailable for sometime • Keeping the spirit up, especially in the middle phase when the groups motivation has decreased	The top-three challenges relates to overall project management, team motivation.

Table 3: Team responses to “What are the top-three challenges when conducting a project like this, from a process perspective?”

Analysis of responses

When analyzing the team's responses to the top-three challenges, a distinction between the mechatronics teams and the embedded systems teams can be seen. The three embedded systems teams all state that the top-three challenges relates to the overall project management: knowing what to do, keeping the project together, having a clear concept, getting the requirements etc. When comparing these with the mechatronics teams, the mechatronics team's comments are considerably more specific and relates not to the overall project organization but rather to how to deal with specific aspects of it or, as in team A:s case, to more practical issues such as conflicting schedules.

In particular, the following is found:

- The embedded systems teams express more challenges relating to team spirit and team motivation
- The embedded systems teams express more challenges relating to understanding the project, to company requirements, to client specifications
- The mechatronics teams refer more explicitly to their method(s) and to challenges related to following these.

The second study refers to overall learning outcomes. The student teams were asked to provide the "top three learning outcomes from the course":

Team	**Team comments (all quotes from student teams)**
Mechatronics team A	• Management of the work in large project groups • Making own engineering decisions • LabView and graphical programming. NI components
Mechatronics team B	• How to organize ourselves in a big project • How working for a company really is (experience) • Acquiring knowledge
Mechatronics team C	• Project organization • Acting as consultants towards product owners • To take responsibility towards the team and the project owner for your technical solution
Embedded systems team D	• Working with requirements making decisions and deliver things • Collaborating in a team • Learn to handle responsibilities
Embedded systems team E	• Experience the full cycle of product realization (to start with a problem and ending up with a prototype) • Handling with large team project • Learn much about RF and localization techniques
Embedded systems team F	• Time and resource (people) management • Reverse engineering • Learning new tools and technical knowledge on hardware components

Table 4: Responses to “What are the top three learning outcomes from the course”

Analysis of responses

When comparing the top overall learning outcomes between the two courses, it’s harder to see any noticeable difference – rather the six teams seem to have made very similar learning experience. All six teams relates their top learning outcomes to both process related, meaning project management, and to improved technical skills and knowledge.

Grades

Table 5 below presents the grades given to the six projects. The grading was done by a team of faculty representatives. The grades reflect the project results and not necessarily the individual learning. As shown in the table, the grades are about the same for the two groups of projects, with a slightly higher average for the embedded systems teams compared to the mechatronics teams.

Team	**Grade**
Mechatronics team A	**B**
Mechatronics team B	**B-**
Mechatronics team C	**C**
Embedded systems team D	**A-**
Embedded systems team E	**B**
Embedded systems team F	**C**

Table 5: Project grades

Summarizing the empirical data

The data that is presented so far does not complete the picture and cannot on its own justify that agile methods increase student responsibility and enhance individual learning, i.e. the hypothesis put forward in the introduction of this article.

On the other hand, the data presented above relating to grades and top learning outcomes, does not clearly differ between the two groups of projects. Top learning outcomes and grades are mostly similar.

In this perspective, a conclusion can be made, that the agile approach did not affect in a negative way. The data cannot be used to show that agile methods did not work in this context.

Supporting anecdotal evidence

To get supportive anecdotal evidence for the argument, discussions with the faculty members and company representatives were held. In these discussions, the following arguments were put forward:

- The embedded systems teams spent a lot of time planning their projects in the initial phases. They produced extensive documentation with project plans etc. None of the teams were able to follow the plans and about half-time in the projects the plans were basically scrapped. After the scrapping of the plans, no new plans were made, the teams basically worked ad-hoc.

- The mechatronics teams also spent considerable time planning. These teams however mainly planned for the next sprint, the next phase, and re-planned after each sprint.
- Both groups of teams re-negotiated project requirements with the companies.
- The mechatronics teams delivered more complete projects than the embedded systems teams.
- The faculty regularly coached and supervised the embedded systems teams, with one coach per project. The coach took an active role in project planning.
- The faculty did not take any active role in project planning for the mechatronics teams. The faculty did take an active role in coaching the scrum-masters.

To conclude, the two groups of teams were coached and supervised differently by the faculty. Basically, the mechatronics teams were given a larger responsibility of project planning and organization than the embedded systems teams.

Discussion

Even if it’s too simplified to state that three teams did use agile methods (Scrum) and three did not, it is clear that the three mechatronics teams did to varying extents try to work agile and was encouraged to use Scrum. The three embedded systems teams all made extensive project plans according to either the waterfall- or the V-model.

The three embedded systems teams project plans has not been under scrutiny in this article, but it is clear that they were created, were extensive, but they were ultimately not followed. In all three cases the three teams heavily underestimated the task, the available time and resources, and consequently the plans were scrapped almost as soon as they were submitted to the teaching team.

The comparison also falters since the two courses differ on so many points, for example the mechatronics course is twice as large as the embedded systems course.

These facts set aside, the following conclusions can still be made:

Conclusions

Two capstone courses have been studied. Both had equally-sized modules where project organization was taught. In one, agile methods were taught side by side with more traditional methods, and consequently agile methods were encouraged in the projects. In the other capstone course, a formal method was required. Student teams of both capstone courses expressed similar learning outcomes, both relating to subject matter learning and to project organization skills.

In the teams where formal methods were required, the students spent a considerable amount of time on formal planning. Most of this was not used in the actual projects. The agile teams were not required to submit nor produce any planning documents.

When comparing the teams statements related to top-three challenges regarding a process perspective, meaning project organization, the agile teams pointed to challenges related to implementing certain aspects of the agile method used, while the teams using formal methods pointed more to challenges related to overall project management. From this it can be assumed that the formal teams were more unsecure in their project plan and chosen project organization than the agile teams; the agile teams were more confident in their chosen method even if they still found challenges in its implementation.

In stronger words, there are signs (anecdotal evidence) that the formal teams did not rely on methods to the same extent that the agile teams did. The agile teams were encouraged to use agile methods, and the formal teams were required to submit a formal project plan. At the core of agile methods, and Scrum in particular, is the notion that the team takes responsibility for the process, which the faculty saw happened with these three teams.

The only hard fact put forward is that, the two groups of teams expressed similar learning achievements and received similar grades. Both groups of teams however expressed quite different challenges relating to project management. This, when put into the perspective of the different coaching and education the two groups were subjected to, can motivate the conclusion that the agile methods did not negatively affect the teams, despite the fact that formal project planning was not required nor supported nor encouraged.

Future research

The hypothesis put forward in this article still is not completely proven. The author believes that there is anecdotal evidence that support the hypothesis, but since the responsibility and effort undertaken by the faculty is not scrutinized nor measured in this article, the author believes that this needs to be studied further to complete the picture.

[1] http://www.scrum.org/storage/scrumguides/Scrum_Guide.pdf, 2012-03-11.

[2] M. Edin Grimheden, Can agile methods enhance mechatronics design education? Accepted for publication in Mechatronics. (2013)

[3] M. Edin Grimheden, Mutual learning experiences – mechatronics capstone course projects based on Scrum. In: Proceedings of the 119th Annual ASEE Conference and Exposition, San Antonio, USA. (2012)

[4] M. Edin Grimheden, Can agile methods enhance mechatronics education? Experiences from basing a capstone course on SCRUM. In: Proceedings of the 119th Annual ASEE Conference and Exposition, San Antonio, USA. (2012)

[5] B. Boehm, Get ready for agile methods, with care, Computer, 35 (2002), pp. 64-69.

[6] http://www.agilemanifesto.org/principles.html, 2012-03-11.

[7] S. Bolhuis, Towards active and self-directed learning: Preparing for lifelong learning, with reference to Dutch secondary education. In: Proceedings of Annual Meeting for the American Educational Research Association, New York, NY. (1996)

[8] D. R. Garrison, Self-directed learning: Toward a comprehensive model. Adult Education Quarterly, (48)1, pp. 18-33. (1997)

[9] H. L., Jr. Hom, M. D. Murphy, Low achiever's performance: The positive impact of a self- directed goal. Personality and Social Psychology Bulletin, 11, pp. 275-285. (1983)

[10] B. Taylor, Self-directed learning. In: Proceedings of the Combined Meeting of the Great Lakes and Southeast International Reading Association, Nashville, TN. (1995)

[11] C. Temple, M. L. Rodero, Active learning in a democratic classroom: The "pedagogical invariants" of celestin freinet (reading around the world). Reading Teacher, (49)2, pp. 164-167. (1995)

[12] P. Chetankumar, M. Ramachandran, Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices International Journal of Software Engineering 2.1 (2009): pp. 3-28.

[13] B. Schatz, I. Abdelshafi. The agile marathon. Agile Conference, 2006. IEEE, (2006)