AC 2012-4312: CAN AGILE METHODS ENHANCE MECHATRONICS EDUCATION? EXPERIENCES FROM BASING A CAPSTONE COURSE ON SCRUM

Dr. Martin Edin Grimheden, Royal Institute of Technology

Martin Edin Grimheden currently holds a position as Associate Professor at KTH and is the Director of Mechatronics Education at KTH.

Can agile methods enhance mechatronics education? Experiences from basing a capstone course on SCRUM.

Abstract

In 2011, an experiment was undertaken at KTH Royal Institute of Technology to introduce agile methods for product development into a mechatronics capstone course. This paper describes the method used, Scrum, and the context of the studied mechatronics capstone course. Mechatronics is here defined as "synergistic integration" of electronics, mechanical engineering, control and software engineering. Mechatronics product development, in this context, therefore deals with the development of complex and intelligent products, which implies multi-disciplinary work and the use of models etc. from several domains and areas.

With the integration of Scrum into the mechatronics capstone course, an educational favorable alternative is identified, to previously used design methodologies such as more traditional stage-gate methods as the Waterfall or method or the V-model. This is due to the emphasis on early prototyping, quick feedback and incremental development. It still might not be the favorable method for use in large scale industrial development projects where formal procedures might still be preferred, but the pedagogical advantages in mechatronics education are valuable. Incremental development and rapid prototyping for example gives many opportunities to reflect and improve. The Scrum focus on self-organizing teams also provides a platform to practice project organization, by empowering students to take responsibility for the product development process.

Among the results of this study, it is shown that it is possible and favorable to integrate Scrum in a mechatronics capstone course and that this can enhance student preparation for a future career as mechatronics product developers. It is also shown that this prepares the students with a larger flexibility to handle the increased complexity in mechatronics product development and thereby enabling the project teams to deliver results faster, more reliable and with higher quality.

Introduction

The experiment described in this paper is based on a capstone course in Mechatronics given at KTH, the Royal Institute of Technology, since 1984. The challenge of today's capstone courses and projects at KTH are not the specific technical competencies but rather organizational issues, as it has been experienced it over many years. The projects are becoming increasingly complex and based on knowledge and competencies in several fields, spread over all members of the student team (and faculty). At KTH, most resources in terms of course development and curriculum design development have been invested on topics such as product development, project organization, teamwork and mechatronics design methodologies. In summary, the trend has been to move from teaching ‘mechatronics’ to teaching ‘mechatronics product development’.

Starting from March 2011, this trend has been emphasized by an organizational shift; prior the students in the capstone projects were organized according to traditional organizations with student project leaders, division of responsibilities. Instead, the faculty now encourages and coaches the students in self-organizing dynamic teams using the Scrum approach. This does affect the learning process within the capstone course by better enabling the delegation of the responsibility to the teams.

In the first section of this paper, agile methods for product development are presented in general, and the specific method used – Scrum. In the next section, mechatronics at KTH Royal Institute of Technology and the studied capstone course is presented. In the following sections, findings are presented with a discussion of the results from combining mechatronics product development with agile methods for product development.

Agile methods and product development

Agile software development methods have changed the way many software developers organize their work and projects, for example as in Scrum by delegating responsibilities, empowering individuals and delaying decisions. The main driver in most methods is “accelerated delivery” realized by focusing on small steps, incremental development, prototyping and quick feedback rather than extensive planning and documentation [1].

The “Agile Manifesto” [2] is based on twelve principles to follow as an agile software developer: from “…satisfying the customer through early and continuous delivery…”, “…welcome changing requirements…”, “…business people and developers must work together…”, “…face-to-face conversation…”, “…self-organizing teams…” to “…the team reflects on how to become more effective…” [2].

The Agile Manifesto was written in 2001 even though the heart of agile methods can be traced back to 1957 [3]. In the mid-1990s so called “lightweight methods” emerged as a reaction to the then established methods such as the V-model, the Waterfall model and various implementations of stage-gate models. These models were then categorized as “heavyweight methods” (especially the Waterfall model) since they rely strongly on heavy documentation, formal processes for control and verification, planning etc. The original authors of the Agile Manifesto saw a need for an alternative to this, to “document-driven, heavyweight software development processes” [2]. The alternative, as stated in the twelve principles, is also described in four value statements, to value “individuals and interactions over processes and tools”, “working software over comprehensive documentation”, “customer collaboration over contract negotiation” and “responding to change over following a plan”.

Scrum

Scrum is an example of one of the first “lightweight methods” which is traced back to 1995. The Scrum method was originally suggested for “managing product development projects” but is today mainly used for software development projects [1].

The Scrum process is based on a few key principles, all somehow related to fast customer feedback, self-organizing teams and constant improvements. The product development team (the Scrum Team) typically consists of around ten members. One member is selected or elected as Scrum Master. The main role of the Scrum Master is to act as an interface between the Scrum Team and any distracting influences, which both implies to keep order in the Scrum Team and ensure focus but also implies to handle contact with, for example, the Product Owner. The Product Owner is a representative of the product to be developed, which can be an external customer or somebody with an interest of the final product [4].

The Scrum Team works in short development cycles, called sprints. Each sprint spans a period of typically a couple of weeks or a month. Each sprint starts with a preparatory task to define the tasks to be undertaken in the sprint and ends with a delivery to the customer or product owner followed by a session called the Sprint Reflection to discuss and propose improvements for next sprint. The product is defined in a Product Backlog (which replaces the traditional documents of requirement specification). The product backlog is agreed upon between the Product Owner and the Scrum Team and typically consists of use cases representing the final product. At each start of a new sprint, a Sprint Backlog is created as a subset of the product backlog in dialogue between the Product Owner, the Scrum Master and the Scrum Team. When defining the Sprint Backlog, emphasis is put upon selecting the features from the product backlog that creates most value for the product owner. During the sprint, daily Scrum meetings are held. These are typically very brief stand-up meetings where each team member answers three questions: what he/she did since yesterday, what he/she plans to do today and finally if the member foresees anything preventing him/her from accomplishing what he/she intend to perform within the foreseeable future [4]. Figure 1 illustrates the key actors, keywords and definitions of Scrum.

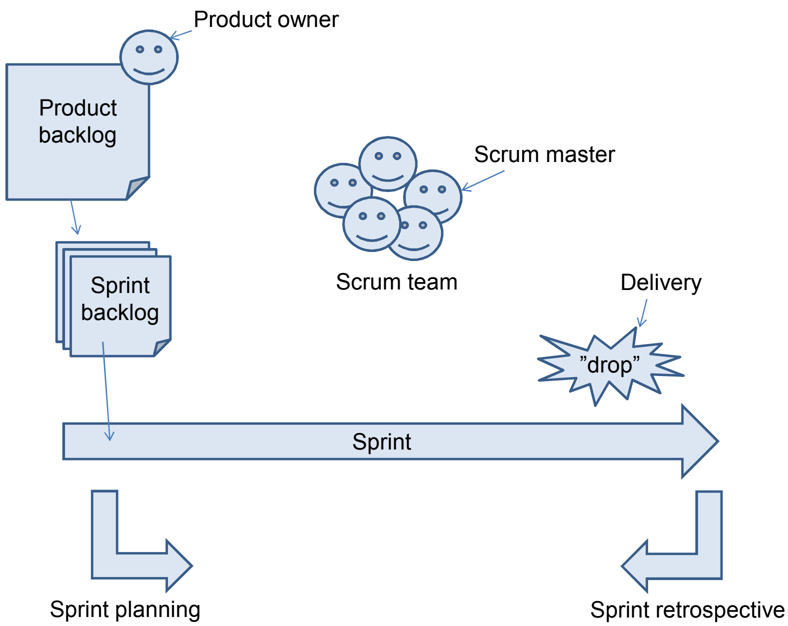

Figure 1. Illustration of Scrum, key actors and definitions

Mechatronics at KTH Royal Institute of Technology

Mechatronics as an academic subject emerged during the late 1960s and the most widely used definition establish mechatronics as “the synergistic integration of mechanical engineering with electronics and intelligent computer control in the design and manufacturing of industrial products and processes” [5]. Mechatronics was introduced at KTH Royal Institute of Technology in 1976 with the first courses offered in the beginning of the 1980s. Currently, Mechatronics is offered both as a masters’ program and as a specialization within four broader engineering masters’ programs (the masters’ programs of mechanical engineering, design and product realization, vehicle engineering and industrial engineering and management). Between 30 and 50 students every year specialize in mechatronics [6, 7, 8], making it the largest producer of mechatronics engineers in Sweden.

The mechatronics masters’ program is a full second cycle program according to the Bologna process [9] consisting of three semesters of courses plus one semester with a master thesis project. The program mainly recruits students with a BSc in mechanical- or electrical engineering. The three semesters of courses conclude with a capstone course spanning 18 ECTS credits (equivalent of 12 weeks of full time studies). This capstone course has been given since the establishment of the mechatronics program in the 1980s and has become popular with Swedish mechatronics industry. Figure 2 shows an illustration of the Mechatronics masters’ program at KTH and the relation between the masters’ program and prerequisites from BSc level. Figure 3 illustrates the capstone course as it is dived between two semesters and split in a preparatory module and a project module.

Between the beginning of mechatronics at KTH, in the 1980s, and now, the subject of mechatronics has shifted from being considered as an "enabling technology" [9] (referring to the introduction of the microcontroller) to a situation where basically everything is mechatronics (as most advanced products are complex and consist of embedded systems). In reality, this means that "mechatronics" now is replaced by "mechatronics product development". This implies an increased focus on mechatronics design methodology and complementary skills such as project organization and management.

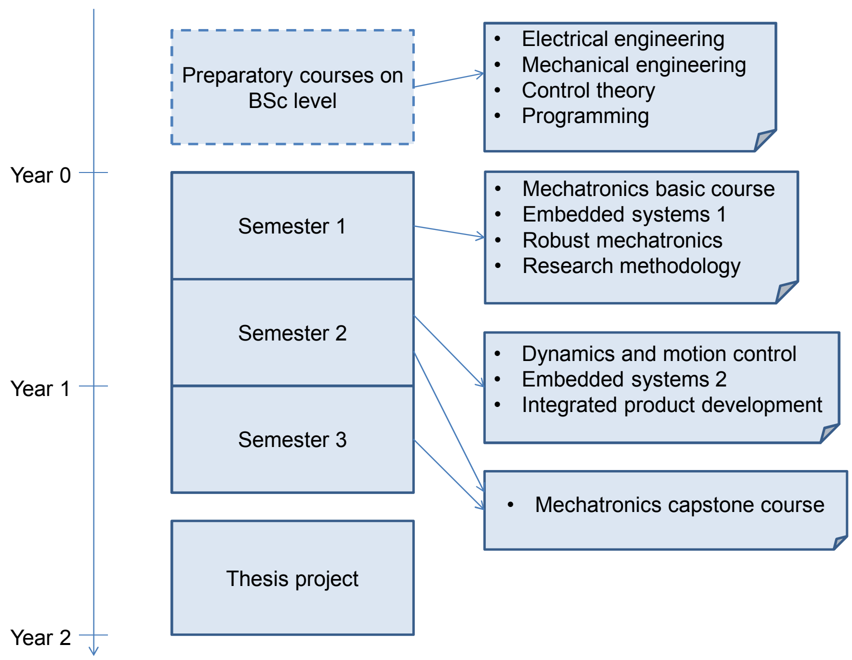

Figure 2. The Mechatronics master program at KTH Royal Institute of Technology

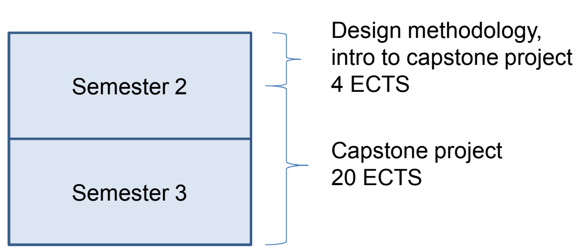

Figure 3. The Mechatronics capstone course with preparatory course module as part of the master's program

Agile methods and Mechatronics product development

After describing agile methods for product development and the academic subject of mechatronics separately, we now turn to the combination of agile methods and mechatronics.

As described in the introduction, Scrum and agile methods is primarily used in software development projects. The main reason is due to the fact that the other technologies in mechatronics product development have considerably longer lead times. Software development typically can have considerably shorter iterative cycles than mechanical product development. It is seldom feasible to create a new mechanical prototype daily or with the same short cycles as new software releases.

From this starting point, two directions now emerge. Let's call them the "easy way" and the "hard way". Mechatronics products do by definition rely on software, or "intelligent computer control". In the history of mechatronics product development, most companies did organize departments etc. according to subject and technology; mechanical engineering department, software, control, electronics etc. Engineers tend to classify themselves according to education (mechanical engineer, control specialists, programmer etc.). The mechatronics partner companies in this and previous studies [6, 7, 8] did have teams of programmers developing the software for the mechatronics products. It is therefore not far-fetched to think that those would benefit from adapting agile methods to their part of the product development. This is also happening as of today, especially in the larger companies. This is also very well and sound, but only if this is considered as software development – and not if considered as mechatronics product development since the holistic approach based on synergy between software, hardware and control hardly can be taken into account. It is rather contradictory to mechatronics design since it implies focusing on one area and reducing disturbances and interference from other parties.

This way, defined earlier as the "easy way", is dangerous since the development team is increasingly split between the software team and the other teams. True mechatronics product development is achieved in synergy between software and hardware, between electrical engineering and mechanical engineering. To organize the software team according to Scrum and creating a self-organizing, independent team might therefore increase distribution and decrease synergy.

The second approach, the "hard way", would instead mean to try to apply agile development methods on the overall mechatronics product development, with a holistic perspective, meaning not by separating software from all other development. To be able to achieve this, a number of obstacles immediately emerge in the horizon. Since agile development methods primarily are used for software development a pitfall in agile mechatronics product development is to do exactly this – to separate software development from the rest. Instead we need to make sure that the core of mechatronics is preserved.

Case study – Mechatronics capstone projects 2011

To investigate this integral approach, an experiment was performed in 2011 and in this section a case study is presented of a capstone course in mechatronics, given at KTH Royal Institute of Technology. In this experiment, Scrum was introduced to five mechatronics student capstone projects running simultaneously.

As described earlier, mechatronics education at KTH has shifted from the previous focus mainly on the specific technical competencies to also include process- and organizational issues, “how to develop mechatronics products”. Historically, student teams and projects mainly have been organized according to the Waterfall model or the V-model, with students acting as project managers organizing the students in hardware teams, software teams etc. Typically, the student teams have been given a project with a corporate sponsor who also provide guidance and participate in steering group meetings at the various stage gates. These capstone projects have thoroughly been described in earlier publications, such as [6, 7, 8].

With the introduction Scrum, it is believed that this can emphasize the core of mechatronics. It is believed that this can be advantageous for the student learning processes, basically by further empowering the student teams. This will be elaborated further upon below.

The Mechatronic capstone projects 2011 involved 37 students, five companies and ten faculty members. The projects were introduced in late March 2011 and the final presentations were held in mid-December 2011. The 37 students were divided into five teams, one team of three students, one of six, two of nine and one of ten students. The two smaller teams were assigned projects from two small companies (one was a university research project disguised as a company) and the three larger teams were assigned projects from larger international companies located in the greater Stockholm area. All five projects related to the development of a stand-alone mechatronic product and were presented in December with a working prototype.

The capstone course spanned over one and a half academic semester. At KTH, each semester is divided into two periods were each period ends with an exam period. A period typically consists of eight weeks of lectures followed by one week of exams. The capstone course spanned three periods with a long summer holiday separating the first and second period. See Figure 4 for details. As shown in the figure each sprint followed the same pattern. The first week and last weeks of each sprint was typically organized by the faculty and had time assigned for work with the sprint backlog, for company presentations and sprint reflections etc.

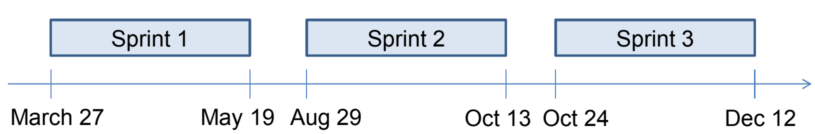

Figure 4. The mechatronics capstone course divided into three sprints

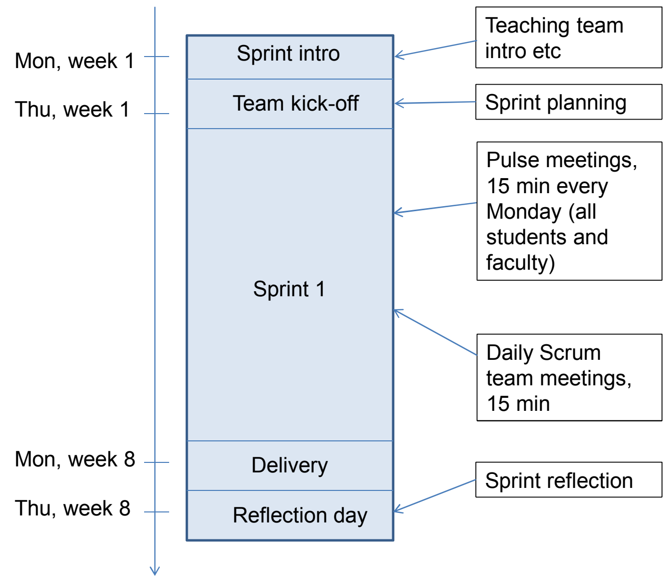

Figure 5. Details of one of the three capstone sprints

Results and findings

In this section, the results and findings of the case study are discussed. First, results from a student questionnaire are presented that gives the student perspective. Secondly, findings from the case study are described with an analysis of their importance for the learning process.

Firstly, three tables are presented that shows the relevance of the course and content in general. In table 1 the students rank this course among the approximately 50 other courses that constitutes their curricula. More than half of the students rank this course as the most or among one of the three most valuable for their future career. Only one student, 3% of the population, rank the course as not being one of the five most valuable.

Table 2 and 3 shows how the students responded to the combination of Scrum and mechatronics product development. In Table 2, when the students respond to the question "did Scrum make it easier or harder to work in the project" most students are mainly indifferent: 41% neutral, 28% said it made life harder and 31% said it made life easier. In Table 3, when the students respond to "do you think Scrum is good for mechatronics development projects in general, a slight overweight is seen toward "yes".

In summary, the three questions from the questionnaire shows that the students indeed considers the capstone course as very important and valuable for their future career, that Scrum did not really make life easier but that they are slightly positive to the notion that Scrum could be good for mechatronics development. The conclusion is that the students had to work hard to learn and adapt to agile product development, that it indeed was not easy to adapt to mechatronics development, but that it can be done and with some benefits. In the following sections, examples from the case study will be presented that further illustrates and motivates this analysis.

Table 1 – Course relevance

This course has been the most valuable to my education so far and future career	16%
This course has been one of the three the most valuable to my education so far and future career	38%
This course has been one of the five the most valuable to my education so far and future career	44%
This course has not really been valuable to my education and future career	3%

Table 2 – Scrum in the capstone project

	Scrum made life much harder				Scrum made life much easier
Did Scrum make it easier or harder for you to work in the project?	9%	19%	41%	25%	6%

Table 3 – Scrum for general mechatronics development

	Definitely not!				Yes – absolutely!
Do you think Scrum is good for mechatronics development projects in general?	9%	16%	31%	34%	9%

Early prototyping

In earlier instances of the mechatronics capstone course, before Scrum was introduced, prototyping was mainly emphasized in the final phase of the course, the final delivery. The main project deliverable for both the company and the faculty consisted of a working prototype and a report documenting the result. Most projects adapted the V-model which typically resulted in large system integration late in the project. As in many such cases, it is hard for inexperienced junior product developer (meaning students) to foresee unknowns and difficulties and plan accordingly, which meant that most system integrations became stressful intensive periods of work around the clock and many last-minute fixes.

In extreme cases, prior to Scrum, a student team could spend the first third of the project to make plans, define work processes and establish procedures. When the faculty firstly turned to agile methods and the Scrum method in particular, it was in a response to the need to get students to prototype earlier – not to reduce the importance of project planning, but to change learning perspective – the motivation was to teach students project planning by doing rather than thinking about planning.

When the capstone course was adapted to Scrum the course deliverables was changed from having students deliver one prototype at the end of the course to three working prototypes, one at the end of each sprint.

In the following, three examples are presented. The idea is not to give deductive evidence or proof of results but to illustrate possible advantages and how these relate to the introduction of Scrum into the mechatronics capstone course.

Example 1 – finding the core value

One of the five teams decided during the first sprint to divide their products into a number of subprojects, with a core “dispensing unit”. Most of these units were complete mechatronics products on their own, containing hardware, software, control etc. In the end, it turned out that the project became immensely more complex than first envisioned by the students and only two of the units were produced in the end. These two units did however represent the core of the product and was very much appreciated by the company/corporate sponsor. The units that were not prioritized were considered more straight-forward for the corporate sponsor and were possible to outsource or finalize in-house by the company. The lesson learnt in this case is that the student team, together with the product owner, defined a subproject which contained the functionality that created the most value for the end product.

In the perspective of the experiences from earlier capstone projects, it is believed that had the team not emphasized the incremental approach the typical student team would have tried to scope the original project further on in the project, very likely spending more time on a system level design than what the time frame allowed. In this case, the student team got the experience of designing and implementing one product completely rather than having been able to finish a much larger and more complex product.

Example 2 – breaking the rules

In this project, the team was given a detailed list of requirements from the company at the start of the project, in the form of a very extensive requirements specification document. This set of requirements included size constraints. In the first sprint, and for the first deliverable, the team decided together with the product owner to produce three mock-ups to verify the size constraints. The team realized early that it was quite hard (if not impossible) to satisfy the size requirements and still achieve the intended purpose and decided to produce three rapid prototypes anyhow. One did satisfy the size requirement but did not satisfy other requirements. The other two satisfied the other performance requirements much better but did break the size requirements.

When presenting these mock-ups to the company the company quickly decided to change the size requirements so that all three prototypes fell within the size boundaries. The advantages gained by breaking the first set of requirements in terms of performance made the company representatives quickly re-think and realizing that the size requirements were not that important after all.

In perspective of earlier capstone projects, the authors experience is that previous student teams would have taken the set of requirements much more seriously and might have missed the opportunity to renegotiate these with the company. The authors do believe that the ability to very quickly show the company a prototype, or a mock-up, enabled the company to rethink the project early enough to change the requirements.

Example 3 – project management

In previous instances of the mechatronics capstone course, a large emphasize has been put upon student project managers; to select, train and support. Typically in a ten person student team, two to three got the opportunity to act as project managers and get experienced in project planning.

When adapting the Scrum method, the task of project planning is very much decentralized by way of visualization and collegiality. In all five projects, Scrum boards were used which illustrated all past, current and future tasks. Scrum emphasizes transparency and democracy.

In one project in particular, the faculty noted a change from previous experiences. One student turned out to be more engaged in project planning than the others. This student expressed quite simply that he was interested in how to break down tasks into subtasks and to try to foresee what tasks would be required in the future. It turned out that the other students in this team very much appreciated this work and relied on the student to perform a lot of the detailed planning for the project. Later on, it turned out that this student often chaired meetings with the faculty and company representatives.

In the perspective of the authors experience from earlier capstone projects, the authors appreciated that a student with a great sense for organization, details and planning did emerge as the de-facto project manager and that the other students acknowledged this. In earlier projects, the faculty most likely would have selected one student who had showed leadership abilities or ambitions (the student in this case did not explicitly express any management- or leadership ambitions) to act as student project leader. The authors believe that this shows another perspective on project management and that with this approach more students might seek and get the chance to practice project organization during the project by focusing more on the practical details of project organization and not on the greater issue of leadership in general.

Conclusions

When combining Scrum with mechatronics product development, as in the studied capstone projects, the emphasize increases on rapid prototyping, quick feedback and incremental development in general. This is in contrast to more traditional mechatronics product development that often follows the Waterfall model or the V-model, with a stronger focus on stage-gate processes and formal procedures when moving from system design to detailed design for example.

In conclusion, the case study presented shows that it indeed is possible to integrate Scrum in a mechatronics capstone course and that this can make the students better prepared for a future career. It is also shown that this prepares the students with a larger flexibility to handle the increased complexity in mechatronics product development and thereby enabling the project teams to deliver results faster, more reliable and with higher quality.

Among the effects of this change is that it is possible to educate mechatronics engineers better capable of organizing mechatronics product development, work in self-organizing teams and to take a greater responsibility for the overall aspects of product development. The difficulties related to marrying Scrum, mechatronics and a capstone course is primarily that it is hard to move away from traditional methods and that Scrum (as with most agile methods) so far is mainly used in the software community.

References

[1] B. Boehm, Get ready for agile methods, with care, Computer, 35 (2002), pp. 64-69.
[2] http://www.agilemanifesto.org/principles.html, 2012-03-11.
[3] C. Larman, V. R. Basili, Iterative and Incremental Development: A Brief History, Computer, 36 (2003), pp. 47-56.
[4] http://www.scrum.org/storage/scrumguides/Scrum_Guide.pdf, 2012-03-11.
[5] F. Harashima, M. Tomizuka, T. Fukuda, Mechatronics-what is it, why, and how? An editorial. IEEE/ASME Transactions on Mechatronics, 1 (1996), pp. 1-4.
[6] M. Grimheden, M. Hanson, Mechatronics - the Evolution of an Academic Discipline in Engineering

Education, Mechatronics 15 (2005), pp. 179-192.
[7] M. Törngren, M. Grimheden, N. Adamsson, Experiences from large embedded systems development projects in education, involving industry and research, ACM SIGBED Review 4 (2007), pp. 55-63
[8] M. Grimheden, Preparing Swedish Mechatronics Engineering Students for a Global Industry, In: Proceedings of the 114th Annual ASEE Conference and Exposition, Honolulu, USA. (2007).
[9] http://en.wikipedia.org/wiki/Bologna_Process, 2012-03-11