INTERNATIONAL CONFERENCE ON ENGINEERING AND PRODUCT DESIGN EDUCATION
8 & 9 SEPTEMBER 2011, CITY UNIVERSITY, LONDON, UK

SPEEDING UP DEVELOPMENT ACTIVITIES IN STUDENT PROJECTS WITH TIME BOXING AND SCRUM

Nis OVESEN, Kaare ERIKSEN and Christian TOLLESTRUP
Department of Architecture, Design & Media Technology, Aalborg University, Denmark

ABSTRACT
This research project investigates how procedures from agile software development can be of benefit to development activities in projects of design engineering students. The agile methods *Scrum* and *Time boxing* are evaluated through a student workshop focusing on near-future concepts for design competitions. Scrum meetings within the student design teams are conducted and video documented each hour throughout the workshop activities as a structured process evaluation tool. Based on a subsequent student survey it is argued that scrum and time boxing are strengthening the focus, communication and awareness of methodical efficiency of the student teams. It is therefore further argued that these methods are both applicable and useful to none-software projects, and that they may correspond to a general tendency of a faster pace in product development within the markets of lifestyle products.

Keywords: Agile development, time boxing, scrum, team communication, double-loop learning

1 INTRODUCTION

This paper describes and evaluates how the agile methods *Time boxing* and *Scrum* are put into use in a compact educational course for design engineering students. Originating from software development, agile methods can be understood as a counter reaction to the age-long and dominating tradition of phased development models. As a subset of iterative methods, agile methods have proven their success in the last years' unstable conditions of the software industry [1], and that is why this research project aims at implementing them into the domain of physical product development. The hypothesis is that agile methods can assist teams in the creative adaption to unpredictable change and project uncertainty:
"In today's fast-paced, fiercely competitive world of commercial new product development, speed and flexibility are essential. Companies are increasingly realizing that the old, sequential approach to developing new products simply won't get the job done." [2]
Now twenty-five years old, the statement from the 1986-edition of Harvard Business Review is more relevant than ever before, as the commercial markets are seemingly moving faster and faster. Adding to this is the fact, that companies are forced to continuously develop new and innovative products in order to survive. Today's design students are tomorrow's employees in this new paradigm, and with respect to building up the students' needed competences in the design educations, new tools for managing speed and flexibility in product development should be introduced.
The initiating question has therefore been: What kind of methods will successfully teach design students to develop products to the fast moving and ever-changing market?
In the search for an answer, agile methods have quickly caught the interest. This present study focuses on agile methods in the early exploration and innovation activities of student project, and this paper concentrates on describing and evaluating the use of *Time boxing* and *Scrum* methods [3], [2] as potential tools in design educations. The study is based on a workshop with design engineering students developing product concepts for a high-uncertainty market, and more specifically, the paper will include the evaluation in respect to applicability of the methods, the workshop result as well as the experiences of both students and teacher.
The rest of the paper is composed as follows. The second section serves as a general introduction to Agile Development and a description of the specific methods used. The third section describes the

methodical setup as well as an overview of the workshop activities. In section four, the outcome from the workshop as well as the gained experiences of both teacher and students are presented. Finally, section five initiates a discussion on the overall learnings from the experiment together and proposes some simple guidelines for implementing scrum and time boxing in similar projects.

2 AGILE DEVELOPMENT EXPLAINED

Agile Development, as a term, was coined early in 2001 during a two-day meeting between seventeen people gathering at Snowbird Ski Resort in the Wasatch Mountains of Utah [4]. The gathered people were representatives from various surfacing disciplines in software development trying to establish a common ground and explicate a united stance in the worldwide software development community. The outcome of the summit in this extraordinary place was *The Manifesto for Agile Software Development*, which, after it's authoring, has had a vastly influential role in the software development community throughout the following ten years. In a sense Agile Development is a response to the traditional – and ultimately failing – software development methods that have been dominating the latter part of the 20th century. Table 1 below shows the value set from The Agile Manifesto. The four statements clearly make up with the command-and-control development processes in traditional development [5].

Table 1. The value set of Agile Development [6]

The Agile Value set
We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value: • **Individuals and interactions** over processes and tools • **Working software** over comprehensive documentation • **Customer collaboration** over contract negotiation • **Responding to change** over following a plan That is, while there is value in the items on the right, we value the items on the left more.

The authoring of the manifesto may be seen as the latest culmination of a decade-long discussion about approaches to development. Highsmith and Cockburn, two of the founding fathers of the manifesto, state the following in an article in 2001: "Traditional approaches assumed that if we just tried hard enough, we could anticipate the complete set of requirements early and reduce cost by eliminating change. Today, eliminating change early means being unresponsive to business conditions – in other words, business failure" [7]. This corresponds especially well with the last of the four values in the manifesto, *responding to change over following a plan*, which also seems to be the primary message of Agile Development.

2.1 Primary characteristics of agile methods

When suggesting de-emphasising the traditional waterfall development model and instead stressing a more dynamic attitude towards changing development conditions, tools need to follow suit. Fortunately, most agile methods answer to exactly this. The umbrella of agile development consists of more than ten separately described methods. It would be going too far to describe them all, thus only the methods of *time boxing* and *scrum* are briefly explained in the following.

2.1.1 Time boxing

In general, agile development is a subset of iterative methods [1] and time boxing is no exception. Time boxing is actually not a specific method, but an often used structural setup that allows for iterative and incremental development in software projects. Whereas a typical iterative development approach is based on delivering functionality in predefined subparts, time boxing is somewhat different as it is the length of the iterations that is fixed and the deliveries that are adjusted to fit the time boxes. This allows a highly structured process, which works well with most agile methods [8].

2.1.2 Scrum

The concept of Scrum was first mentioned in relation to development by Takeuchi & Nonaka as early as in 1986. The term originates from the strategy used in Rugby for getting an out-of-play ball back into play. The name was chosen because of the similarities between this game and product development – both are adaptive, quick, self-organising, and have few rests [9]. With scrum is the emphasis on an empirical process rather than on a defined process [1].

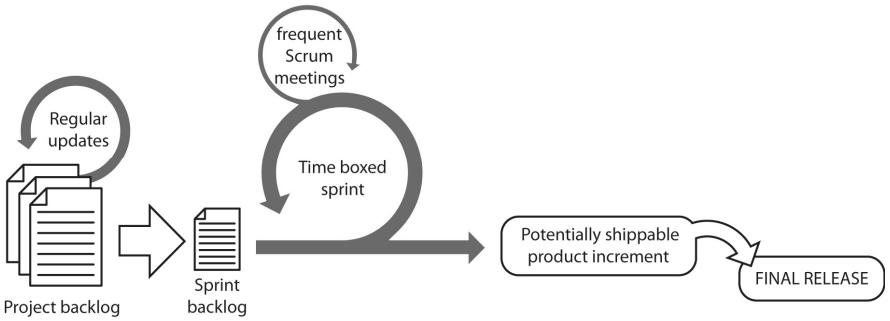

Figure 1. Iterative process with scrum. (Simplified after [10])

Rather than ultimately deciding variables such as requirements, resources, technologies, and tools only at the beginning of a project, the development phase is organised in short iterative cycles called sprints, where these variables are continuously revised and thoroughly controlled. A sprint focuses on the development of only a few collectively chosen features in the product backlog list. Scrum emphasises self-organising teams and most importantly frequent scrum-meetings between all the team members. Each sprint ends with a sprint review and a revision of the backlog, and the development phase ends when the requirements are completed through several sprints-cycles.
Scrum leaves the actual methods in the practical development activities up to the team and is thereby solely a tool for managing the development process rather than an actual development method.

3 METHODICAL SETUP

The research project presented in this paper revolves around a concept development workshop with design engineering students. As most designers may have experienced at some point, the early idea generation and concept development phases of a product development process can be lengthy, frustrating and at times even unpleasant if "Creativity fatigue" and discouragement kick in. This is the origin of the main question here: *Can time boxed scrum activities improve efficiency and communication in teams working with exploration and innovation activities within high-uncertainty areas?* In the following the methodical setup for the workshop is described.

3.1 Workshop setup and program

The two-day workshop "Near-future product concepts" was organised around two recent design competitions. Both competitions asked for product and service concepts for near-future scenarios spanning from 2013 to 2025. By nature, these conditions dictated uncertainty as a foundation of the students' work and thereby trying to mimic the conditions of high-uncertainty markets. The workshop consisted of a lecture each day followed by work organised in small teams of two to four students. The first lecture introduced a recognised idea development model [11] which the students were encouraged to use.
Inspired by the above mentioned agile methods, the teams were asked to work according to a scheme that combined the procedures from time boxing and scrum, but in a rather compact format. The time boxes were cut down to one-hour sprints and in-between the sprints the individual teams had to answer a set of process-focused questions in hourly scrum meetings.
The questions used were adopted from scrum instructions of Larman [1]:
What have you done since the last Scrum meeting?

1. What will you do between now and the next Scrum meeting?
2. What is getting in the way of meeting the iteration goals?
3. Are there any new tasks to add to the process?
4. Have you learned or decided anything new, of relevance to some of the participant? (technical, requirements, etc…)

These directions, here called "Ultra Scrum", were to be followed by teams, and furthermore were the scrum meetings to be held in front of a video camera placed in a corner of the workshop area.
The experiences with this ultra scrum method were collected through an online questionnaire after the workshop.

4 OUTCOME OF THE WORKSHOP

Creating this experimental ultra scrum setup for the workshop has been accompanied with a curiosity for its outcome, and as part of an educational setting, the setup should be evaluated on a number of aspects. Obviously, evaluating the students' experiences is a key aspect, but the influence on the resulting is arguably equally important to evaluate. As the latter element have not been quantified and documented in same way as the student experiences, they are presented as parts of the discussion in section 5. It is important to emphasise that this evaluation is purely based on one single workshop. Ideally, a parallel workshop should have been setup without the agile elements in order to make a thorough comparative analysis, which has not been done. It can also be criticised that the number of workshop participants has been relatively low as a statistical evidence to support eventual conclusions. However, the outcome might present some tendencies valid for the evaluation.

4.1 Student experiences

The workshop counted a total of 15 registered students divided into five groups. The students were asked to individually fill out an online questionnaire after the workshop, and 73% (11 of 15) students did this. The results of the questionnaire may be biased due to the conditions mentioned above. Furthermore, as this workshop is not a part of the ordinary curriculum, it can be argued that the students registering to extra workshops are resourceful above the average, and therefore do not constitute a representative student group. Nevertheless, the questionnaire is briefly summarised below.

4.1.1 Experiences with hourly scrum meetings

The students primarily answered question related to the influence of the scrum meetings, both in respect to their personal experiences and their collective team experiences. When asked about how the scrum meetings influenced the students as individuals, 81% (9 of 11) answered that they found scrum meetings useful as part of a self-reflection process. Furthermore, three students chose to elaborate on the question and answered that scrum meetings made them more aware of ineffective phases as well as their process as a whole. When asked about how scrum affected the teamwork, 91% (10 of 11) found the scrum meetings efficient as reflection upon the just executed tasks, whereas also 64% found the meetings useful as status meetings for the work to come. Finally, according to 72% of the students, the scrum meetings had a positive effect on their efficiency and work speed.

4.1.2 Experiences with video recording as a medium for reporting

During the workshop, the design students were asked to answer the scrum questions to a video camera. It has therefore also been of interest to evaluate this as a possible medium for systematically reporting project progress as a supplement to the ordinary supervision. 91% of the students found the method useful, whereas 9% would rather not use the camera.

4.1.3 Summarising the student experiences

When summarising the students' experiences on the basis of their answers to these and similar questions in the questionnaire, the pattern seems rather clear. The students find Scrum meetings useful to both the teamwork and to themselves as individuals. According to the students, the scrum meetings provide focus and performance awareness as well as strengthening the communication within the team. The video reporting medium was also found useful by the vast majority of the students, and several of them were expressing a desire to use video reporting in other projects. Given the above mentioned preconditions, these conclusions should be taken cautiously, but may nevertheless indicate that scrum and time boxing have a value to non-software product development.

5 DISCUSSION

This paper describes a workshop setup as an experiment in applying procedures from agile software development. The motivation for this has been somewhat twofold: Firstly, the argument of increasingly faster market development suggests a need for new tools for developing lifestyle

products. Secondly, it is argued that idea development phases may provoke "creativity fatigue" and discouragement if too lengthy or stagnant. *Agile* is about dynamics and pace and is therefore chosen for the experiment. The outcome of the workshop is discussed in the following sub-sections

5.1 Formalising a approach to learning

The highly structured procedures of hourly scrum meetings with a certain set of questions enable the students to systematically look both back and forward on the development process and reflect upon the application of the methods used. The steps seem almost archetypical to the fundamental concept of learning and the scrum method can be said to systematically facilitate this behavioural pattern.
Linking the learning process with the design process has been illustrated earlier by others. Schön describes this as *reflection on action* as an overlay to the *reflections in action* [12]. Whereas the students continuously evaluate how specific experiments solve a certain problem in an iterative process, the scrum meetings gives room for critically evaluating the chosen methodical approach. In other words, this allows the students to evaluate whether or not it was the appropriate way to handling the problem. Figure 2 below describes this double-loop learning [13].

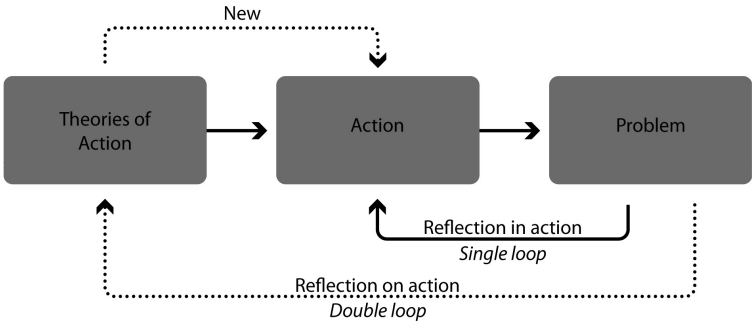

Figure 2. Reflection in action and *Reflection on action* [14]

5.2 Rushing creativity calls for additional methods

It is argued above that the hourly scrum meetings are formalising a structure for reflection on the work process, but one could also argue that the high frequency of scrum meetings creates a sense of urgency within the team. As several students expressed during the workshop, the scrum meetings helped the teams maintain focus on their tasks. On the other hand, stressing a team to be more productive is probably just as destructive to the creative process as is the initially stated problem of creativity fatigue in relation to process stagnation. Some would argue that forcing a higher pace in an idea generation process would be pointless, but it is the experience of this experimental workshop, that exploration activities and concept generation thrive under the conditions of one-hour time boxes and video documented scrum meetings – especially when supplemented with a set of idea development methods to cultivate the actual create practices.

5.3 Setting the right length of the time boxes

This experiment was conducted through a two-day workshop, and for this relatively short period, the length of the time boxes seemed relatively fitting. This high intensity may not be suitable for longer-term projects, thus the length of the time boxes should probably be adjusted according to the full project length. During the course of the workshop, the benefits from the scrum meetings seemed to vary a bit. As the program of day one focussed on idea generation and development, it was rather decision- and concept-intensive, whereas day two was dedicated to "producing and finalising" the concepts. With the shift in activities the intensity of the scrum meetings also changed and became more alike, probably as a result of less need to synchronise and evaluate the work of the group. From day one the students had been successfully managing the hourly scrum meetings themselves with almost no interference from the workshop facilitator, but during the production-focussed activities at day two, the groups were more hesitate and often needed reminders. From this observation, it can be argued that the time boxes should vary in length with the types of activities carried out as shown in figure 3 below. Asking whether or not the length of the time boxes fits the present development activities could then be an additional question to be answered during the scrum meetings.

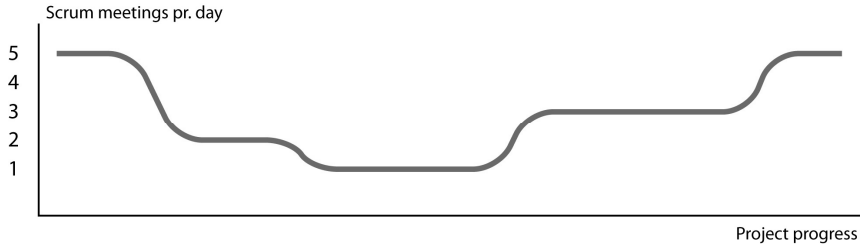

Figure 3. Varying the number of daily scrum meetings throughout a project

5.4 Same solution, different problem?

This evaluation of the applicability of "agile development" to a non-software domain shows that some elements *are* beneficial to adopt. When looking back on the motivation for explicating the Agile Manifesto in 2001, it may have been an attempt to break with some decade-long traditions and values in software development that are not necessarily problematic to physical development. Nevertheless, it is now argued in this paper, that strict time boxing and scrum meetings can strengthen focus, communication and awareness of methodical efficiency within a team developing physical product concepts. It is, however, important to emphasise that the adoption of scrum and time boxing can be faced with challenges when it comes to maintaining the strict pace in scrum cycles throughout the course of a full project. When returning to this paper's initiating statements about a need for new tools to designers in the fast paced market, agile development practices are unlikely to be the whole or only answer, but it might – as proposed in this paper – be *one* way for designers to handle the challenges of increasing speed.

REFERENCES

[1] Larman, C., Agile *and Iterative Development: A Manager's Guide*, 2004 (Addison-Wesley Professional, Boston).
[2] Takeuchi, H and Nonaka, I. The new new product development game, *Harvard Business Review*, 1986, 64(1), 137-146.
[3] Smith, Preston G. *Flexible Product Development: Building Agility for Changing Markets*, 2007 (John Wiley & Sons, California).
[4] *History: The Agile Manifesto*, http://agilemanifesto.org/history.html, 21-02-2011.
[5] Suscheck, C.A. and Ford, R., Jazz Improvisation as a Learning Metaphor for the Scrum Software Development Methodology, in *Software Process Improvement and Practice,* June 2008, 13, pp 439-450.
[6] Highsmith, J., *Manifesto for Agile Software Development*, 2001, agilemanifesto.org, 21-02-2011.
[7] Highsmith, J. & Cockburn, A., Agile Software Development: The Business of Innovation, in *Computer*, September 2001, 34(9) pp120-122.
[8] Jalote, P., Palit, A. and Kurien, P., The Timeboxing Process Model for Iterative Software Development. *Advances in Computers*, 2004, 62, 67-103.
[9] Schwaber, K. And Beedle, M., *Agile Software Development with Scrum*, 2002 (Prentice Hall, New Jersey).
[10] Abrahamsson, P., Salo, O., Ronkainen, J. and Warsta, J., *Agile Software Development Methods – Review and analysis*, 2002 (VTT Publications, Finland).
[11] Striim, O., *Kreativ Problemløsning & Praktisk Idéudvikling 2nd ed.*, 2001 (Merko, Gyldendal Uddannelse, Copenhagen).
[12] Schön, D., *The Reflective Practitioner*, 1983 (Basic Books, New York).
[13] Tollestrup, C., *Value and Vision-based Methodology in Integrated Design*, 2004.
[14] Argyris, C. and Schön, D., *Organizational Learning: a theory of action Perspective*, 1978 (Addison-Wesley).