

ADAPTING SCRUM DEVELOPMENT METHOD FOR THE DEVELOPMENT OF CYBER-PHYSICAL SYSTEMS

Fabian Mulder

Fac. of Industrial Design Engineering
Delft University of Technology
The Netherlands
f.a.mulder@tudelft.nl

Jouke Verlinden

Fac. of Industrial Design Engineering
Delft University of Technology
The Netherlands
j.c.verlinden@tudelft.nl

Tsubasa Maruyama

Dept. of Information Science and Technology
Hokkaido University
Japan

ABSTRACT

At present, the development of Cyber-Physical Systems is governed by methods from embedded systems. However, the complexity of developing human-centered software, hardware, and services is difficult to control. Agile approaches from software engineering to yield fast learning of problem and solution spaces. In particular, Scrum proposes iterative problem solving with short cycles, with a strict development schedule, some tools such as a Scrum board, and a stable team. This paper probes the application of Scrum as agile development method to realize a cyber-physical system with a multi-disciplinary team. A student team with little familiarity in sailing had to develop an embodied sailing simulator in 5 weeks time. Through applying roles, procedures, and tools implied by Scrum, the team could quickly acquire user knowledge to define and verify requirements. One major benefit of the Scrum development was that it stimulated accelerated delivery and focused on weekly cycles or sprints. After each sprint, a part of the product was delivered as a working prototype that was evaluated by users. The final result was a competitive sailing game with an immersive head-mounted display and a custom-made controller including haptic feedback. This game was presented to the sailing coaches and athletes as well as the general public. This application is a case study on the utility of Scrum in multi-disciplinary development processes that encompass CPS. It forced the team to focus on usability and domain knowledge that is iteratively

assessed by the client or focus groups. Furthermore, a Scrum board establishes a physical planning instrument that all team members and other stakeholders can access and understand.

KEYWORDS

Agile methods, cyber-physical systems, scrum, product development, serious gaming

1. INTRODUCTION

At present, little methods and tools are available for the development of Cyber-Physical Systems. The complication of developing software, hardware, and services is difficult to control with respect to the application context. Agile development provides tools and methods for complex problems, in which case many aspects are unknown when a project starts, and a full set of product specifications cannot be developed. The theoretical underpinning of agile development in product design stem from software engineering and cover a wide palette of design paradigms, methods, routines and so forth. As summarized by Dybå and Dingsøy the essence of agile software development is that high-quality adaptive software is developed by small teams using the principles of continuous design improvement and testing based on rapid customer feedback and change [1].

As these methods are mainly used in the development of software, we investigate in this paper if these methods can be used to address some design

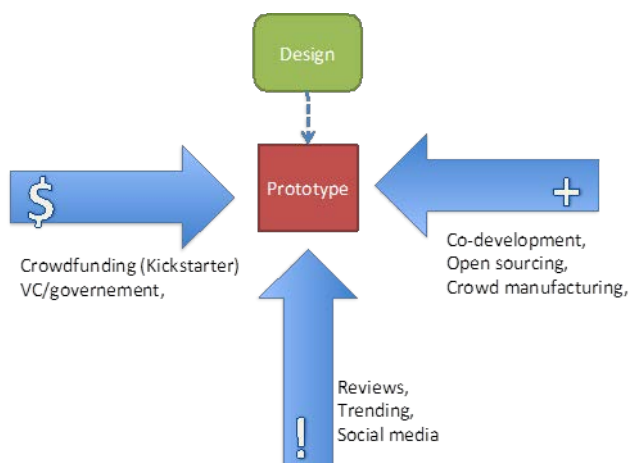


Figure 1 Emerging attention to prototype in product development processes increases the importance of prototypes and early versions

challenges of cyber physical-systems (CPS). Furthermore, the attention of all stakeholders is shifting towards prototypes and early versions (alpha, beta) of the product, less on the product requirements or the design process. Emerging aspects such as crowdfunding, co-development and social media should be incorporated as much as possible to survive competition (Figure 1). Crowdfunding, for example provide companies with the opportunities to directly address end users who can be used to generate user knowledge value to improve the final product. As CPS consider both hardware and software elements, agile methodologies might need to be adapted before they can successfully be used for the development of CPS. Therefore, we want to test in a case study representative of a CPS, if the Scrum method and tools are suitable to be used in the development process.

In this manuscript, we explore the use of agile development methodologies for the development of a cyber-physical system, in particular to a embodied sail simulator. This sail simulator is a complex system involving a variety of sensors, actuators, network- and control algorithms. In the future this simulator will be used for training purposes of beginning sailing athletes. In a case study done within the InnoSportLab sail simulator project at the Delft University of Technology, undergraduate students developed a low cost sailing simulator using a Scrum-approach.

The first section of this paper describes current development methodologies used for the design of CPS. Chapter 3 starts with an introduction to agile

methods for product development in general, followed by a description of the in the case study used methodology Scrum. The paper continues with a description of the developed CPS, followed by the used development procedure. This paper is concluded with the presentation and discussion of the results.

2. DEVELOPING CYBER-PHYSICAL SYSTEMS

As Lee explained, the term CPS refers to a category of systems with integrated computational and physical capabilities that can interact with humans through many new modalities [2-5]. In addition, CPS are dynamic systems which collect information in the physical world through sensors, elaborate the information in the cyber world, and change the physical world through actuators [6,7]. The design of such systems requires understanding the joint dynamics of computers, software, networks, and physical processes [3].

Current methods for the development of CPS are mostly adopted from software and embedded systems development. Most literature focuses on four development methodologies which are: V-model, Model-based development, Component-based development and Platform-based development:

V-Model is a structured methodology which proposes a logical sequence of phases where each phase relies on verification from the previous step before advancing to the next [8]. The v-shaped life cycle of the model describes the activities to be performed and the results that have to be produced during product development. During the development process many testing activities are implemented well before coding. After concluding the requirements generation phase the requirements are fixed and not further iterated. We are considering the development of new CPS where a full set of product specifications cannot be defined. To overcome this issue, the iterative type of problem solving nature of Scrum is proposed, to yield a fast learning of problem and solution spaces. Therefore, it is not wishful to fix the requirements before starting to code.

Model-based development (MBD) is often used by engineering designers and relies on the creation of mathematical- and visual models of the to be designed system. Derler et al. uses a MBD approach to focus on the challenges of modelling a CPS and proposes several technologies to address these

challenges [9]. However the focus is mainly on the technical challenges and not on user and stakeholder involvement during product development. As identified by Gerritsen and Horvath, one of the main obstacles of MBD is the stakeholder involvement problem [10]. Stakeholders are incapable of expressing and encoding their demands to designers and designers cannot communicate their code to the stakeholders due to stakeholders cannot de-codify them.

Component-based development (CBD) focusses on building systems from existing components instead of developing applications from the start. Törngren et al. compared CBD with MDB and concluded that CBD focuses on integrating and reusing software components whereas MDB focusses on the creation of models to analyse, support communication and synthesis during product development [11]. Furthermore, Törngren et al. suggests that an integration of concepts from CBD and MDB is required for the development of complex systems.

Platform- based design (PBD) is based on defining platforms at all of the levels in the design process. Furthermore, the components of a platform are in general partially or completely pre-designed and the upper layer is used to decouple the “application” from the implementation of the platform [12]. Sangiovanni et al. merges PBD design with contract-based design to create a methodology where design requirements are implemented using as much elements from a library of components [13]. Element integration is done based on textual requirements in the form of contracts. This methodology is suitable for complex standard technical systems where the usability is less important.

Furthermore, in literature it is debated what approach CPSs require either an transdisciplinary approach or interdisciplinary, multidisciplinary, transdisciplinary at the same time [14]. According to Horváth and Gerritsen, interdisciplinarity creates a bridge between two knowledge domains (cyber- and physical domain), multidisciplinary involves more than two knowledge domains (for instance physical, biological, engineering and information sciences), and transdisciplinarity extends the knowledge from the various domains towards implementation, and focus on providing application domain independent architectures and technologies to realize the cyber-physical artifacts and services [14]. Assuming a transdisciplinary approach is needed, the self-organizing, multidisciplinary and non-hierarchical

structured aspects of Scrum teams might provide a way to assist collaboration between different domains. Furthermore, involving the clients as Product Owner in the development process adds collaboration at multiple levels, implying a more transdisciplinary approach. In addition, Horváth and Gerritsen pointed out that the traditional separation of computation (software) from physicality (hardware) does not work for CPSs [14]. Instead the physical system platform is complemented by five computing sub-platforms namely networkware, hardware, software, firmware, and knowledgeware sub-platforms.

3. AGILE DEVELOPMENT

Agile methods have as main driver ‘accelerated delivery’ by focusing on small steps, incremental development, prototyping and quick feedback rather than extensive planning and documentation [15].

In the mid 1990 agile software development methods evolved as a reaction against the waterfall-oriented methods, which were characterized by their critics as being heavily regulated, regimented, micromanaged and overly incremental approaches to development.

In 2001 the “Agile manifesto” [16] was written containing 12 principles to define the approach now known as agile software development.

3.1. Scrum

In 1986 Takeuchi and Nonaka [17] studied six technology driven multinationals in the US and Japan that have taken a new approach on managing the product development process. Product development was done on a iterative basis by self-organized cross-functional project teams. This holistic approach showed correspondences with the game of rugby, introducing the term rugby- approach.

Sutherland initiated the first Scrum process in 1993 and worked with Schwaber to formalize Scrum development process [18]. The core elements of this development process are: value prioritization, self-steering teams, joint task definition and continuous improvement. Three aspects need further discussion: roles, procedure, and tools.

Roles

A Scrum team consists of a Scrum Master, Product Owner and the Scrum Team. The Scrum Master has a management role and is responsible for employing the Scrum process to build a system or product [19].

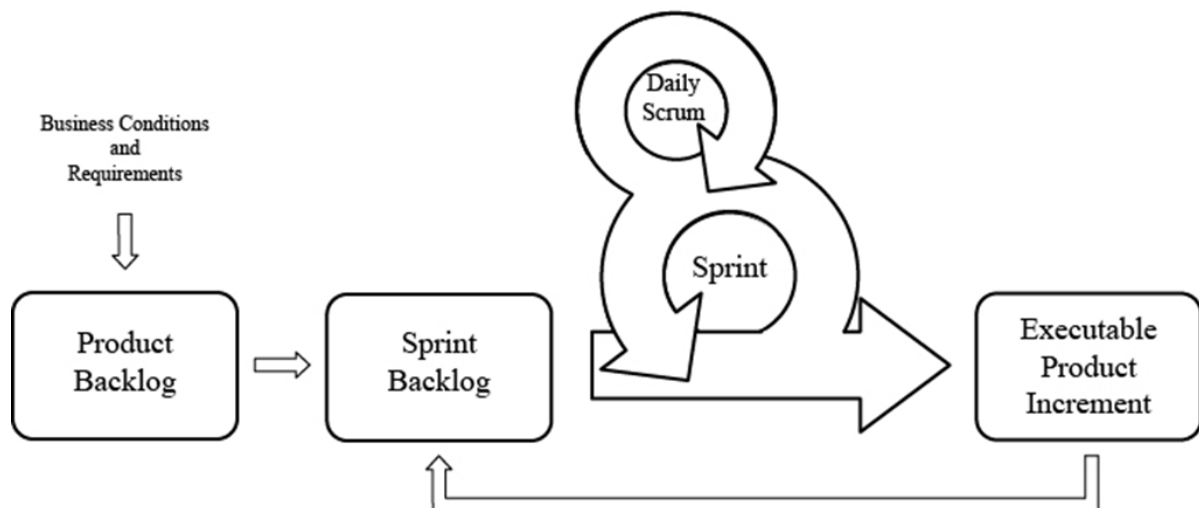


Figure 2 Illustration of the Scrum framework [5].

The Scrum Master is part of the team and ensures that that Scrum values, practices and rules are enacted, facilitates communications with the Product Owner and management.

The Product Owner is officially responsible for the project and determines the business conditions, product requirements and features which are listed in the product backlog. This person manages and updates the product backlog which indicates the features and priorities to the team.

Finally, the Scrum team which is responsible for developing the product features listed in the product backlog. The team works in short development cycles called sprints. Each sprint starts with the definition of a sprint goal and ends with the delivery of a product feature to the Product Owner.

Procedure

Figure 2 illustrates an overview of the Scrum development process. A new project starts with the generation of a prioritized list of all product requirements called the Product Backlog. The Product Backlog is filled with all sorts of content generated by users, marketing, sales and engineering. However, the Product Owner can prioritize items in the Product Backlog and therefore influence the order of the product feature development. The length of a sprint determines the items transferred from the product backlog to the sprint backlog. The Scrum team start the iterative development process which length typically ranges from a couple of weeks to one month, by working on the list of tasks in the sprint backlog. During a sprint the Scrum team meets daily for a 15 minute meeting which is called Daily Scrum.

In this meeting the team explains what was accomplished since the last meeting and what is going to be done for the next meeting including the obstacles there are to achieve this. The sprint is always finished with the delivery of a working prototype or executable product functionality. This deliverable is reviewed during the sprint review by the Product Owner, generating valuable knowledge which is used to update the product and sprint backlog or in the worst case stop developing the feature.

Tools

Scrum makes use of a specific type of task board called the Scrum board (Figure 3). On this board all the sprint backlog items are displayed and sorted on priority. In combination with a burndown chart this board provides the project team with a quick overview of the status of the sprint.

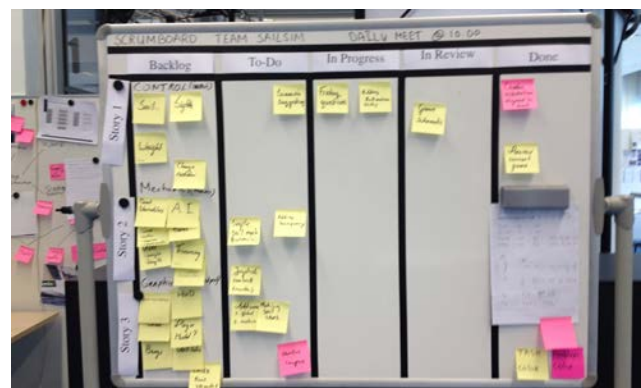


Figure 3 The Scrum board consists of five columns (Backlog, To do, In progress, In review, and Done) indicating the status of an item.

3.2. Limits to applying agile methods

Agile methods as for instance Scrum are still rarely applied in a multi-disciplinary context, and mainly used within the software development. Stelzmann [20] studied the differences between context of agile software and hardware systems by analysing prior work about the context of agile software development and conduct interviews in system developing companies. Stelzmann concluded that “in contrast to software, hardware systems that have to be produced physically often are difficult to be developed in small cyclic steps”. In addition, Stelzmann states that “Only if prototyping, testing, and implementing changes can be done quickly and cheaply, this principle is feasible”.

Due to the advancement of computer support, virtual testing and rapid prototyping, this requirement becomes feasible for many system development projects.

In literature several studies are done on the implementation of agile development in system development projects. Cooke et al. proposed a method to adapt agile development methods for use in complex multi-disciplinary projects [21]. However, no validation of effectiveness of the developed method was done.

Grimheden et al. discusses several studies of multi-disciplinary project teams and conducted a study where agile methods were used in mechatronic education [22]. This study showed positive results as Scrum enabled the students to deliver results faster, more reliable and with higher quality.

Glas and Ziemer researched the challenges of agile methods in the development of complex products in the aviation industry [23]. The authors concluded that agile development is a promising tool to diminish the risk of the development process considerably. In a similar fashion, the Wikispeed project applies Scrum to automotive industry [24]. Through scrum practice combined with open-sourced development and crowdfunding, this project presented a functioning car that from scratch in three months.

A few papers exist describing a proposal to implement agile development methodology for embedded system design. Cordeiro et al. applies agile methodology in a case study for the design of a pulse oximeter [25] whereas Kelly and Keenan investigate the suitability of agile methods for development of home care systems [26]. Both papers do not formulate a clear way to implement and use

agile development for embedded systems. Furthermore, the focus lies on the development of software or mechatronic system without involving the user in the development process.

The discussed literature shows that it is possible to implement elements of agile development methods in mechatronic education, system engineering, aircraft- and automotive industry. However, no clear description is given how agile development methods need to be adapted for the type of product or system. Without a clear description on how to implement Agile development methods and a solution how to cope with industry regulations, most firms find it hard to abandon the traditional development methods. Moreover, the in chapter 2 discussed CPS development methodologies do not provide designers with a method to quickly acquire user knowledge to define and verify requirements. Communication with stakeholders and end users was indicated as difficult because stakeholders do not have the knowledge to de-codify simulation models or code.

Summing up, the following four challenges need to be explored:

- 1) Usability and user satisfaction cover the amalgamation of physical and cognitive ergonomic aspects.
- 2) The manufacturing and maintenance is crucial, especially considering lead times, industry regulations and sustainability aspects.
- 3) Even with intangible products, the attention of all stakeholders is shifting towards prototypes and early versions (alpha, beta) of the product, less on the product requirements or the design process.
- 4) Emerging aspects such as crowd funding, co-development and social media should be incorporated as much as possible to survive competition.

In our opinion the iterative nature of the Scrum development framework with its accelerated delivery aspects might be suitable to overcome these challenges. In addition the close collaboration with stakeholders assist in the process of clearly define requirements.

4. AN EMBODIED SAIL SIMULATOR IN 4 SPRINTS

In this paper, we set up an experiment with a particular CPS application, namely an embodied sail simulator for competitive sailing. It fits within a

running research project, to develop a sail simulator system, to be used for training fleet racing and boat handling skills. Critical in this simulator is its embodied interaction with the user, i.e., the physical coordination of balance, spatial awareness and haptic forces [27]. To address this, the simulator encompasses a mechatronic system, a three-dimensional display, computational simulation, and game mechanics to measure performance.

Due to the complexity of such an embodied system, a design inclusive research approach was used to generate knowledge which one can only get through design [28]. Due to the iterative nature of the development process many prototypes were built to generate knowledge, which could otherwise not be obtained. The principles of continuous design improvement based on user feedback correspond to agile (software) development principles [1].

4.1. Context

A Scrum approach was tested with undergraduate students participating in the fall semester minor program Advanced Prototyping, given at the Delft University of Technology [29]. In a time period of 5 weeks students with different backgrounds work together with researchers to develop a CPS. The participating students were third year bachelor students and had a diverse background not necessarily design or engineering (2 architecture, 1 industrial design and 1 computer science in this case study).

Figure 4 depicts the planning of this project team. In

Table 1 Example Sprint Planning

Date/week	Activities
Monday	- Sprint planning
Tuesday Wednesday Thursday	- 15 minute scrum meetings - Scrum team works on product
Friday	- 15 minute scrum meeting - Scrum team finalizes product prototype - Review new version product - List improvements for new version

the first two weeks, the students got familiar with the assignment and the Scrum methodology. Together with the researchers, the students formed the Scrum team and worked in four weekly sprints as specified in Table 1. Each Friday, the intermediary results were reviewed by the Product Owner. In week 6, the project was concluded by demonstrating the final product at a public exhibit.

4.2. Approach

All elements of the Scrum process were adopted by the Scrum team: a Scrum board was used to visualize the workflow of the sprint and to monitor progress. The first author of this article acted as Scrum Master, facilitating the process, in particular the weekly planning meeting on Monday. The prioritized features from the product backlog are used to decide the sprint planning. During the weekly sprint, the Scrum teams worked on the prioritized features from the product backlog. Each day started with a daily Scrum meeting. Because of the different backgrounds of the students, they were encouraged to

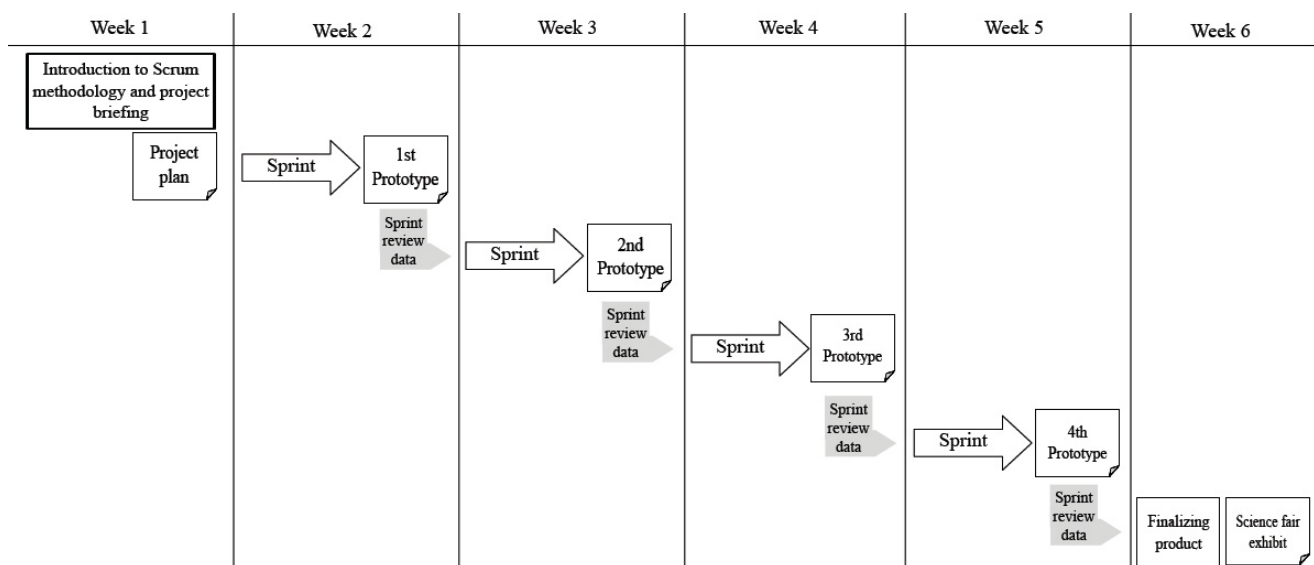


Figure 4 Planning of the Scrum team.

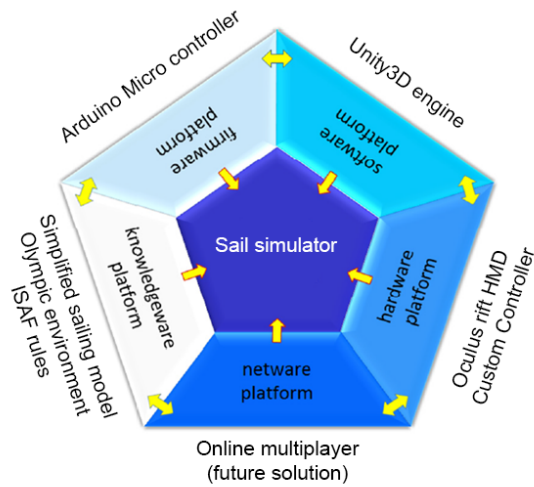


Figure 5 CPS sub-platforms of the embodied sailing simulator.

work in pairs. Each sprint ended with the evaluation of a prototype product (demo). In this review session the developed product was reviewed by sailing athletes and the Product Owner. In a reflection meeting, the knowledge generated during the review session is applied to change or add features to the product backlog.

5. CASE STUDY RESULTS

Figure 5 illustrates the composition platform of Horváth and Gerritsen [14] applied to the final sailing simulator product. The project started with

the assignment to create a low cost simulator for training competitive sailing. Specifically, it would employ a gaming head mounted display, the Oculus Rift. The Oculus provides an extended horizontal field of view of 110 degrees, stereoscopic vision, and head tracking [30]. As software platform, the Unity3D game development system was used. This game engine encompasses a multi-platform rendering and physics engine, with a set of intuitive tools and rapid workflows to create interactive 3D content [31]. It also supports graphics display options and orientation sensing of the HMD system through a drag-and-drop SDK.

These enabling technologies were used by the Scrum team to develop an embodied simulator without the need of extensive of programming skills.

In six weeks, four sprint cycles were completed. Table 2 provides an overview of the intermediate results for each sprint, split into test focus, overall goal of the simulator, used hardware, sail boat mechanics (implementation in the game engine) and additional software features. The test focus of a sprint were represented as separate items on the scrum board, the amount of sub problems are specified between parentheses.

The final design consisted of sailing simulation software in combination with new hardware in the shape of a physical controller shown in Figure 6. The custom made physical controller provided the interaction between the simulator and user, and communicated with the software by a Arduino

Table 2 Overview of product features tested during each sprint

Sprint	1 st	2 nd	3 rd	4 th
Test focus (Identified problems)	- Simulation purpose (2) - Sailing scenario (2) - Oculus Rift (2)	- Revised scenario (3) - Boat controls (3) - Boat mechanics (4) - Visualization (5)	- Revised boat mechanics (2) - Revised boat controls (2) - Revised scenario (5) - Revised visualization (4)	- Physical controller (1) - Visual feedbacks (2) - Game mechanics (1)
Simulation goal	Training for advanced sailing athlete	Training for beginners		
Used hardware	Game pad and Oculus Rift	Controller prototype and Oculus Rift		Controller haptic feedback prototype and Oculus Rift
Sailboat mechanics	As a motor boat	Sailing with sail (Constant speed)	Improved sailing realism (Speed data from speed table including sail efficiency)	
Software features	Island environment model with simplified sailing track	Rio de Janeiro terrain model with realistic Ocean movement	Adding wind effect and visual feedbacks for training	Adding Olympic sail track with buoy model

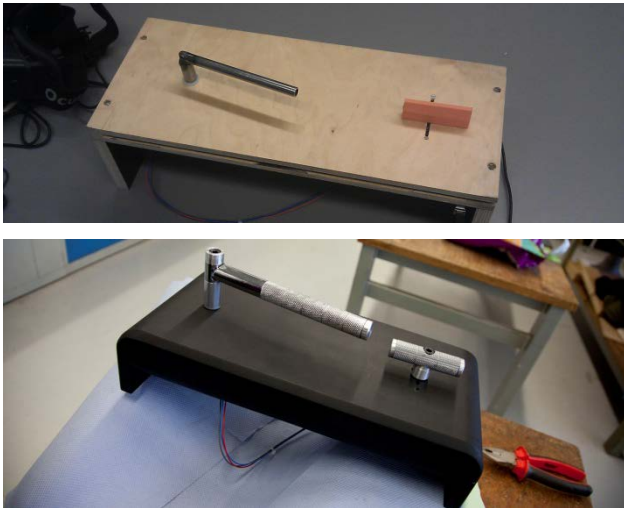


Figure 6 Two iterations of the developed controller (top: sprint 2, bottom: sprint 4).

microcontroller [32] with a synchronized serial connection. The controller provided the user with haptic feedback to provide sailors with a realistic sailing experience.

To enhance the visual effects the software contained realistic modelled environment, sailing dinghy model, and sailing tracks, shown in Figure 7. The Unity Ocean Forum provided the Scrum team with an advanced Ocean Shader asset. This software component allowed quick implementation of boat movement and buoyancy to the sailing software.

The sailboat dynamics were implemented after testing several different mathematic models with the Product Owner. The final model was based on a by athletes officially used speed table containing ideal boat speed data depending on wind speed and the angle between boat and wind. To enhance the learning effect of the user the software provides visual feedback by showing current boat speed and sail efficiency.

6. CRITICAL REFLECTION

The first author closely followed the process as Scrum Master. Additional results were collected from interviews, recordings of the Scrum board items, and a debriefing questionnaire (Appendix). The key responses from the latter are summarized in Table 3

6.1. Interdisciplinary team learning

The students started with no experience of the Scrum development methods and many of them were new to

the sailing sport. In the first week the students were introduced to Scrum by the Scrum Master.

Overall comments on the Scrum development process by the students were positive and in the questionnaire (appendix) they clearly indicated that the Scrum development process made the project easier. Some students had experience with the water-fall development approach and favoured the fact that Scrum is a less documentation-heavy process. In addition, predicting future problems and requirements to plan accordingly would have been difficult due to the lack of knowledge about sailing and software development. Unexplored system as this low cost sail simulator, which use emerging technologies benefit from the Scrum approach. It allowed the development team to generate human-centred insights combined with novel cyber-physical principles in an early phase of the development project dealing with their inexperience in the topic.

6.2. Roles

Selecting members of the Scrum team and defining their roles including project leader is more important than expected. The lack of a project leader in the Scrum team affected the team dynamics and performance. Students with a less active attitude waited too long before stepping up and take over the backlog items from students who were ill. A good project leader would keep better track off the progress made and take responsibility when changes in the planning had to be made. Students were selected on their background and study and less on their skills.

6.3. Procedures

The procedures of the Scrum development could be implemented in the low cost simulator project with

Table 3 Selected items from the questionnaire (n=5)

Question (1= Disagree, 5= Agree)	Average
The length of the sprint was too short for me	2.6
The 15 minute daily Scrum was not used optimally	3.8
The review sessions did provide us with knowledge we otherwise could not find	4.8
The Scrum board was a good planning tool	4.4

variable success. The next sections discuss the observations by the Scrum Master and the student perspective collected from the interviews and questionnaire.

Weekly sprint

In this project, the students reflected that sprints of one week were suitable for this student project. As this project also required hardware development, a strict time planning was required due to the fact that prototyping took more time. Rapid prototyping techniques allowed faster prototyping nevertheless most students preferred more time to construct the hardware elements of the product. However, one student suggested to shorten sprints in the first phase of a project with many review sessions to quickly generate knowledge value on the features and direction of the to be developed product while using longer sprints at later stages of a project to build these features.

Daily Scrum

Questionnaire results showed that the daily Scrum meeting was not optimally used in this project. The Scrum team members had different schedules as a result of different courses. Although the time of the daily meeting was scheduled to overcome this problem some students lacked the work ethics to show up on time for this meeting. As a result team members did not share their progresses and did not know who to address for questions on a developed feature. A solution might be to manage process and problems by using a short daily report through the web-system or use an online Scrum board with more detailed information to each task, which can be accessed anywhere.

Review sessions

Figure 8 illustrates a review session. All students stated that the review session changed their view of the product completely. This weekly confrontation with the Product Owner and athletes provided a solid understanding of competitive sailing they otherwise could not have found. However, to maximize the results of the review sessions, it was suggested to use a detailed test planning and to schedule internal review sessions to efficiently use the valuable review time.

The rules of Scrum prescribe that there is only one Product Owner who is responsible for the product backlog and reviewing the product [33]. However, in in this experiment, prospective users (experienced

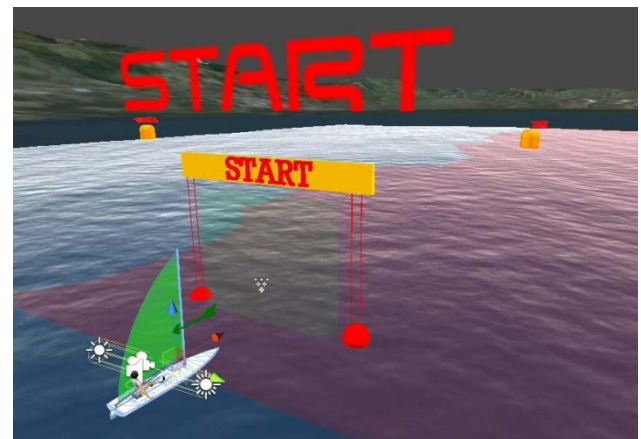


Figure 7 Screenshot of the final sailing software.

sailors) accompanied the Product Owner during reviews. This enabled the consideration of different types of sailors during evaluation of the intermediate results.

6.4. Tools

The main tool used during this case study was the Scrum board. All team members throughout the sprint quickly accepted the use of the Scrum board as a planning tool. However, the questionnaire indicated mixed results on the information it should indicate. The students indicated that it was hard to see who was working on which feature or to find information on an already accepted feature. This lead to problems when team members were unavailable due to illness. The lack of documentation made it different for the remaining team members to finish these features. Using an online Scrum board might improve information sharing in such occasions.

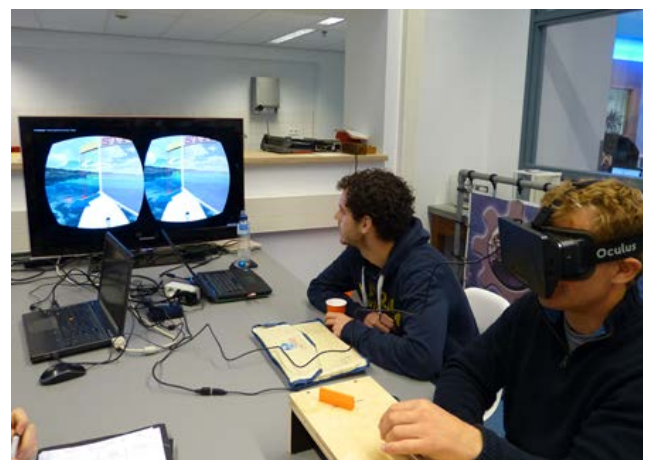


Figure 8 A Scrum review session with the end user and Product Owner.

7. CONCLUSIONS AND RECOMMENDATIONS

The Scrum approach is often used in multidisciplinary project teams and might be useful to address design challenges of CPS. We applied it to the development of an embodied sail simulator, including hardware, software and sensors. Applying Scrum helped to quickly generate user knowledge value and to define and verify requirements. In only five weeks time, a product was ready to present to the public.

7.1. Conclusions

The scale of the case study in this paper is too small to address all the challenges identified in chapter 3. However, it provided an indication in which direction solution can be expected.

1) Usability and user satisfaction cover the amalgamation of physical and cognitive ergonomic aspects.

When requirements for CPS are unknown Scrum can help to quickly generate user knowledge value and requirements. The review sessions clearly aided in the challenge to increase usability and user satisfaction which cover the amalgamation of physical and cognitive ergonomic aspects.

2) The manufacturing and maintenance is crucial, especially considering lead times, and sustainability aspects.

In this project software and hardware tools were selected which are used by a large community. The major benefit of this was that the Scrum team had access to knowledge, assets and code made available by members of the community allowing quicker prototype development. Furthermore, depending of the type of product users can contribute to the development process when using open source hard- and software allowing continues improvement.

3) Even with intangible products, the attention of all stakeholders is shifting towards prototypes and early versions (alpha, beta) of the product, less on the product requirements or the design process.

One major benefit of the Scrum development is that it stimulates accelerated delivery and focuses on incremental product development. After each sprint, a part of the product is ready in the form of a prototype which is testable by users. In this case study a basic low cost simulator was ready after three sprints. This points out one major area of interest,

namely when a product should be introduced to the market. Different from software is that CPS contain not only software but also hardware which make updates more difficult.

4) Emerging aspects such as crowd funding, co-development and social media should be incorporated as much as possible to survive competition.

To make a project successful a dedicated Product Owner is required. However, emerging aspects such as crowd funding, give end-users a more important role in the development process. Companies might benefit by this by using these so-called ‘backers’ as source of information to assist Product Owners in the review process.

7.2. Recommendations

For future research on how to implement Scrum development methods for CPS development we recommend:

- In this case study we encountered that team cohesion is only achieved when all members followed the Scrum procedures: the daily scrum was not optimally used causing knowledge loss when certain team members were unavailable due to illness.
- One of the major drawbacks of using Scrum is the lack of documentation which increases the risk of requirements mismatch and knowledge loss. We suggest to use a Scrum board with more detailed feature descriptions and to implement a checklist to determine when a feature is ready (clear end goal).
- After each review session a document which summarizes current and updated product requirements should be used to update product backlogs and share the requirements and goals with reviewer, Product Owner and the Scrum team.
- The length of the sprint should be selected based on the experience, available knowledge and motivation of the Scrum team.

ACKNOWLEDGMENTS

This work has been co-financed by the European Regional Development Fund (ERDF) for the project InnoSportLab Den Haag: Innovatie in Zeilen. The authors would like to thank all the participants in the review sessions for their cooperation.

APPENDIX – DEBRIEFING

QUESTIONNAIRE RESULTS

Participant scores (1= disagree, 5=agree).						
	Questions	P1	P2	P3	P4	P5
1	The assignment was manageable in 5 weeks	3	4	4	3	5
2	Working with unity was too much of a challenge for me	4	4	1	3	3
3	Working with the oculus rift was a nice feature of the assignment	3	5	4	4	4
4	Working with Scrum made the project easier	5	4	5	3	3
5	The assignment challenged me to learn new aspects not related to my own study (f.i. programming, concept design, prototyping etc)	4	3	5	5	3
6	Some of these new aspects I learned from my fellow team members	4	3	4	4	4
7	I think Scrum can be interesting for <u>non-software</u> development projects.	3	5	3	4	5
8	I think I might be using Scrum for one of my future projects	2	5	5	2	3
9	I am satisfied with the final product we presented at the science fair	4	5	5	5	4
10	The workload during the project was divided equally	3	2	3	4	2
11	Working in short sprints works well for me	4	2	5	2	4
12	The length of the sprint was too short for me (more time needed)	3	2	1	4	3
13	During the sprint I always knew what I had to do	2	2	5	4	4
14	During the sprint I preferred working at home	1	2	1	5	2
15	The focus during the sprint should be on only one part of the product	2	2	3	2	2
16	The Scrum board was a good planning tool	4	4	5	5	4
17	I never encounter a situation where I did not know what to do next	4	2	4	4	2
18	Features should be grouped in only one post it with less detail	3	2	2	2	4
19	The Post-its on the Scrum board should contain more details	3	4	5	3	2
20	The 15 minute daily scrum was not used optimally	5	4	4	3	3
21	The daily Scrum should be longer	2	2	1	3	2
22	The in the sprint developed prototypes were sufficient to review with users	4	4	4	2	3
23	The in the sprint developed prototypes should have been more detailed	3	2	3	4	4
24	The review sessions did provide us with knowledge we otherwise could not find	5	4	5	5	5
25	The review session changed our view of the product completely	4	4	4	4	5
26	The review sessions should be with more clients	3	3	2	4	3

REFERENCES

- [1] Dybå, T., and Dingsøyr, T., (2008), "Empirical studies of agile software development: A systematic review", *Information and software technology*, Vol. 50 (9), pp. 833-859.
- [2] Lee, E.A., (2006), "Cyber-physical systems-are computing foundations adequate", *Proceedings of the Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, Vol. 2, Citeseer.
- [3] Lee, E.A., and Seshia, S.A., (2011), *Introduction to embedded systems: a cyber-physical systems approach*, Lee & Seshia.
- [4] Shi, J., Wan, J., Yan, H., and Suo, H., (2011), "A survey of cyber-physical systems", *Proceedings of the Wireless Communications and Signal Processing (WCSP), 2011 International Conference on*, IEEE, pp. 1-6.
- [5] Rajkumar, R.R., Lee, I., Sha, L., and Stankovic, J., (2010), "Cyber-physical systems: the next computing revolution", *Proceedings of the Proceedings of the 47th Design Automation Conference*, ACM, pp. 731-736.
- [6] Klein, R., Xie, J., and Usov, A., (2011), "Complex events and actions to control cyber-physical systems", *Proceedings of the Proceedings of the 5th ACM international conference on Distributed event-based system*, ACM, pp. 29-38.
- [7] Conti, M., Das, S.K., Bisdikian, C., Kumar, M., Ni, L.M., Passarella, A., Roussos, G., Tröster, G., Tsudik, G., and Zambonelli, F., (2012), "Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber-physical convergence", *Pervasive and Mobile Computing*, Vol. 8 (1), pp. 2-21.
- [8] DEUTSCHLAND, B., (2004), "Part 1: Fundamentals of the V-Modell", www.v-model-xt.de, 2004.
- [9] Derler, P., Lee, E.A., and Vincentelli, A.S., (2012), "Modeling Cyber-Physical Systems", *Proceedings of the IEEE*, Vol. 100 (1), pp. 13-28.
- [10] Gerritsen, B.H., and Horváth, I., (2012), "CURRENT DRIVERS AND OBSTACLES OF SYNERGY IN CYBER-PHYSICAL SYSTEMS DESIGN", *Proceedings of the ASME 2012 International Design Engineering Technical Conferences*, Chicago.
- [11] Torngren, M., Chen, D., and Crnkovic, I., (2005), "Component-based vs. model-based development: a comparison in the context of vehicular embedded systems", *Proceedings of the Software Engineering and Advanced Applications*, 2005. 31st EUROMICRO Conference on, IEEE, pp. 432-440.
- [12] Sangiovanni-Vincentelli, A., (2002), "Defining platform-based design", *EEDesign of EETimes*.
- [13] Sangiovanni-Vincentelli, A., Damm, W., and Passerone, R., (2012), "Taming Dr. Frankenstein: Contract-Based Design for Cyber-Physical Systems*", *European Journal of Control*, Vol. 18 (3), pp. 217-238.
- [14] Horváth, I., and Gerritsen, B., (2012), "Cyber-physical systems: Concepts, technologies and implementation principles", *Proceedings of the Proceedings of the Tools and Methods of Competitive Engineering Symposium-TMCE*, pp. 19-36.
- [15] Cohn, M., (2009), *Succeeding with agile: software development using Scrum*, Pearson Education.
- [16] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., and Jeffries, R., (2001), "The agile manifesto", *The agile alliance*, Vol. 200 (1).
- [17] Takeuchi, H., and Nonaka, I., (1986), "The new new product development game", *Harvard business review*, Vol. 64 (1), pp. 137-146.
- [18] Schwaber, K., (1997), "Scrum development process", in: *Business Object Design and Implementation*, Springer, pp. 117-134.
- [19] Schwaber, K., and Beedle, M., (2002), *Agile software development with Scrum*, Prentice Hall Upper Saddle River.
- [20] Stelzmann, E., (2012), "Contextualizing agile systems engineering", *Aerospace and Electronic Systems Magazine*, IEEE, Vol. 27 (5), pp. 17-22.
- [21] Cooke, A., Bonnema, G.M., and Poelman, W., (2012), "Agile development for a multi-disciplinary bicycle stability test bench".
- [22] Edin Grimheden, M., (2013), "Can agile methods enhance mechatronics design education?", *Mechatronics*.
- [23] Glas, M., and Ziemer, S., (2009), "Challenges for agile development of large systems in the aviation industry", *Proceedings of the Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, ACM, pp. 901-908.
- [24] Denning, S., (2012), "How Agile can transform manufacturing: the case of Wikispeed", *Strategy & Leadership*, Vol. 40 (6), pp. 22-28.
- [25] Cordeiro, L., Barreto, R., Barcelos, R., Oliveira, M., Lucena, V., and Maciel, P., (2007), "Agile development methodology for embedded systems: A platform-based design approach", *Proceedings of the Engineering of Computer-Based Systems*, 2007.

ECBS'07. 14th Annual IEEE International Conference and Workshops on the, IEEE, pp. 195-202.

- [26] Kelly, S., and Keenan, F., (2010), "Investigating the Suitability of Agile Methods for Requirements Development of Home Care Systems", JSEA, Vol. 3 (9), pp. 890-893.
- [27] Mulder, F.A., Verlinden, J.C., and Dukalski, R.R., (2012), "The effect of motion on presence during virtual sailing for advanced training".
- [28] Horvath, I., (2008), "Differences between'research in design context'and'design inclusive research'in the domain of industrial design engineering", Journal of Design Research, Vol. 7 (1), pp. 61-83.
- [29] Geraedts, J., Doubrovski, E., Verlinden, J., and Stellingwerff, M., (2012), "Three Views on Additive Manufacturing: Business, Research, and Education", Proceedings of the, TMCE.
- [30] Oculus-VR, (2013), "Oculus Rift information", <http://www.oculusvr.com>, 2013.
- [31] Unity3D, (2013), "Unity 3D platform information", <http://www.unity3d.com>, 2013.
- [32] Arduino, (2013), "Arduino information", <http://www.arduino.cc>, 2013.
- [33] Scrum.org, (2013), "The scrum guide: The Definitive Guide to Scrum", <http://www.scrum.org/Scrum-Guides>, 2013.

