

PRAGMATIC TEAM COMPOSITIONS IN SCRUM-BASED DEVELOPMENT PROJECTS

Ovesen, Nis
Aalborg University, Denmark

Abstract

Agile Development techniques have become the industry standard in Software Development in the pursuit for better performance. The popularity of these techniques and methods has caused them to slowly spread to other domains. Thus, the aim of this paper is to identify how some of these Agile Development techniques influence the team organisation, when implemented in the development environments of traditional product development companies.

A case study of seven companies has been carried out in order to identify and analyse the challenges of composing teams in development environments where the agile process framework called Scrum is implemented.

The case study has resulted in the identification of five specific challenges in regard to team composition when implementing Scrum in integrated development environments. The challenges are analysed and discussed and the team organisation of the respective companies are furthermore presented.

Keywords: Integrated product development, Organisation of product development, Agile Development, Scrum

Contact:

Nis Ovesen
Aalborg University
Department of Architecture, Design and Media Technology
Denmark
nove@create.aau.dk

Please cite this paper as:

Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

Faster, Better, Faster, More! Performance within the field of Product Design and Development has become the new agenda in most modern companies operating in a global market. Traditionally, the discussion about performance has been centred around manufacturing systems, partly caused by the arrival of concepts like Lean Manufacturing (Krafcik, 1988). But an increasing awareness of New Product Development activities as a significant factor in achieving strategic and financial goals has since then broadened the performance discussion to also include the development processes (Hertenstein and Platt, 2000). This research paper concentrates on the development activities prior to manufacturing and investigates how a group of companies organise their development teams in order to improve performance.

In the software development industry similar performance requirements have been the norm for the last couple of decades and development techniques coined under the term Agile Development have become industry standards when addressing the challenges of increasing development speed, market change and product complexity.

Struggling with the same kind of challenges, the group of companies – all within the domain of integrated product development – have adopted certain agile initiatives in their development environments. Thus, the aim of this paper is to identify how these agile initiatives have affected the team organisation in the development environments and to unfold the challenges, which appear as a result of this.

A series of case studies have been carried out in seven Danish companies using the agile process framework called Scrum (Schwaber, 2004), which have become an industry standard and best practice in software development in the recent years. To some extent, the iterative and incremental process of Scrum resembles the creative process of product developers. However, the study shows that the implementation of Scrum in integrated product development environments is not without its difficulties despite this fact.

As this paper attempts to uncover some of these aspects, the rest of it is composed as follows: The section following immediately after this introduction is *Background and Research Methods* and has a twofold focus: It will contain descriptions of both the method used to complete this research *and* the Scrum framework, which serves as a background for the research project. The subsequent section will present the results of the data collection effort and is followed by a section focused on analysing the results. Finally, a section dedicated to concluding and discussing the insights that have been revealed as a result of the research effort ends the paper.

2 BACKGROUND AND RESEARCH METHODS

As mentioned, this section comprises of two parts. Firstly, it includes a presentation of the background of the Scrum framework, which include a short overview of the values of Agile Development. Secondly, the methodological approach in this research project is briefly outlined.

2.1 Agile Development and the Scrum framework

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 (Highsmith, 2001a). 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 its authoring, has had a vastly influential role in the software development community throughout the following years. The excerpt in Table 1 below shows the value set from The Agile Manifesto of Software Development. The four statements clearly make up with the command-and-control development processes in traditional development (Suscheck and Ford, 2008).

Table 1. The value set of Agile Development (Highsmith, 2001b)

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.

As a supplement to the values above, the Agile Manifesto also contains a series of 12 principles to follow (Highsmith, 2001b). The principles related to team composition and team dynamics are:

- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- The best architectures, requirements, and designs emerge from self-organizing teams.

Scrum is a process framework that fits into the agile dogmas and was first mentioned by Takeuchi & Nonaka in relation to development 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 Schwaber and Beedle, 2002). In comparison to a traditional organisation of development teams, Agile Development and the Scrum framework put much emphasis on the dynamics and composition of teams, and just as Agile Development proposes the values and principles above, so does Scrum prescribe a series of guidelines about development teams. The present state of Scrum is formulated by Jeff Sutherland and Ken Schwaber (2013) and includes the following:

- Development Teams are self-organizing.
- Development Teams are cross-functional.
- Scrum recognizes no titles for Development Team members other than Developer.
- Scrum recognizes no sub-teams in the Development Team, regardless of particular domains.
- Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole.
- Optimal Development Team size is small enough to remain nimble and large enough to complete significant work. More specifically between 3 and 9 team members.

Apart from the Development Team, Scrum also consists two other supporting roles: The Product Owner and the Scrum Master, which relate to the Development Team in different ways. Scrum in general is regarded as an iterative and incrementally oriented process framework and often depicted as in Figure 1 below.

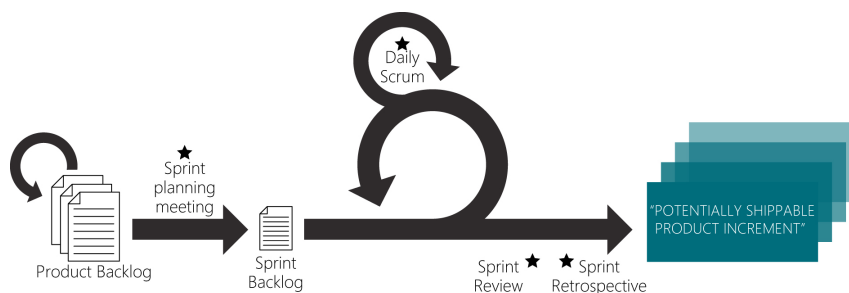


Figure 1. Scrum framework with circular arrow representing a 2-4 week development cycle.

Rather than ultimately deciding variables such as requirements, resources, technologies, and tools only at the beginning of a project, the development phase in Scrum 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 prioritised and collectively chosen features. As stated above, Scrum emphasises self-organising teams and most importantly frequent scrum-meetings between all the Development Team members. Each sprint ends with a Sprint Review and a revision of the development plan, and the development phase ends when all requirements are completed through several Sprints-cycles.

2.2 Research Methods

The objective in this research project is to identify the team composition challenges of implementing Scrum in integrated product development environments. In order to fulfil this objective, a number of companies had to be found. Thus, A thorough search through networks and online discussion groups resulted in a list of seven companies with varying experience with Scrum, who were all willing to take part in the research project.

All the seven companies, which are regarded as separate cases, have been investigated through a series of interviews with 1 to 3 employees. All interviews have been conducted as semi-structured interviews (Kvale and Brinkmann, 2009) with the assistance of an interview guide, and all the interviews have been documented by recorded audio. Table 2 below presents an overview of the seven cases and their respective characteristics.

Table 2. Case overview and characteristics

Case number	Scrum experience	Primary disciplines	Team organisation	Physical organisation
A	4 years	- Software - Hardware - Mechanical - Industrial design	Cross-disciplinary team	Co-located teams
B	1 month	- Software - Hardware - Mechanical	Cross-disciplinary team	Dispersed teams
C	10 months	- Software - Firmware - Hardware - Mechanical	Cross-disciplinary team + Software team	Co-located teams
D	3 years	- Firmware - Hardware - Mechanical	Firmware team + Hardware team + Mechanical team	Co-located teams
E	1,5 years	- Software - Firmware - Hardware - Mechanical - Industrial design	Software team + Firmware/hardware team + Mechanical team (Incl. Industrial design)	Co-located teams
F	1,5 years	- Firmware - Hardware - Mechanical - Industrial design - User Experience	Cross-disciplinary team + (software team as sub-supplier)	Co-located teams
G	6 months	- Software - Hardware - Mechanical	Cross-functional teams <i>or</i> Functional teams (depending on project)	Co-located teams <i>or</i> dispersed teams

The chosen cases range from development of consumer products to industrial components and healthcare equipment. Furthermore, all case companies have implemented Scrum into their existing and respective variations of Stage-Gate models. From the cases a list of team composition challenges have been identified. These are together with the organisational layouts of the companies presented in the following section.

3 RESULTS

Seven cases have been investigated. All of them have each in their own way attempted to organise Development Teams in accordance to the guidelines of Scrum, but also in respect to the existing organisations and their traditions. This means that all case companies have adapted the Scrum framework in slightly different ways and that all of them have different setups, and as it is seen in Figure 2 (next page). However, all the development environments in the cases are familiar with the above-mentioned guidelines from Scrum and have taken these into consideration in their organisational setups.

3.1 Team compositions and Distribution of Roles

In general, six different team compositions are represented in the seven cases. These range from one single cross-functional team in the most basic setup to more complicated combinations of cross-functional and functional teams working together on a single development project. Naturally, this highly depends of the complexity of the project. Whereas some projects are relatively simple and manageable for a cross-functional team of 7 or 9 developers, others are vastly complex and require certain combinations of domain specific expertise. In that type of cases the team composition tends to include functional teams.

The supporting roles of Scrum Master and Product Owner are both present, but in very different setups. All case companies have Scrum Masters placed *in* or *between* the Development Teams. Often the role is borne by a project manager, as this position often still exists in the companies due to its organisational legacy. In most cases, each team has its own Scrum Master, but in two cases, the Scrum Master is shared between collaborating Development Teams.

The Product Owner role is less present in the cases. It seems difficult to fit this relatively empowered role in as a natural part of the existing organisations. Whereas the Scrum Master's focus on process facilitation and team dynamics seems less disruptive to the traditional organisation, the decision power and responsibility of the Product Owner is typically placed in management boards, portfolio managers or other high-ranking parts of the companies. In most case companies this shows in the fact that the Product Owners title is not officially used.

3.2 Identified Team Composition Challenges

During the interview with the respective company representatives several challenges in relation to team composition were revealed. Table 3 below presents a summary of the challenges, and the following section will go into a deeper analysis of them.

Table 3. Team Composition Challenges

#	Challenge	Description
1	Extreme Cross-disciplinarity	Need for too many disciplines to fit into reasonable sized development teams.
2	Cross-team synchronisation	As the wide span of disciplines may require different Sprint length due to their different conditions, it becomes increasingly difficult to keep results from different functional teams in sync.
3	Team Polarisation	Some disciplines perform better than others and fit more naturally to the rules of Scrum.
4	Team location	Scrum requires Development Team members to be co-located, which to some companies is difficult to comply with.
5	Role- and empowerment structures	The roles and empowerment structures of Scrum are different from traditional patterns, and are therefore challenging to fully implement.

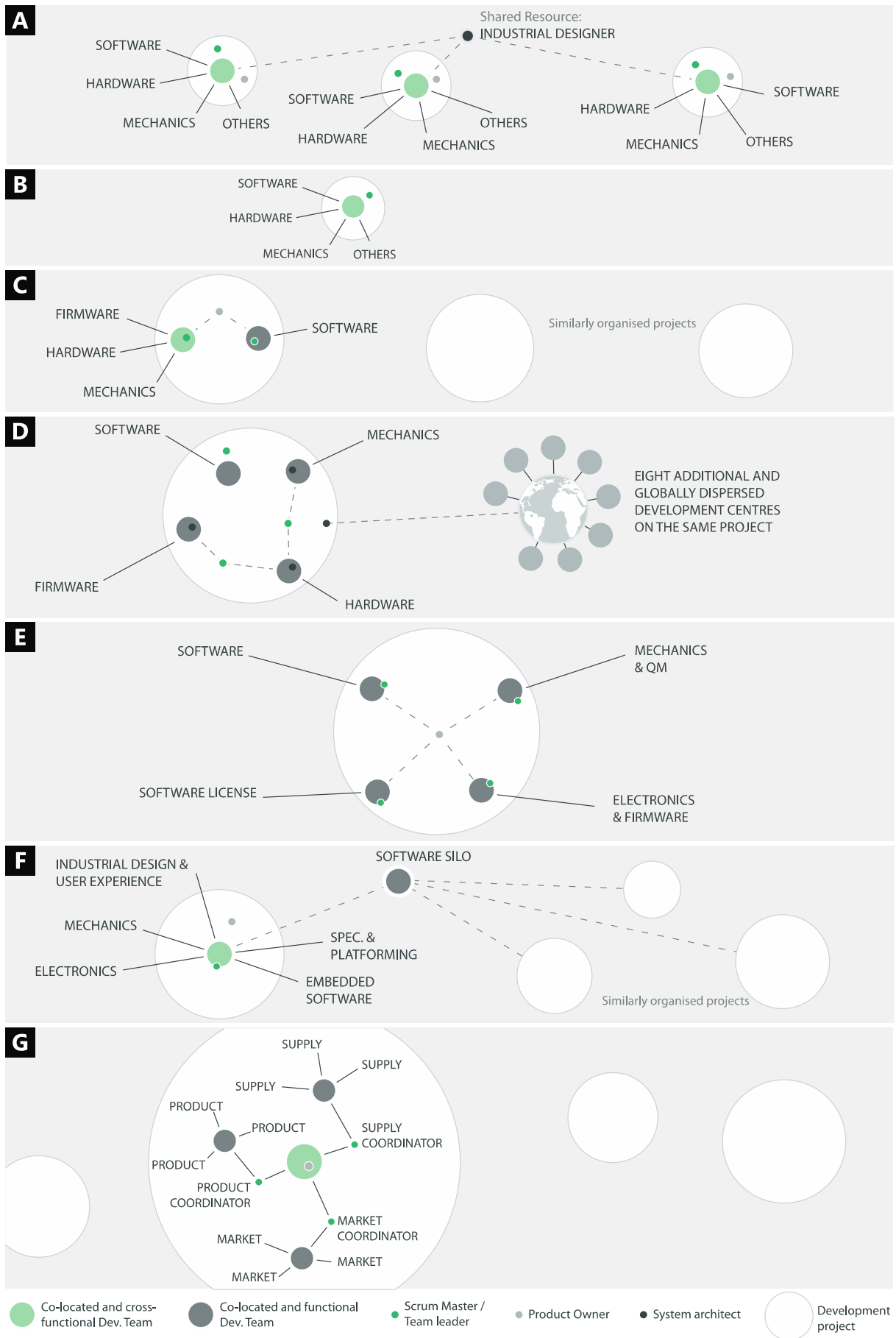


Figure 2. Overview of team structure and organisation in Case A to G.

4 ANALYSIS OF THE CHALLENGES

Not two organisations are alike, and team composition is highly dependent on the organisational setup in the respective companies. This may also be the reason why all seven cases in this research project are different in regard to how development teams are set up. The companies with team composition challenges do not seem to have other significant correlations than the fact that they all tend to deploy relatively large teams or multiple teams working on the same project. Beside that, they all differ in regard to their level of compliance to Scrum, their experience with Scrum and their respective team compositions.

4.1 Extreme Cross-disciplinarity

The official Scrum Guide promotes cross-functional teams, and in software development, which is the original domain of the Scrum Guide, cross-functionality means a mix of disciplines *within* the software domain. In integrated product development cross-functional teams entail a significantly larger variety of involved disciplines. This fundamental difference in the transition from software development to integrated product development clearly has some consequences to the teams in the investigated cases. Extremely cross-functional teams have certain communication issues, which, in some cases, lead to a drop in motivation for the individual developer. The fundamental difference between the two domains is that in software, team members have software development as a unifying discipline; the majority of a cross-functional software development team has to some extent overlapping competences. As it is illustrated in Table 2 earlier, this is not necessarily the case in cross-functional teams in integrated product development, which may very well include software, firmware, hardware, mechanics, industrial design and more.

The Scrum Guide (Sutherland and Schwaber, 2013) requests cross-functionality on one hand and a large amount of involved disciplines on the other. In most cases this would compromise the guide's recommendations in regard to team size. In the investigated companies, this dilemma has been solved in a variety of ways as illustrated in Figure 3 below.



Figure 3. Overview of the different combinations of team compositions

As it is seen in Figure 3 above, the teams are composed in different ways in almost all seven cases. No universal solution seems to exist. Interviewee 1 from Case F reflects on the difficulty of having a cross-functional team in an integrated product development project: *“In the software silo there is a critical mass of developers with the same competences. In our project team we can easily be just one electro technician, one chemical engineer and one mechanical engineer. They just really can’t share tasks other than getting the coffee.”*

As indicated by Interviewee 1, the cross-functional team is not able to achieve the same collaboration synthesis as is achievable in the software development silo. This might very well be the reason why the organisation is slowly starting to build up competence silos of mechanics, firmware and chemistry similar to the software development silo. In Case D, which describes three synchronised and functionally divided teams, interviewee 2 argues that even with functional teams the close collaboration may be difficult, due to the need of a broad variety of experts: *“I have six developers on the Mechanics team, all with very different competencies, so this is not true Scrum. I cannot just put anyone onto a certain task, and that’s a challenge. One person does all the plastics and another does this and that.”*

Together, the two quotes above reflect some of the difficulties in composing teams in integrated product development. It can be argued that two different paths can be taken, when composing Scrum development teams for integrated product development of a certain complexity: Several parallel functional teams *or* large and extremely cross-functional teams

In this regard, the challenge is to balance cross-functionality with a certain critical mass of homogeneous competence in the development teams.

4.2 Cross-team synchronisation

In both Case F and Case D multiple Scrum development teams are working together on the same project. However, the two companies are handling this with varying success. While the teams in Case D are systematically communicating through both Scrum Masters and System architects, the management in Case F has not yet established a formal way of synchronising collaborating teams. In Case F the teams are not conducting the Sprints synchronously; and the fact that the software development silo is supporting several other development projects, makes a close collaboration a big challenge:

Case F Interviewee 1: *“If we just focus on our own little area of competence, we deliver in time for sure. We just don’t deliver it in time to those who need it, and that’s a really big challenge.”*

The difference in the two cases may be explained by the difference in the experience with Scrum. While Case D has had a couple of years more to establish an efficient way to work with multiple teams, Case F may still be in the process of implementing the basic Scrum process within the separate teams.

4.3 Team Polarisation

In Case E the Scrum development teams are also divided by disciplines. In this case, a project room with a common Scrum board and visible task descriptions across all the involved Scrum development teams supports the communication and synchronisation between the teams. However, in that case the division in functional teams leads to another team composition issue:

Case E Interviewee 2: *“When it really works well here and not that well there, it might create some kind of polarisation. There’s the team that knows how to do it, and then there’s the team that doesn’t. I think we need to learn how to handle this in a way. We need to get an understanding of the basic differences. Why we act in this way and you act in that way – as part of the same overall team.”*

According to Case E Interviewee 2, the Mechanics team has difficulties in sticking to the Scrum practice in the same way as the other teams, and this creates a polarisation in the overall development team. In the quotation of interviewee 1, it is argued that it has something to do with a certain culture in the Mechanics team. In the same way Interviewee 2 describes mechanical development *as an amoeba*, which could be interpreted as if mechanical development were beyond any rational structure or logic.

To some extent, the distinctive character of the Mechanics team is also seen in Case F, where Interviewee 1 argues that the Industrial designers working with the development team are beyond any pedagogical reach: *“We have this group which is simply just beyond any pedagogical reach [pointing at Industrial Design and Usability]. There’s no planning, things turn up from day to day, and they expect that everything can be done by tomorrow.”*

In both cases, the specific groups have difficulties in adjusting themselves to the rather strict practice of Scrum, which may somehow result in a polarisation of the team.

4.4 Team location

The fourth team composition issue brought forward here concerns dispersed versus co-located teams. The majority of the development environments from the seven cases are co-located; but both Case B and Case G include dispersed teams. It seems rather clear that dispersed teams are a challenge as Scrum is promoting a relatively close contact and daily communication within the team. Case C is an example of an organisation starting out with Scrum in dispersed teams. However, the development department was quickly reorganised and the Scrum development teams were gathered in co-located teams, which improved the situation considerably.

Interviewee 1 from Case G describes the challenge: *“They are placed all over the country, and, believe me, sitting in a building across the street can be a long distance. Originally we had a vision about the developers working closely together in project rooms, but this is just not the way it works. They are divided into their respective areas of disciplines and scattered across the country.”*

4.5 Roles and empowerment structures

The last challenge presented here concerns the Scrum Team in a broader perspective. The Scrum framework comes with its own set of roles and when combined with the existing roles and positions of a typical organisation, it seemingly creates certain conflicts.

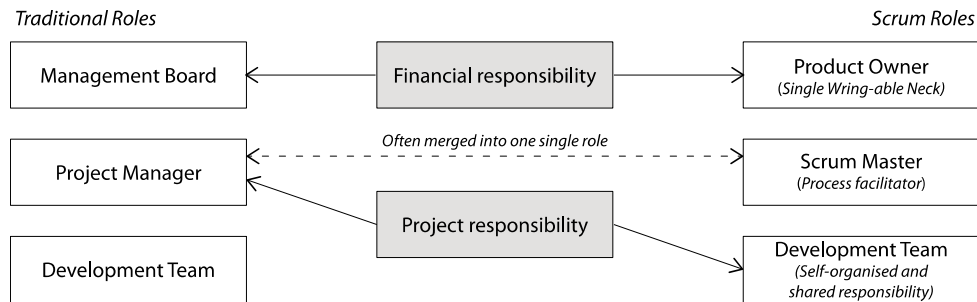


Figure 4. Roles and areas of responsibility in traditional setups and Scrum.

Scrum roles and traditional management roles, such as the project manager role, have a certain overlap, but they may be said to represent two different paradigms in development. While the responsibility for the execution of projects is born by the project manager in the traditional management model, project management is a shared responsibility in the self-organising Scrum development team as argued by Rudman (2010). The Scrum Master is only facilitating the process, and the Product Owner represents the customer and the business perspective.

In many of the investigated cases, the Product Owner is absent, and often the traditional management model and the Scrum framework only flank each other in the presence of one single person playing the dual role as both Project Manager and Scrum Master as it is seen in Figure 4 above. However, according to the data obtained through the cases, this dual role does not seem to cause any significant problems. On the other hand the role of the Product Owner – perhaps more rightly the absence of it – seems to cause some frustrations in the development teams as in Case D: “*There have been a lot of battles and they have taken a huge amount of time – at that point it would have been nice if an actual Product Owner would have taken those decisions.*”

The Scrum framework does not give any guidance to how it should be implemented in a large organisation. This is decided by the management, and to some extent also the development environment in which it is implemented. Only in a few of the cases, all roles of Scrum are present and given responsibility in accordance to Scrum.

5 CONCLUSION

This paper has been looking into the challenges of composing Scrum Development Teams in integrated product development environments. Seven case companies have been investigated, and a series of challenges related to team composition have been identified and analysed.

As it has been mentioned in the analysis, all seven cases have organised their respective Scrum Development Teams in different ways. Some are organised across multiple teams and others in just one single team. This shows that there are several strategies for conducting Scrum, and it shows that each company has to find a way to set up the Scrum development teams that is right for that specific situation. The concept of cross-functional teams is promoted in the original Scrum Guide, but in integrated product development it seems to almost always be preferred. Even if the teams are functionally divided, a certain cross-functionality seems to be inevitable. However, seen from an organisational perspective, the challenge in regard to team composition seems to be to compose the right teams based on the available resources, taking into account cultural differences and domain-related difficulties, and lastly to ensure synchronisation and close communication between teams working together on the same project.

Seen from the developers’ perspective, collaborating in a cross-functional team can be just as difficult as it can be enriching. In Scrum the Development Team collectively commits itself to the specific work packages of a Sprint, and in cross-functional teams it may be difficult to the individual developer to assess the extent of the work outside his or her own area of competence, just as the communication in such a team may prove difficult. The discussion about whether functional or cross-functional teams

are preferred is much influenced by the organisational conditions, but it is also a matter of the type of the developers.

Clearly there are plenty viable ways of conducting Scrum-like practice in integrated product development from a pragmatic perspective. The challenge is due to the fact that not two organisations are alike, and that each organisation has to find its own way to adjust to the principles of Scrum when setting up development teams. Nevertheless, there are still *some* ways and experiences shown by the cases, which may generally improve the performance in the development environment.

REFERENCES

- Hertenstein, J.H. and Platt, M.B. (2000) Performance Measures and Management Control in New Product Development. *Accounting Horizons*, Vol. 14, No. 3, pp. 303-323.
- Highsmith, J. (2001a) History: The Agile Manifesto [online], <http://agilemanifesto.org/history.html> (2014-12-11).
- Highsmith, J. (2001b) History: The Agile Manifesto [online], <http://agilemanifesto.org/> (2014-12-11).
- Krafcik, J.F. (1988) Triumph Of The Lean Production System, *Sloan Management Review*, Vol. 30, No. 1, pp. 41-52.
- Kvale, S. and Brinkmann, S. (2009) *InterView - Introduktion til et håndværk*, 2nd ed. Copenhagen: Hans Reitzels Forlag.
- Rudman, C. (2010) Answer to Jeff Watson [online], http://www.linkedin.com/answers/business-operations/project-management/OPS_PRJ/611543-1643668 (28-05-2012)
- Schwaber, K. (2004) *Agile Project Management with Scrum*. Boston: Microsoft Press.
- Schwaber, K. and Beedle, M. (2002) *Agile Software Development with Scrum*, New Jersey: Prentice Hall.
- Suscheck, C.A. and Ford, R. (2008) Jazz Improvisation as a Learning Metaphor for the Scrum Software Development Methodology. *Software Process Improvement and Practice*, Vol. 13, June, pp. 439-450.
- Sutherland, J. and Schwaber, K. (2013) *The Scrum Guide* [online], <http://www.scrumguides.org/scrum-guide.html> (2014-12-11)

ACKNOWLEDGMENTS

For a very constructive dialogue about the challenges of working with Scrum in Integrated Product Development, the author would like to thank all the seven case companies and all the interviewees, participating in this investigation, for their valuable time and effort.