



AGILE BEYOND SOFTWARE - A STUDY OF A LARGE SCALE AGILE INITIATIVE

L. Lindlöf and J. Furuhjelm

Abstract

Central to the effectiveness of design teams is how the planning of the design work is organized and coordinated. This paper describes how Saab Aeronautics, a large Swedish developer of jet fighter airplanes, has adopted an agile methodology on a large scale. The paper discusses central mechanisms of agile methodology including development of both software and hardware. The findings indicate two main factors that play an important role in reaching the benefits of agile in a large organization: setting up a system allowing focused team-work and giving the teams an empowered role in planning.

Keywords: agile, teamwork, large-scale engineering systems, software engineering, case study

1. Introduction

The popularity of agile methodology is constantly increasing, and in addition to the traditional focus on software development, agile is currently slowly including also other contexts. That transition sparks both a discourse on a principle level around the definition of agile and its purposes and a discourse on the more practical challenges related to the transition. Also, the initial application of the agile manifesto was typically focused on teamwork in small projects and organisations. Over time, the interest for large scale implementation involving many teams working together has increased. This paper primarily contributes to the discourse on the practicalities of scaling agile from individual teams to a larger network of teams, based on a close-up case study of a large agile initiative at Saab Aeronautics in Sweden. Saab has been working with agile methodology in different contexts and to differing extents for over a decade. Interestingly, they are generally considered to have a well-performing development organization by business journalists (e.g. de Briganti, 2017), and they have also been pointed out as an example in the May 2016 issue of Harvard Business Review (Rigby et al., 2016). This study aims to capture some of the central learnings that the organization has made during its agile initiative.

2. Theory

Agile software development methods follow a set of principles to allow a team of developers to continuously adapt to changing requirements. Those principles were formulated in 2001 in the Agile Manifesto (AgileAlliance, 2001) by a group of software developers as a reaction to the perceived inadequacies of the prevailing waterfall method (Larman and Basili, 2003). The Agile Manifesto builds on four core values:

*“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.”
(AgileAlliance, 2001)

Agile methods rely on iterative and incremental development where the delivery of software is done continuously rather than in one instance at the end of the project. The main purpose to do this is to be better able to adapt to a changing environment – to be more “agile” (Cao and Ramesh, 2007). To achieve this, employing self-organizing cross-functional teams is considered central (Hoda et al., 2012). Delegation and empowerment of the teams is considered a way to increase their potential commitment to the tasks, and there are evidence that this improves efficiency (Tessem, 2014) as well as individual well-being (Laanti, 2013).

The principles that the Agile Manifesto brings together constitutes a platform on which many methods and process designs build. One of the most prominent methods is Scrum which is also the main source of inspiration for Saab’s agile initiative. Scrum is based on planning the development in “sprints” – typically 2 to 4 weeks, in which the team takes on a number of tasks that they estimate can be done within the limits of the sprint. A manager called “Product Owner” prioritizes among the tasks that need to be done, so that the team always works on the most prioritized tasks at every point in time. But the team decides how many of the tasks they can manage within a sprint. Tasks that do not fit within the sprint need to wait for the next sprint, where a new prioritization is made and the team takes on new tasks. Scrum as a method is often summarized in three roles, three artefacts, and four meetings (Schwaber, 2004):

Scrum roles

The **Product Owner** is responsible for the overall project and represents those actors that have an interest in what the development team works with. The Product owner prioritizes among all the tasks that are allocated to the team. The **team members** work with development of the functionality they decided to work with during that specific sprint. Their main task is to figure out how to solve the problems they have committed to during that sprint. To help them keep track of their progress, they have a **Scrum master** that also coaches them in the scrum process and helps the team to focus on their work.

Scrum artifacts

To help the different roles follow the scrum process, scrum suggests using three different artefacts. First of all, the **product backlog** is a comprehensive list of all the requirements for the end product that are known at any given time. Importantly, the requirements are listed in order of importance, so that when the team plan their sprint, they make sure that they prioritize the most important work. The product owner is responsible for the prioritization, and the team is responsible for how much work they can commit to during one sprint. To keep track of how the team is doing in relation to the overall project, a **burndown chart** is used. It shows how much work is left in the project and at what speed the requirements are ticked off, giving an estimate of a time of completion. In addition, a **sprint backlog** is used to keep track of the requirements that are addressed within the specific sprint.

Scrum meetings

The backbone of scrum is its strategy of dividing all work into equally large “time boxes” referred to as sprints. A sprint has a standard length (typically 2-4 weeks depending on the nature of the work) and a standard format including four team meetings with different purpose: On the **sprint planning meeting**, the team decides what to do for the next sprint in dialogue with the product owner. They look at the product backlog to understand what needs to be done, and select an amount of work that they estimate can be done within one sprint – using the prioritization set up by the product owner. During a sprint, the team members meet daily to discuss the status and progress of the sprint. These **daily scrums** last for about 15 minutes. When the sprint is over, the team members hold a **sprint review meeting**, also called demonstration, to present to the product owner and any other stakeholders the outcome of the sprint. After that, a **retrospective meeting** is held to discuss how the team can improve for the next sprint based on the experiences from the previous sprint.

2.1. Leadership

There are some fundamental differences in preferred management practices in the case of scrum versus traditional waterfall type development. First of all, management gives the teams a higher degree of empowerment than in traditional development, and instead of trying to control the team, management focuses on protecting the team from interruptions and removing obstacles for the team (Moe and Dingsøy, 2008). Detailed planning and follow-up are taken care of by the teams themselves. In this context, trust becomes important. The team understands its velocity and thereby the optimal sprint contents, and therefore, management cannot push more into the sprint scope without hampering the sense of commitment and losing efficiency.

Also, when shifting from waterfall to scrum, the management focus shifts from deadlines to priorities. The detailed top-down plans are replaced by the backlog and the sprint plan, which is managed by the team themselves, in collaboration with the product owner. When following up team performances, the managers focus on optimal priority and team velocity/efficiency. As said, agile management focuses on removing impediments. Both small ones, which could be issues that needs to be solved within hours, and larger ones, like changing supporting software systems that could take months. During product development many unpredicted events occurs. To ensure the focus and the minimum of multitasking the product owner has the job of a door guard; identifying incoming demands on the team and in case they are urgent and high priority break the sprint contents (Paasivaara et al., 2012). Not to hamper the sense of commitment lower priority tasks should then be taken out of the sprint scope.

2.2. Agile beyond software

Although agile methods were developed by and for software developers, the discourse on their relevance for other types of development projects has been intensifying the past years. On a principle level, agile aligns well with what development process theorists call an “experiential strategy” (Eisenhardt and Tabrizi, 1995), and what strategic management theorists call “dynamic capabilities” (Teece et al., 1997). Larman and Basili (2003) account for an historical overview of the development of the iterative and incremental development strategy that characterizes agile methods and their study concludes that the ideas have been around at least since the 1950’s. Abbas et al. (2008) argue that although agile methods have been around that long, they have not been given due attention until the emergence of the agile methods after 2001, when the agile manifesto marked a turning point in the attitude towards agile methods.

Recently, Cooper (2016) and Cooper and Sommer (2016) contrasted agility with a classical stage gate process which in essence can be considered a waterfall strategy. Cooper is often attributed the formulation of the stage gate process (Cooper, 1990). In their effort of merging the two (seemingly contradictory) concepts, they identify a complementing relationship between the two, where the gated process is on a macro-level, supporting management in taking strategic decisions on what projects to prioritize and the agile process is on a micro-level supporting the development team in taking tactical and operational decisions on what to prioritize within the projects (Cooper, 2016). While their overall evaluation is that stage-gate and agility are not incompatible, they do identify two main challenges related to using a software-accented agile strategy in the development of a physical product (Cooper and Sommer, 2016); 1) *Defining a done sprint*. Defining a sprint and the conditions for whether the sprint is done or not seems to be substantially more difficult in hardware development. Software can more easily be divided into subparts that can be handled individually within weeks. The idea of using a sprint-tactic is to have a result at the end of the sprint that can be tested and evaluated. This is not necessarily feasible when developing hardware. 2) *Resource Allocation*. Project teams in software development are more often dedicated to a single project, and also co-located. This makes it a lot easier to use the sprint-tactic than if individuals of the team are working on a number of projects in parallel.

3. Method

This paper is based on a qualitative study of the agile methodology applied at Saab Aeronautics, a large Swedish engineering company in the aircraft industry. Using an insider-outsider approach, the two authors have studied the transition from very close distance during 10 years, through the active presence of the insider. The transition at Saab has initially focused on software development, but over time

gradually also included more and more hardware development. Saab is a suitable case when studying agile transitions, partly because they have extensive experience in the agile methodology, and partly because they can showcase strong development performance in terms of lead time for developing a new generation airplane (e.g. de Briganti, 2017).

3.1. Our study

The research strategy for this study is to draw on the advantages of having both an insider – someone who is a part of the studied context and shares identity, language and experiences with the people who are a part of the study (Asselin, 2003), and an outsider – someone who is external to those commonalities and therefore can distance themselves from biases, prevailing paradigms and potential role-conflicts. In this study, using interplay between those two roles is employed as a strategy to gain both of the mentioned benefits.

The insider has spent in total 10 years at the company, working in different parts of the engineering department including assignments on different levels, e.g. team leader, subproject manager and project manager as well as different technical roles within software, hardware and equipment integration. Importantly, he has throughout these years and as a part of his work been engaged in building knowledge over time regarding working methods and organizational improvement. This gives him a very good understanding of how the organization responds to changes, and how the past activities shape the present. The main role of the insider in this project is to gather, compile and analyse qualitative data from the organization.

The outsider has been working as an outsider with the company in similar previous research projects, so he is familiar with the organization, but has never been a part of it. Instead, he has experiences from other similar and dissimilar organizations and provides a more theoretical perspective on the findings. With that as a starting point, the main role of the outsider is to join the insider in the analytical work and together shape the analysis and formulate the conclusions that can be drawn from the research project. To map out the most relevant findings from Saab's implementation of agile, a comprehensive list of items were created by the authors jointly. The list then became the basis for an iterative analysis including the addition and subtraction of items as well as the refinement of items. In the later stages of analysis, the list were validated with two experts at Saab. This validation rendered only minor changes and in the end, they lead to the formulation of the sub headers in the following chapter and ultimately the conclusions of the paper.

3.2. Saab and their agile initiative

Saab Aeronautics is part of the Saab Group which is active in the security and defence business. Even though Saab is active on the worldwide market, the vast majority of the development in the airplane sector is taking place in Linköping, Sweden. At that site, Saab has been working with the agile concept for more than a decade. Even before the studied project started, agile methodology was well-spread. Today, a significant part of the teams in this project apply agile practices such as scrum and kanban. It is most commonly applied throughout software development including test, systems design, and integration, but also in engineering design and technical support functions.

This case study involves the development of Saab's new "Gripen E" fighter jet. Based on the existing Gripen C/D platform, the Gripen E aims to be one of the most advanced multi-role fighters in the world by combining advanced technology and operational effectiveness yet in an affordable package. The project includes a major upgrade of all technical system - sensors, avionics, datalinks etc. It also includes a new cockpit layout with a large wide area display for improved situation awareness as well as integrating a new stronger engine. In short - the whole airplane is upgraded and the project involves around 2000 engineers.

With the start and expansion of the of Gripen E project, there were enough people with significant knowledge on scrum to make the method get traction in the project. It started to spread throughout the project, and new recruits quickly adopted the team practises. Also, the management within software had a good understanding of the management practices related to the agile methodology. With a strong foothold in software, the practises have then spread to other parts of the Gripen development. Team-members and managers naturally move around within the Gripen E project and take the knowledge and

practices with them. A daring trial-and-error and continuous improvement culture at Saab helped, as a strict text-book scrum implementation has shown not to be optimal in all teams. Rather, with a healthy retrospective practices in use, the application of the scrum methodology regulates itself towards a situation where each team finds the method most useful.

There has not been any central top-down initiative regarding agile implementation within Saab Aeronautics. There are a few, support persons to provide coaching if required by any team. Neither is it common with external consultants providing assistance. Rather, the roll-out has been evolutionary. Knowledge has been growing and implementations changing character over time. Teams implement scrum and adopt it to their needs, some teams leave scrum for kanban and other methods if required.

4. Findings

This chapter accounts for the empirical findings of the study. They are categorized under 3 headlines; “Focus” and “Team-work” deal with the findings from the agile initiative at Saab in general, and “Scrum in mechanical engineering” describes differences between the applicability of agile in software and mechanical engineering in particular.

4.1. Focus

At Saab, the development teams applying scrum all have the same sprint length, three weeks, and the same start and stop dates. The teams start the sprint with a sprint planning where the content of the sprint is defined. Through this definition, the sprint content is “fence-gated” and the aim is to keep the content stable throughout the sprint. This provides the teams with the possibility to focus their attention on the tasks that are prioritized at any given point in time. Within many of the teams, this focus has led to higher efficiency and a less stressful working situation for the developers.

To enable the sprint content to be stable, thus allowing the focus, there is a system at Saab to synchronise deliveries between teams. Before the start of a new sprint, each team identifies the need of input from other teams and gives the providing teams a heads-up alert about the need. They can then take into consideration the need of deliveries to other teams during their sprint planning. This means that there is a time-window to ask for deliveries and if they miss that opportunity, next time will be in three weeks, before the next sprint. This cause for discipline where teams need to think ahead about their needs of input not to disturb other teams during an ongoing sprint. This is a major advantage with the synchronized sprint cycle.

The sprint planning meeting decides on the sprint content will be the focus of the team members for the next three weeks. The workload is balanced and take into account the capacity of the team during the period. The number of parallel tasks undertaken are kept to minimum, to allow an optimal focus, and to ensure that tasks reach a completion. The everyday working conditions are thereby characterised by limited multitasking and the loss of efficiency that comes with that. But perhaps more importantly, the “fence-gated” sprint contents function as a “protection” of the team members from incoming requests to prioritize something else than what was originally planned. Instead, requests that are posed to the team are evaluated and if not super urgent end up in the backlog. That means that they are queued up for future sprints. The team members can commit to the chosen tasks and the synchronised system at Saab creates good conditions for engineers to spend their time working with the highest prioritized tasks.

4.2. Team-work

Within the Gripen E project, there is on the top level a time plan in a traditional setup. This includes overarching deliveries and is broken down within the subprojects. With these plans, the teams’ product backlogs are aligned. The detailed planning, i.e. the sprint planning discussed above, is done by the teams themselves. They are in that sense not assigned tasks, they take on tasks themselves. This has a significant positive impact on the team-spirit, that the sprint goals are a common team undertaking. They can control their own situation and the risk of making overly optimistic time plans is reduced.

The team picks their tasks from the prioritized backlog, which is a joint responsibility of the team and product owner. The product owner at Saab is not a direct representative of the customer as Scrum prescribes. That role needs to be broken down when it comes to products of the complexity that Saab

handles, and to comply with the organizational distance from the developer to the customer. This slight change in the role of the product owner becomes an important difference between small scale and large scale application of scrum. In some subprojects the product owner covers only one team each and in some as much as 6-8 teams. The main responsibility of the product owner is to ensure that the product backlog is complete and prioritized at the beginning of each sprint so that the team can make an informed decision on what tasks to assign to that particular sprint.

The development of Gripen E takes place in steps with basic airplane functionality first and tactical functionality added subsequently. So the teams move from concept to detailed development and they bring their methodology with them. The retrospective that takes place at the end of the sprint ensures an optimisation of working practises. The teams have in their mind-set to constantly improve the way they work. There is no internal standard on how to form the backlog, tools for follow-up of sprint tasks, measuring velocity etc. This is optimised on a team level taking local factors into account. Over time, the team improves their ability to estimate their work capacity during a sprint, which means that the predictability of the development work improves. The focus of the retrospective is identifying potential improvements. Some ideas for improvements go directly into the teams' backlog, other ideas are distributed to other teams, for example support teams, yet other issues are escalated to management. For this to work, Saab has acknowledged that the stability of the teams is essential for efficient team-work. It takes some time to organise the work optimally within a team and that disrupts if team members are moved around. Within the Gripen E almost all engineers work only within one team. But the project has over recent years been expanding so teams split naturally when they become too large.

4.3. Scrum in mechanical engineering

So far, the findings have been related to both software and hardware teams. But there are some differences between the two types of development that can be seen at Saab and deserves due attention. Within the Gripen E project, there are hundreds of engineers outside software that apply scrum. Saab has experience from scrum teams dealing with tasks ranging from systems design, product safety, and equipment integration to vibration analyses. Also within the engineering design the practises are well-known. But the implementation varies more outside software development, and "textbook scrum" is applied only to a certain degree.

Tasks at Saab are in many teams outside software of a more extensive character, duration could be several hundred hours, with a limited possibility of breaking down in smaller tasks. In this case, the sprint planning focuses on identifying the sprint goal, a subset of the completion of the task. Within the Gripen E project the teams applying scrum outside software tend to have more infrequent stand-up meetings. The "dailies" are not daily. This is because tasks are longer in duration and are assigned on a more individual basis. To everyday describe the progress for the fellow team members make thus less sense.

Another difference seen in the application outside software at Saab is that tasks are typically of a more individual character. Software teams are generally more homogenous from a competence perspective. That means that within the software teams a higher degree of the tasks can be carried out by any of a number of team members, for example testing. In for example engineering design, the developers to a larger extent have a certain area of the airplane as their responsibility. This could be for example hydraulic piping in the mid-section of the aircraft. This has an effect on several aspects of how scrum is applied. The allocation of a certain task to a certain engineer generally takes place already during the sprint planning in contradiction to software where this is allocated during the daily stand-up. The importance of not overloading a sprint is the same, but it comes to a more individual level of identifying an overload. Setting of priority is thus also done on a more individual basis.

The findings from Saab regarding the nature of the benefits of scrum do not vary much between software and non-software teams. The focus it brings and with that minimum multitasking during a workday is central. Clear priorities and a balanced workload arguably boost efficiency and employee satisfaction wherever applied.

5. Conclusions

In conclusion, the case study identified three areas where experience from Saab particularly add to previous knowledge on large scale agile initiatives. Firstly, the large scale implementation of scrum with

the synchronised input/output deliveries from the teams add to development efficiency since this establishes a common rhythm. The input/output coordination enables a focus for the teams, allowing them to work for three weeks on the content they decided on in the sprint planning meeting, without having to stop and re-prioritize whenever external requests come in. The question whether synchronized or overlapping sprints in multi-team settings are more favourable is debated by both practitioners and researchers (Greening, 2010; Nortier et al., 2011), and the evidence from this study favours synchronized sprints. Findings support the Coopers (2016) argumentation where a traditional stage-gate set up on a macro level can be coupled with a successful agile implementation on a micro/team level. Secondly, within the Gripen E the project team-work is in focus. In line with Hoda et al. (2012) emphasis on the self-management of the teams and Moe and Dyngsørs (2008) contribution on the importance of a management that allows the team to set their own goals and how they should work to achieve them, this study puts forward that the efficiency on a team level is highly attributed to the sense of commitment within the team. In Saabs case, it seems like the commitment stems from the fact that the teams set their own sprint plan, thus make a common undertaking to accomplish the sprint goals. They also have a high degree of freedom regarding their working practises. All in all, the team-work as a central principle, the ability of the team to focus as discussed above together with the self-management of the teams all contribute to the sense of commitment to the team goals identified in the study. Thirdly, experience from scrum outside software is less discussed in literature. Saab has significant experience since several years regarding this. The study shows that the same benefits occur as in software, i.e. efficiency rise due to the focus and priority system, but some adoptions need to be made regarding its implementation. While in software a higher extent of all tasks could be carried out of by a number of the team members, for example testing the software, the tasks in mechanical engineering teams are allocated more individually because of the higher competence diversity. This requires the sprint planning, with prioritized tasks, also to be on a more individual level as also discussed by (Cooper and Sommer, 2016). Finally, this paper aims to contribute to the discussion on how to apply an agile methodology in large organizations developing both software and hardware. Saab presents an interesting case in that they are considered having an efficient development of new jet fighters, and they have employed the agile framework for quite some time, including an increase in the hardware development the past years. Their experiences can prove to be an important input to other organizations striving for a more agile and efficient way of working.

References

- Abbas, N., Gravell, A.M. and Wills, G.B. (2008), "Historical roots of agile methods: Where did 'Agile thinking' come from?", In: Abrahamsson, P., Baskerville, R., Conboy, K., Fitzgerald, B., Morgan, L. and Wang, X. (Eds.), *Agile Processes in Software Engineering and Extreme Programming*, Springer, Berlin, pp. 94-103. https://doi.org/10.1007/978-3-540-68255-4_10
- AgileAlliance (2001), *Agile manifesto*. [online] Available at: <http://www.agilemanifesto.org>.
- Cao, L. and Ramesh, B. (2007), "Agile software development: Ad hoc practices or sound principles?", *IT professional*, Vol. 9 No. 2, pp. 41-47. <https://doi.org/10.1109/MITP.2007.27>
- Cooper, R.G. (1990), "Stage-gate systems: a new tool for managing new products", *Business Horizons*, Vol. 33 No. 3, pp. 44-54. [https://doi.org/10.1016/0007-6813\(90\)90040-I](https://doi.org/10.1016/0007-6813(90)90040-I)
- Cooper, R.G. (2016), "Agile–Stage-Gate Hybrids: The Next Stage for Product Development Blending Agile and Stage-Gate methods can provide flexibility, speed, and improved communication in new-product development", *Research-Technology Management*, Vol. 59 No. 1, pp. 21-29. <https://doi.org/10.1080/08956308.2016.1117317>
- Cooper, R.G. and Sommer, A.F. (2016), "The Agile–Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity", *Journal of Product Innovation Management*, Vol. 33 No. 5, pp. 513-526. <https://doi.org/10.1111/jpim.12314>
- de Briganti, G. (2017), *First Flight of Gripen E Will Reveal True Cost of Fighter Development*. [online] defense-aerospace.com. Available at: <http://www.defense-aerospace.com/articles-view/feature/5/183525/first-flight-of-gripen-e-will-reveal-true-cost-of-fighter-development.html>
- Eisenhardt, K.M. and Tabrizi, B.N. (1995), "Accelerating adaptive processes: Product innovation in the global computer industry", *Administrative Science Quarterly*, Vol. 40 No. 1, pp. 84-110. <https://doi.org/10.2307/2393701>

- Greening, D.R. (2010), "Enterprise scrum: Scaling scrum to the executive level", *Proceedings of the HICSS 2010 / 43rd Hawaii International Conference on System Sciences, Honolulu, USA, January 5-8, 2010*, IEEE, pp. 1-10. <https://doi.org/10.1109/HICSS.2010.186>
- Hoda, R., Noble, J. and Marshall, S. (2012), "Developing a grounded theory to explain the practices of self-organizing Agile teams", *Empirical Software Engineering*, Vol. 17 No. 6, pp. 609-639. <https://doi.org/10.1007/s10664-011-9161-0>
- Laanti, M. (2013), "Agile and Wellbeing--Stress, Empowerment, and Performance in Scrum and Kanban Teams", *Proceedings of the HICSS 2013 / 46th Hawaii International Conference on System Sciences, Wailea, USA, January 7-10, 2013*, IEEE, pp. 4761-4770. <https://doi.org/10.1109/HICSS.2013.74>
- Larman, C. and Basili, V.R. (2003), "Iterative and incremental developments. a brief history", *Computer*, Vol. 36 No. 6, pp. 47-56. <https://doi.org/10.1109/MC.2003.1204375>
- Moe, N.B. and Dingsøyr, T. (2008), "Scrum and Team Effectiveness: Theory and Practice", In: Abrahamsson, P., Baskerville, R., Conboy, K., Fitzgerald, B., Morgan, L. and Wang, X. (Eds.), *Agile Processes in Software Engineering and Extreme Programming*, Springer, Berlin, pp. 11-20. https://doi.org/10.1007/978-3-540-68255-4_2
- Nortier, B., Von Leipzig, K. and Schutte, C. (2011), "The development of a software development framework by combining traditional and agile methods to address modern challenges", *Proceedings of ISEM 2011, Stellenbosch, South Africa, September 21-23, 2011*, Stellenbosch University, Stellenbosch.
- Paasivaara, M., Heikkilä, V.T. and Lassenius, C. (2012), "Experiences in scaling the product owner role in large-scale globally distributed scrum", *Proceedings of the ICGSE 2012 / 7th International Conference on Global Software Engineering, Porto Alegre, Brazil*, IEEE, pp. 174-178. <https://doi.org/10.1109/ICGSE.2012.41>
- Rigby, D.K., Sutherland, J. and Takeuchi, H. (2016), "Embracing agile", *Harvard Business Review*, Vol. 94 No. 5, pp. 40-50.
- Schwaber, K. (2004), *Agile Project Management with Scrum*, Pearson Education.
- Teece, D.J., Pisano, G. and Shuen, A. (1997), "Dynamic capabilities and strategic management", *Strategic Management Journal*, Vol. 18 No. 7, pp. 509-533. [https://doi.org/10.1002/\(SICI\)1097-0266\(199708\)18:7<509::AID-SMJ882>3.0.CO;2-Z](https://doi.org/10.1002/(SICI)1097-0266(199708)18:7<509::AID-SMJ882>3.0.CO;2-Z)
- Tessem, B. (2014), "Individual empowerment of agile and non-agile software developers in small teams", *Information and software technology*, Vol. 56 No. 8, pp. 873-889. <https://doi.org/10.1016/j.infsof.2014.02.005>

Dr. Ludvig Lindlöf, Post-doc
 Chalmers University of Technology, Technology Management and Economics
 Vera Sandbergs Allé 8A, 41296 Gothenburg, Sweden
 Email: ludvig.lindlof@chalmers.se