Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Antti Pitkänen

Agile Transformation

A case study

Master's Thesis
Espoo, September 30, 2015

Supervisors: Professor Casper Lassenius
Advisor: Ville Heikkilä D.Sc. (Tech.)

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author: Antti Pitkänen

Title:
Agile Transformation - A case study

Date:	September 30, 2015	**Pages:**	94
Major:	Software Engineering and Business	**Code:**	T-76
Supervisors:	Professor Casper Lassenius		
Advisor:	Ville Heikkilä D.Sc. (Tech.)		

Agile methods have been widely adopted in the software engineering industry. In addition, agile software development has been studied extensively during the past decade. In general, studies tend to recommend having experts working in small, self-organizing and cross-functional teams. However, the environment in an enterprise can comprise of multiple, interlinked programs and dozens of teams working on them, which poses many challenges to adopting agile methods.

In this case study, we studied scaled agile methods at Rally Software, a software enterprise with over 500 employees. We conducted two rounds of interviews in the case organization. The first round of interviews assessed the initial state of the organization. The second round was performed after major changes in the organization and processes. In addition, we performed a literature review on case studies of other organizations utilizing large scale agile.

The results of the interviews present Rally's steps in striving for improving its agile scaled process. The first round results revealed a mature agile organization with multiple accumulated issues and willingness to turn itself around. After the transformation process, the second round of interviews showed positive trends in several areas. By enforcing more disciplined agile processes, Rally Software was able to increase its feature delivery rate and improve trust in the engineering department.

The literature review results present common challenges and success factors related to adopting and maintaining scaled agile in large organizations. The most frequently mentioned challenges were inter-team communication and coordination, requirements management and integrating other functions of the company into the scaled agile process. The most common success factors were coaching and training, piloting agile in smaller projects, change leadership and having a structured approach to facilitate inter-team communication.

Keywords: scaled agile, case study

Language: English

Aalto-yliopisto
Perustieteiden korkeakoulu
Tietotekniikan koulutusohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Antti Pitkänen		
Työn nimi:	Ketterän ohjelmistokehityksen muodonmuutos - Tapaustutkimus		
Päiväys:	30. syyskuuta 2015	**Sivumäärä:**	94
Pääaine:	Ohjelmistotuotanto ja-liiketoiminta	**Koodi:**	T-76
Valvojat:	Professori Casper Lassenius		
Ohjaaja:	Tekniikan tohtori Ville Heikkilä		

Ketterät kehitysmenetelmät ovat saavuttaneet vankan suosion ohjelmistoalalla ja niitä tutkittu laajasti viimeisen vuosikymmenen aikana. Tutkimukset usein suosittelevat asiantuntijoiden muodostamia pieniä, autonomisia ja monitaitoisia ohjelmistokehittäjäryhmiä. Suurissa ohjelmistoyrityksissä tuotekehitys voi kuitenkin koostua monista toisistaan riippuvista ohjelmista ja kymmenistä erillisistä ryhmistä, mikä tekee ketterien menetelmien käytöstä haasteellista.

Tässä tapaustutkimuksessa olemme tutkineet Rally Softwaren ketterää ohjelmistokehitysprosessia. Rally Software on suuri ohjelmistoyritys, jonka palveluksessa on yli 500 työntekijää. Suoritimme yrityksessä kaksi haastattelukierrosta, joista ensimmäinen kartoitti lähtötilanteen ja jälkimmäinen tilanteen merkittävien organisaatio- ja prosessimuutosten jälkeen. Tämän lisäksi teimme kirjallisuuskatsauksen, joka keskittyi tapaustutkimuksiin ketterien kehitysmenetelmien soveltamiseen suurissa organisaatioissa.

Haastattelutulokset esittelevät Rally Softwaren vaiheita sen pyrkiessä parantamaan ketterien menetelmien käyttöään. Ensimmäisen haastattelukierroksen tulokset esittelevät pitkään ketteriä kehitysmenetelmiä käyttäneen organisaation, jossa oli paljon kasaantuneita ongelmia ja tahto muutokseen. Muutosprosessin jälkeen tehty toinen haastattelukierros kertoo positiivisista suuntauksista useilla osa-alueilla. Kurinalaisella ketterien menetelmien käytöllä Rally Software sai parannettua tuottavuuttaan ja luottamusta ohjelmistokehitysorganisaatioon.

Kirjallisuuskatsaus esittelee tyypillisiä haasteita ja onnistumisen edellytyksiä liittyen ketterien kehitysmenetelmien käyttöönottoon suurissa ohjelmistokehitysyrityksissä. Yleisimmät haasteet olivat ryhmien välinen kommunikaatio ja koordinointi, vaatimustenhallinta ja muiden yrityksen toimintojen sisällyttäminen ketterään kehitysmalliin. Useiten mainitut onnistumisen edellytykset olivat valmennus ja koulutus, ketterien kehitysmenetelmien kokeilu pienemmissä projekteissa, muutosjohtajuus ja järjestelmällinen lähestymistapa ryhmien välisen kommunikaatioon.

Asiasanat:	ketterät kehitysmenelmät, tapaustutkimus
Kieli:	Englanti

Acknowledgements

I would like to thank Rally Software for enabling me to observe the transformation process in person at the headquarters in Boulder, Colorado. In addition, I thank all the interviewees for their time and effort in making this case study possible. I would like to give special thanks to Serina Clark for providing invaluable support for this thesis project at Rally Software.

I thank both Casper Lassenius and Ville Heikkilä for their interest in the research topic and guidance during the thesis writing process. In addition, I would like to thank Ville Saarinen for taking the time to proofread this thesis and giving me advice on improving the fluency of the text.

Espoo, September 30, 2015

Antti Pitkänen

Abbreviations and Acronyms

SAFe	Scaled Agile Framework
ALM	Agile Lifecycle Management
PSI	Potentially Shippable Increment
PPM	Product Portfolio Management
SOA	Service-oriented Architecture
COP	Communities of Practice

Contents

Chapter 1

Introduction

Agile methods have been widely adopted in the software engineering industry (Rodriguez et al., 2012). Agile software development has also been studied extensively during the past decade (Senapathi and Srinivasan, 2013). In general, studies tend to recommend certain types of environments for agile development. These environments usually consist of experts working in small, self-organizing and cross-functional teams (Boehm and Turner, 2005) (Dyba and Dingsoyr, 2009).

Inspired by the success of agile methods, enterprises are introducing agile software development into their large scale contexts (Razavi and Ahmad, 2014). However, the environment in an enterprise can comprise of multiple, interlinked programs and dozens of teams working on them. In the large scale of an enterprise, it is unlikely that everyone is an expert or cross-functionality in teams is common. This poses an additional challenge to adopting agile methods in enterprises: how can the methods be scaled to fit this suboptimal environment?

Rally Software is a software product enterprise that uses agile methodologies in its organization of over 500 employees. In this thesis, we aim to study Rally's agile methodologies and – more specifically – its implementation of the Scaled Agile Framework (Leffingwell, 2010).

Well-known agile methodologies, like Scrum or XP, do not take a stance on whether or how they should adapted to enterprise-scale. Nevertheless, many enterprises practice agile software development using methodologies like Scrum or XP (Dyb and Dingsøyr, 2008). Both of these methodologies were originally focused on teams, whereas in an enterprise-scale organization the individual teams are a part of the big picture. Subsequently, some large-scale implementations of the methodologies have been suggested by the authors, like Enterprise Scrum (Beedle), or by other parties, like Industrial XP (Industrial Logic, Inc.).

To adopt agile methods within the whole organization, enterprises must utilize some additional tools alongside the team-level process models. These tools must acknowledge that there can be multiple interconnected products or even portfolios of products within the software development organization. Consequently, the amount of stakeholders increases and dependencies are far more difficult to identify and manage. In addition, respecting the core principles of agile philosophy (Beck, K. and Beedle, M. and Bennekum, A. and Cockburn, A. et al) – valuing individuals and interactions over processes and responding to change – are not likely to manifest organically and thus need support from all the levels of the organization. Given these issues, there needs to be a holistic approach to successfully adopt agile methods in the enterprise scale.

1.1 Objectives

This thesis aims to evaluate how well Rally Software has been able to implement scaled agile software development. We compare challenges and success factors of scaling agile (as described in the literature) to the results of the interviews performed at Rally Software. In addition, we intend to evaluate to what extent the Scaled Agile Framework manages to address the challenges related to scaling agile software development in enterprises.

Besides the process point of view, the organizational implications of scaling agile software development are of interest to this thesis. The Scaled Agile Framework or any other agile process framework cannot exist in a vacuum. The organization may impose external restrictions to the framework and even resist agile values. Without full organizational support, the framework is not likely to reach its full potential. Thus, this thesis focuses on both the process and the organizational culture aspects of scaled agile software development.

The research questions of this thesis are:

1. What practices have been implemented in the case organization to support agile software development?
2. What are the common issues with scaled agile software development?
3. What issues has the case organization faced with scaled agile software development?
4. What approaches does literature and interviews suggest for resolving the current issues?

1.2 Structure of the thesis

In chapter two, we begin by presenting related work. Next, we continue with presenting background information about the case organization, Rally Software. In the fourth chapter, we present the Scaled Agile Framework and its implementation in the case organization. In the fifth chapter, we outline the research methods of this thesis. The sixth chapter focuses on presenting the interviews performed at Rally Software. In the seventh chapter, we discuss the results of the interviews and implications for the case organization. In the final chapter, we present a summary and a conclusion.

Chapter 2

Related work

This chapter presents previous work on scaled agile methodologies and their usage in the industry. First, we present existing process models for scaling agile methodologies. We aim to provide an overview of what process model alternatives exist in the industry. Next, we survey previous case studies on scaled agile and the Scaled Agile Framework. The purpose of the survey is to provide context for this case study of scaled agile. The survey also provides insights from other organizations that are using scaled agile practices. Parts of note from the other case studies are any identified challenges and success factors in scaling agile.

2.1 Process models for scaled agile

There have been several propositions for agile process models that support scaling. Scott Ambler and Mark Lines have developed the Disciplined Agile Delivery model that is defined in their book of the same name (Ambler and Lines, 2012). Mike Beedle is the author of the Enterprise Scrum process model, which expands the Scrum framework to increase scalability (Beedle). Arguably the most well-known scaled agile process model is the Scaled Agile Framework (SAFe) created by Dean Leffingwell (Leffingwell, 2010). The Scaled Agile Framework aims to aid enterprises in adopting agile practices. It provides a framework for organizing the enterprise and set of practices to use. The author of the framework claims it has been successfully used in enterprises employing thousands of software developers. However, there is a lack of academic studies supporting this claim.

In general, there are few studies of process models used in scaled agile software development. In particular, the Scaled Agile Framework has not received much attention from researchers, despite having been utilized by multi-

ple large enterprises (Leffingwell LLC, b). Therefore, we have chosen to make a case study of a company that has been using Scaled Agile Framework for several years.

2.2 Case studies of scaled agile

We chose to focus on finding case study papers of organizations utilizing scaled agile methods in this literature study. This was done to limit the literature study results to papers that had comparable evidence to the case organization of this thesis. Thus, we searched for systematic literature reviews on scaled agile and found two recent ones (Dikert, 2014; Razavi and Ahmad, 2014). The papers from these systematic literature reviews were included in our literature review. In addition, case studies that describe the usage of Scaled Agile Framework were of interest to this thesis. Thus, we included case study papers that cite Leffingwell's book on the Scaled Agile Framework (Leffingwell, 2010).

We reviewed the potentially relevant papers from the systematic literature reviews by reading the full text. The papers citing Leffingwell's book (Leffingwell, 2010) were first filtered by title and abstract and the ones deemed relevant by full text as well. As this thesis is focused on a specific case, papers that did not provide valuable insights into the context of this case study were excluded. A paper was excluded if it was not a case study or only described the themes presented in the following listing.

1. Issues of waterfall to agile transformation
2. Lack of management support for agile methods
3. Test or build automation issues

The stages of the literature review process and the references to the papers are presented in Figure 2.1. In the next subsections we present common challenges and success factors of scaling agile identified in the found case study papers.

Papers citing the Scaled Agile Framework book

Stage	No. of papers	References
Total number of papers	185	
Papers deemed relevant by title and abstract	3	Díaz et al. (2014); Heikkilä et al. (2013, 2015)

Systematic literature review by Dikert et al

Stage	No. of papers	References
Total number of papers	52	
Papers deemed relevant by full-text	43	Abdelnour-Nocera and Sharp (2007, 2008); Atlas (2009); Bang (2007); Beavers (2007); Benefield (2008); Berczuk and Lv (2010); Brown (2011); Chung and Drummond (2009); Cloke (2007); Cowan (2011); Evans (2008); Farrow and Greene (2008); Fecarotta (2008); Federoff and Courage (2009); Gat (2006); Goos and Melisse (2008); Hajjdiab et al. (2012); Hanly et al. (2006); Hansen and Baggesen (2009); Laanti (2008); Lee (2008); Leszek and Courage (2008); Lewis and Neher (2007); Long and Starr (2008); Maples (2009); Mencke (2008); Moore and Spens (2008); Nielsen and McMunn (2005); O'Connor (2010, 2011); Prokhorenko (2012); Ranganath (2011); Rodríguez et al. (2013); Ryan and Scudiere (2008); Schatz and Abdelshafi (2005); Schnitter and Mackert (2011); Seffernick (2007); Silva and Doss (2007); Smits and Rilliet (2011); Spayd (2003); Sutherland and Frohman (2011); Vlaanderen et al. (2012)

Table 2.1 (continued on next page)

New papers deemed relevant by title and abstract	13	
Papers deemed relevant by full-text	6	Heikkilä et al. (2015); Moe et al. (2014); Paasivaara et al. (2014a); Paasivaara and Lassenius (2014); Paasivaara et al. (2014b); Sekitoleko et al. (2014)

Systematic literature review by Razavi et al

Stage	**No. of papers**	**References**
Total number of papers	34	
Papers deemed relevant by full-text	9	Beavers (2007); Benefield (2008); Cloke (2007); Laanti (2008); Lee (2008); Mencke (2008); Moore and Spens (2008); Ryan and Scudiere (2008); Seffernick (2007)

Combined results

Total number of papers	51	

Table 2.1: Papers in the literature review

2.2.1 Common challenges

This literature study found several common challenges in implementing and maintaining scaled agile development. For a challenge to be regarded as common, it was required to be mentioned in at least three different sources. These challenge groups are described in the following sections. Other identified challenges are presented in their own section. The challenges are listed in Figure 2.2 along with references.

2.2.1.1 Integrating other functions of the organization

In order for an organization to be fully agile, all its functions need to adopt agile values. Several papers described difficulty in incorporating these functions into the scaled agile process. Problems became apparent when trying to include functions like marketing (Abdelnour-Nocera and Sharp, 2007, 2008; Beavers, 2007; Benefield, 2008; Maples, 2009; O'Connor, 2011; Rodríguez et al., 2013; Schnitter and Mackert, 2011; Smits and Rilliet, 2011), human resources (Atlas, 2009; Benefield, 2008; O'Connor, 2011; Rodríguez et al., 2013; Spayd, 2003), sales (Maples, 2009; Smits and Rilliet, 2011), legal (Maples, 2009), finance (Maples, 2009) and customer support (Smits and Rilliet, 2011).

The most common problem was aligning the marketing department with a scaled agile process. Abdelnour-Nocera and Sharp describe how marketing did not see value in creating user stories. The user stories were seen as duplicate work to high level marketing requirement documents (Abdelnour-Nocera and Sharp, 2007, 2008).

2.2.1.2 Incorporating user-centered design processes

Popular agile models (such as Scrum or XP) do not describe how user-centered design process should be incorporated. According to Benefield, introducing an incremental process triggered resistance from the user experience people (Benefield, 2008). Similarly, Mun-Wai and Drummond describe how it was problematic to maintain a holistic view of the user experience while working in iterations (Chung and Drummond, 2009).

Federoff and Courage claim that there is a lack of literature guidance on including user-centered processes into agile. They also regard the incremental nature of agile processes as problematic to maintaining a holistic view on a product's user experience. In addition, the fast pace of agile development makes it difficult to identify target users' needs in time. User experience specialists were initially assigned to multiple teams, which caused problems in prioritizing work from different teams. (Federoff and Courage, 2009)

2.2.1.3 Requirements management of large and complex products

The most common challenge found in the literature study was maintaining efficient requirements management in a scaled agile environment. When multiple teams are involved, the software products tend to be large and complex by nature. Some organizations were using marketing requirement documents or similar tools to manage high level requirements. Mapping these high level requirements to small enough, implementable and unambiguous user stories proved to be problematic for many organizations.

Several papers identify the root cause of requirement management issues as a gap between short and long term planning (Abdelnour-Nocera and Sharp, 2007, 2008; Beavers, 2007; Chung and Drummond, 2009; Cloke, 2007; Cowan, 2011; Gat, 2006; Laanti, 2008; Maples, 2009; Mencke, 2008; Schnitter and Mackert, 2011; Smits and Rilliet, 2011). For example, Gat describes how it is difficult to maintain the right level of requirement details during the development lifecycle (Gat, 2006). In some cases, the gap resulted from ambiguous high level requirements (Beavers, 2007; Cowan, 2011; Mencke, 2008; Smits and Rilliet, 2011).

Sekitoleko et al describe how assigning related tasks to different teams created unexpected dependencies. These dependencies needed to be resolved by adding more tasks to the team backlogs, which in turn required reprioritization of the tasks in the team backlogs. Thus, dividing the work between teams resulted in a significant amount of extra overhead. (Sekitoleko et al., 2014)

Heikkilä et al present a case in which software developers were unavailable to support project planning. This was problematic since the developers' estimates were considered crucial for performing cost/benefit analysis for the candidate requirements of the project. In addition, the case organization experienced difficulty in monitoring the progress of the software development during the project. (Heikkilä et al., 2015)

According to Prokhorenko, the developers are not able to interact with all the stakeholders in large projects. This makes high level requirements elicitation difficult when the stakeholders are not accessible (Prokhorenko, 2012). On the other hand, Dieste et al describe a case where the requirements elicitation was too laborious. One of their case study organizations did quarterly, one week long kick-offs with user story identification, planning and estimation. This was ultimately regarded as too exhaustive for the product owners and the teams did not get any development done for one week. (Vlaanderen et al., 2012)

2.2.1.4 Top-down approach creates resistance towards change

Agile transformation is a major change that has the potential to cause a backlash. If the change is introduced by a top-down approach, change resistance can occur (Abdelnour-Nocera and Sharp, 2007, 2008; Evans, 2008; O'Connor, 2010; Spayd, 2003). Abdelnour-Nocera and Sharp suggest that a top-down approach can induce change resistance as the reasons behind the change are not understood at the lower level. This is also supported by Evans, who adds that developers may feel that agile will be replaced with something else at any time (Evans, 2008). O'Connor claims that a top-down decision without internal team support is unlikely to be successful. (O'Connor, 2010)

Moreover, Spayd believes that a top-down approach to agile transformation can only be partially successful. According to Spayd, agile needs grassroots level buy-in as it is collaboration-oriented (Spayd, 2003).

2.2.1.5 Not understanding the agile values behind practices

Some organizations saw agile through the used practices and ignored the core values. In the case organization analyzed by Abdelnour-Nocera and Sharp, agile practices were used at the team level, but ingrained old processes coexisted in the higher levels of the organization. In addition, the management did not see value in agile requirements management and chose to stick with very high level marketing requirement documents. (Abdelnour-Nocera and Sharp, 2007, 2008)

Similarly, Bang describes how agile values were not understood and practices were carried out without understanding the underlying purposes (Bang, 2007). The lack of understanding also resulted in a tendency to strip some practices and enhance others in certain cases (Lewis and Neher, 2007; Vlaanderen et al., 2012). In one organization, agile was just viewed as a new project management tool (Smits and Rilliet, 2011).

Focusing on implementation details of practices instead of core values was problematic for two case organizations (Hanly et al., 2006; Long and Starr, 2008).

2.2.1.6 Inter-team communication and coordination

The second most frequently mentioned challenge is inter-team communication and coordination. Communication and coordination issues between teams seem to be major hindrances to making scaled agile efficient in the studied organizations (Beavers, 2007; Berczuk and Lv, 2010; Chung and Drummond, 2009; Cloke, 2007; Farrow and Greene, 2008; Gat, 2006; Hansen and Baggesen, 2009; Laanti, 2008; Lee, 2008; Lewis and Neher, 2007; Moore and Spens, 2008;

Ranganath, 2011; Spayd, 2003; Moe et al., 2014; Paasivaara and Lassenius, 2014; Sekitoleko et al., 2014).

The lack of communication and coordination can be responsible for issues in managing dependencies and integration of components (Chung and Drummond, 2009; Cloke, 2007; Gat, 2006; Laanti, 2008; Lewis and Neher, 2007; Moore and Spens, 2008), unclear responsibilities between teams (Beavers, 2007), duplicate work (Lee, 2008), not seeing the bigger picture (Berczuk and Lv, 2010; Farrow and Greene, 2008; Moore and Spens, 2008) and low level of collaboration (Lewis and Neher, 2007; Moore and Spens, 2008; Ranganath, 2011; Rodríguez et al., 2013; Spayd, 2003; Moe et al., 2014).

Hansen and Baggesen explain how lack of inter-team collaboration caused team centricity. This resulted in negative phenomena like fragile architecture, different code styles and distrust between teams. (Hansen and Baggesen, 2009) Likewise, Moore and Spens describe how focusing entirely on their own goals and avoiding external distractions caused teams to be isolated from each other and program level activities (Moore and Spens, 2008). Farrow and Greene suggest that autonomous teams may not think about the bigger picture. (Farrow and Greene, 2008) Cloke describes a case of internal segmentation in which teams operated with differing priorities and agendas.(Cloke, 2007)

Berczuk and Yi explain how optimizing between team goals and product goals proved to be a challenge. According to them, teams with too much focus on own goals created strong boundaries and thus created impediments for coordination and collaboration. On the other hand, some teams worked well with other teams, which resulted in achieving product and organizational goals. (Berczuk and Lv, 2010)

Inter-team communication problems are greatly amplified in a geographically distributed environment. Several papers describe how distributed teams experienced issues in collaborating and communicating effectively. (Beavers, 2007; Hansen and Baggesen, 2009; Lewis and Neher, 2007; Smits and Rilliet, 2011; Moe et al., 2014)

Moe et al describe a case in which the organization scaled its software development department by adding new offshore teams. In this case, the new teams were less likely to collaborate with each other than their counterparts in the onshore location. (Moe et al., 2014)

Paasivaara and Lassenius present a case in which Scrum of Scrums failed as a inter-team coordination mechanism. According to them, the Scrum of Scrums practice was removed as the teams did not share enough common interests. (Paasivaara and Lassenius, 2014)

2.2.1.7 Diverged agile models between teams

As agile promotes continuous improvement, the initial agile implementations are likely to evolve over time. Mun-Wai and Drummond describe how some teams that started with Scrum diverged into a mixture of Scrum and other methodologies. In addition, these teams discarded practices when pressured by release deadlines. (Chung and Drummond, 2009)

Goos and Melisse present a case where stress caused people to revert back to the old, pre-agile ways of working (Goos and Melisse, 2008). Lewis and Neher describe a case where teams had the tendency to strip some practices while enhancing others (Lewis and Neher, 2007). Dieste et al describe how practices quickly diverged in the case organizations as choices were made without having a clear motivation and reasoning. (Vlaanderen et al., 2012)

Maciaszek and Loucopoulos present a case where the agile practices of teams eroded over time. This was especially pronounced in teams without a Scrum mentor. (Schnitter and Mackert, 2011)

O'Connor suggests that diverged agile models make switching teams more difficult. According to O'Connor, people tend to try introducing the practices of the previous team, which may not work in the context of the new team. (O'Connor, 2011)

Ryan and Scudiere suggest that communicating clear expectations and having non-negotiable items of an agile process prevents divergence. In their case organization, teams struggled with implementing test-driven development as the expectations were not clearly defined upfront. (Ryan and Scudiere, 2008)

Benefield (Benefield, 2008) describes how it was difficult to find a balance between prescribing a by-the-book implementation (which may put people off) and giving too much freedom in the agile methods (which may weaken core practices).

Paasivaara et al present a study of an organization that gave its teams full autonomy on deciding their ways of working. The only common process for teams was having a weekly demo and a regular retrospective. This led to teams using different process models (Scrum, Kanban, Scrumban, etc) and not having common basic concepts like sprints. Some of the interviewed team members hoped for a common framework that could be customized to fit each teams needs instead of starting from scratch. (Paasivaara et al., 2014a)

In another case study, Paasivaara et al explain how an organization needed to align multiple sites that had different ways of working. In this case, the management suggested that the lack of common values was the cause for the diverged ways of working. (Paasivaara et al., 2014b)

2.2.1.8 Lack of literature guidance on scaling agile practices

Several papers identified lack of literature guidance on scaling agile as a challenge. Scaling the existing, well-known models into the context of multiple teams proved to be problematic (Farrow and Greene, 2008; Federoff and Courage, 2009; Hajjdiab et al., 2012). Some organizations faced difficulties in choosing the initial agile model (Cloke, 2007; Long and Starr, 2008).

Smith and Salvendy suggest that there is lack of literature guidance on how to integrate user-centered design processes into scaled agile (Federoff and Courage, 2009).

2.2.1.9 Incorporating quality assurance and performance testing into the scaled agile process

Many organizations struggled with incorporating quality assurance or performance testing into the scaled agile process. Beavers explains how quality assurance no longer had any allocated time before releases in the agile process (Beavers, 2007). Smits and Rilliet describe a similar case where integrating quality assurance into a single development cycle was challenging (Smits and Rilliet, 2011). Maples describes how performance testing was always lagging one sprint behind in the case organization (Maples, 2009). Similarly, Gat explains how incorporating performance testing in the scaled agile process was challenging (Gat, 2006). Maciaszek and Loucopoulos found that there were communication issues between engineering and quality assurance (Schnitter and Mackert, 2011).

2.2.1.10 Other challenges

Sutherland and Frohman explain how high-performing agile software development teams were draining the resources of the operations and infrastructure departments. Additionally, the aggressive hiring of new software developers was putting additional strain on the scaled agile process. (Sutherland and Frohman, 2011) Seffernick also describes similar issues in the operations and infrastructure team not being able to keep up with the agile software developer teams (Seffernick, 2007).

Paasivaara et al present a case where the constant change in process was seen as challenging. One interviewee described a frustrating situation in which changes did not yield the desired effect and thus further changes were introduced to correct the course. (Paasivaara et al., 2014a)

Type of challenge	Described in
Integrating other functions of the company	Abdelnour-Nocera and Sharp (2007, 2008); Atlas (2009); Beavers (2007); Benefield (2008); Maples (2009); O'Connor (2011); Rodríguez et al. (2013); Schnitter and Mackert (2011); Smits and Rilliet (2011); Spayd (2003)
Including user-centered design processes into scaled agile	Benefield (2008); Chung and Drummond (2009); Federoff and Courage (2009)
Requirements management with large/complex products	Abdelnour-Nocera and Sharp (2007, 2008); Beavers (2007); Chung and Drummond (2009); Cloke (2007); Cowan (2011); Gat (2006); Laanti (2008); Maples (2009); Mencke (2008); Prokhorenko (2012); Rodríguez et al. (2013); Schnitter and Mackert (2011); Smits and Rilliet (2011); Vlaanderen et al. (2012); Sekitoleko et al. (2014); Heikkilä et al. (2015)
Top down approach creating resistance towards change	Abdelnour-Nocera and Sharp (2007, 2008); Evans (2008); O'Connor (2010); Spayd (2003)
Not understanding the agile values behind practices	Abdelnour-Nocera and Sharp (2007, 2008); Bang (2007); Hanly et al. (2006); Lewis and Neher (2007); Long and Starr (2008); Smits and Rilliet (2011); Vlaanderen et al. (2012)

Continued on next page

Table 2.2 – *Continued from previous page*

Type of challenge	**Described in**
Inter-team communication, team coordination, dependency management	Beavers (2007); Berczuk and Lv (2010); Chung and Drummond (2009); Cloke (2007); Farrow and Greene (2008); Gat (2006); Hansen and Baggesen (2009); Laanti (2008); Lee (2008); Lewis and Neher (2007); Moore and Spens (2008); Ranganath (2011); Rodríguez et al. (2013); Smits and Rilliet (2011); Spayd (2003); Moe et al. (2014); Paasivaara and Lassenius (2014); Sekitoleko et al. (2014)
Teams too focused on own goals instead of the bigger picture	Berczuk and Lv (2010); Cloke (2007); Farrow and Greene (2008); Hansen and Baggesen (2009); Moore and Spens (2008)
Diverged agile models between teams	Chung and Drummond (2009); Goos and Melisse (2008); Lewis and Neher (2007); O'Connor (2011); Ryan and Scudiere (2008); Schnitter and Mackert (2011); Vlaanderen et al. (2012); Paasivaara et al. (2014a)
Lack of literature guidance on scaling agile practices	Cloke (2007); Farrow and Greene (2008); Federoff and Courage (2009); Hajjdiab et al. (2012); Long and Starr (2008)
Incorporating quality assurance and performance testing into the scaled agile process	Beavers (2007); Gat (2006); Maples (2009); Schnitter and Mackert (2011); Smits and Rilliet (2011)

Table 2.2: Common challenges in implementing and maintaining scaled agile software development

2.2.2 Success factors

This literature study found several common success factors for implementing and maintaining scaled agile development. For a success factor to be regarded as common, it was required to be mentioned in at least three different sources. These success factor groups are presented in the following sections. Other identified success factors are presented in their own section. The complete list of success factors is presented in Figure 2.3 along with references.

2.2.2.1 Communities of practices

Several papers described how communities of practices (COPs) were important for continuous process improvement. Some organizations called these meetings agile communities (Atlas, 2009; Evans, 2008; Silva and Doss, 2007) or Scrum Master meetings (Vlaanderen et al., 2012). Rodríguez et al suggest that communities of practices also helped in promoting a culture of transparency in the case organization (Rodríguez et al., 2013).

Paasivaara and Lassenius present a case in which the role of COPs evolved in the organization. At first, COPs were based on roles like Scrum Master or recreating functions that did not exist in the new organization structure. In the next phase, COPs supported scaling the process and replaced the unsuccessful Scrum of Scrums practice as an inter-team coordination mechanism. Eventually, COPs became a tool for continuous improvement in the organization. (Paasivaara and Lassenius, 2014)

2.2.2.2 Change leadership

Having change leadership is regarded as success factor in many of the papers (Atlas, 2009; Bang, 2007; Cowan, 2011; Goos and Melisse, 2008; Maples, 2009; Mencke, 2008). Especially if the change was introduced to the whole organization at once, strong change leadership was seen as an important success factor (Cowan, 2011; Mencke, 2008). Even when the change was introduced using a bottom-up approach, having clear expectations and deliverables for development and business was considered a success factor (Ryan and Scudiere, 2008).

2.2.2.3 Coaching

The most common success factor described in the papers was team coaching. Having internal or external coaches helped the teams properly adopt and maintain agile practices over time. (Atlas, 2009; Benefield, 2008; Brown, 2011; Chung and Drummond, 2009; Cowan, 2011; Gat, 2006; Goos and Melisse, 2008; Hanly et al., 2006; Long and Starr, 2008; Mencke, 2008; Rodríguez et al., 2013;

Schatz and Abdelshafi, 2005; Seffernick, 2007; Silva and Doss, 2007; Smits and Rilliet, 2011; Paasivaara et al., 2014a) Mun-Wai and Drummond suggest that without coaching, teams can even be detrimental by applying practices improperly (Chung and Drummond, 2009) .

Schatz and Abdelshafi describe how external coaches were able to provide objective feedback to the teams (Schatz and Abdelshafi, 2005). On the other hand, Benefield suggests that internal coaches have the advantage of having a deep understanding of the organization (Benefield, 2008). Silva and Doss explain how the case organization used a coach pipeline model where external agile mentors train internal to-be agile coaches (Silva and Doss, 2007).

Rodríguez et al found that coaches can react to issues immediately and help the team focus on agile principles and not on the tools themselves. (Rodríguez et al., 2013)

2.2.2.4 Training

Many papers attributed at least a part of the success to agile methods training (Cowan, 2011; Fecarotta, 2008; Gat, 2006; Hanly et al., 2006; Long and Starr, 2008; Mencke, 2008; O'Connor, 2010; Seffernick, 2007; Paasivaara et al., 2014a).

Cowan describes how a company-wide roll-out of agile methods was facilitated by giving management extensive training on Scrum. However, Cowan suggests that the result could have been even better by also providing engineering with similar training. (Cowan, 2011)

Seffernick presents a case where training was able to spark the enthusiasm of product owners (Seffernick, 2007). Gat describes how training was also provided for executives to prepare them for the organizational changes needed for adopting agile (Gat, 2006).

2.2.2.5 Using pilot projects to test and evaluate changes in process

One of the most common success factors was the usage of pilot projects to test and evaluate process changes (Brown, 2011; Chung and Drummond, 2009; Cloke, 2007; Fecarotta, 2008; Gat, 2006; Lee, 2008; Maples, 2009; Nielsen and McMunn, 2005; O'Connor, 2011; Prokhorenko, 2012; Ranganath, 2011; Schnitter and Mackert, 2011; Silva and Doss, 2007).

Sillitti et al describe how extensive piloting was used to gather experiences and adjust the process for a large-scale roll-out. The successes of the pilot projects helped in justifying company-wide adoption of the new process. (Brown, 2011) Maciaszek and Loucopoulos explain how pilot projects provided valuable insights into various issues arising from agile adoption (Schnitter and Mackert, 2011).

Similarly, Gat presents a case where piloting helped convince the rest of the organization about the benefits of the change (Gat, 2006). Fecarotta describes how executive management approved the change after successful pilot projects (Fecarotta, 2008).

In some cases, the results of successful pilot projects were publicly shared in the organization. This created enthusiasm towards agile methods and willingness to change. (Prokhorenko, 2012; Ranganath, 2011)

2.2.2.6 Synchronizing practices of teams

Sillitti et al describe how a common agile delivery process was deployed to a large number of teams. Having a common collaborative agile workbench helped teams to communicate and coordinate activities. (Brown, 2011)

Evans presents a case where 10 teams were able to synchronize their ways of working. This led to successful scaling of Scrum in the case project. (Evans, 2008)

Farrow and Greene describe a case where a custom release framework was developed for addressing inter-team communication. They suggest the framework increased the level of agility and self-organization in the whole organization. (Farrow and Greene, 2008)

Ryan and Scudiere recommend having a set of standards for expectations, behaviors and deliverables when rolling out a change in the organization. According to them, this helps in assessing teams and relocating members between teams. In addition, publicly tracking how teams meet the expectations provides increased transparency to the organization. (Ryan and Scudiere, 2008)

2.2.2.7 Pragmatic customization of the agile process

Some papers emphasize the importance of small, gradual improvements to the agile process. Instead of fixating on a single approach or constantly changing everything, the process evolved through clearly motivated and justified customizations. (Federoff and Courage, 2009; Long and Starr, 2008; Maples, 2009)

Federoff and Courage describe a case where the software developers embraced the new scaled agile process while the user experience specialists resisted it. After identifying the issue through survey, the organization took steps to properly incorporate the user experience work into the scaled agile process. Through integrating user experience specialists into teams and introducing new user experience specific practices, the organization was able to significantly improve their job satisfaction. (Federoff and Courage, 2009)

Maples explains how a case organization needed to deviate from the rules of Scrum in order to scale agile in their context. Nevertheless, Maples rec-

ommends keeping the agile manifesto in mind and having an agile purist to ensure the adaptations will not deviate too far. (Maples, 2009)

2.2.2.8 Structured approach to facilitate inter-team communication

Inter-team communication was regarded as a challenge in many of the papers. The most popular approach to address the issue was using the Scrum of Scrums practice (Lee, 2008; Ryan and Scudiere, 2008; Vlaanderen et al., 2012).

On the other hand, Farrow and Greene describe a case where a custom release framework was developed for addressing inter-team communication. In this framework, the release manager was responsible for visiting teams and resolving issues in the release level. (Farrow and Greene, 2008) Gat presents a case where inter-team dependencies were resolved using specialized tools (Gat, 2006).

Maciaszek and Loucopoulos present a concept of product teams which are formed to synchronize the work of up to seven software development teams. These product teams consist of product owner, developer experts and other problem area experts. The product team is responsible for product management, assigning backlog items to teams, collecting status information and synchronizing work between teams, for example. (Schnitter and Mackert, 2011)

Moe et al describe a case in which a Technical Area Responsible (TAR) role was added to the process. The purpose of the role is to mentor and support teams while acting as bridges for cross-team knowledge. This role was assigned to the most skilled and senior software developers. According to Moe et al, the role was a crucial method for knowledge sharing when adding new teams or sites. (Moe et al., 2014)

2.2.2.9 Focus on getting the product owner role right

Hansen and Baggesen describe how the case organization tried several approaches to implementing the product owner role in a distributed setting. In the end, the extra focus on perfecting the product owner role in the process yielded better collaboration between all the stakeholders. (Hansen and Baggesen, 2009) Similarly, Paasivaara et al describe a case where a workshop was arranged to clarify the responsibilities of the Product Owner role after some initial confusion. Having the workshop and adding a Product Owner team improved the situation in the case organization. (Paasivaara et al., 2014a)

Prokhorenko explains how a case organization succeeded in scaling the agile process and the product owner role by using additional proxy product owners. (Prokhorenko, 2012) Fecarotta emphasizes the importance of having active product owners in projects (Fecarotta, 2008). Similarly, Seffernick de-

scribes the importance of having engaged product owners from the very start of the project (Seffernick, 2007).

2.2.2.10 Communication of changes

Several papers suggest that sharing positive results created enthusiasm towards the change (Prokhorenko, 2012) (Ranganath, 2011; Seffernick, 2007).

Ryan and Scudiere explain how a structured approach to communicating changes was a major success factor in the case organization. In this case, the change leadership team created a set of standards for expectations, behaviors and deliverables for development and business. They suggest that transparency in the organization was increased by publicly tracking whether teams meet the communicated expectations. (Ryan and Scudiere, 2008)

Mencke suggests that having high visibility and even over-communicating the new model were success factors in changing the organization (Mencke, 2008).

Smits and Rilliet emphasize the importance of good communication about the goals of the change. The case organization used Kotter's eight step process model (Kotter, 1996) to introduce the change. (Smits and Rilliet, 2011)

2.2.2.11 Other success factors

Several papers describe a gap between marketing and engineering when scaling agile. Some cases resolved the issue by introducing a special role of a requirements architect, who was responsible for bridging the gap between marketing and engineering by facilitating requirements management in the process. (Beavers, 2007; Gat, 2006)

Maples describes a case where the gap between marketing requirements documents and team backlogs was resolved by introducing quarterly agile roadmapping sessions (Maples, 2009). Laanti describes using experienced program managers for maintaining long-term vision while short-term strategy was based on backlogs (Laanti, 2008).

Benefield suggests investing resources in finding employees with a deep understanding of agile principles and training them to help their own teams. According to Benefield, this is key for scaling agile efficiently in large enterprises. (Benefield, 2008)

Maples emphasizes the importance of continuous efforts to maintain agile processes and culture in growing organizations. New teams, adding influential members to existing teams or scaling out the process beyond physical locations can have an impact on the culture and require re-adjustment. (Maples, 2009)

As described in the challenges section, incorporating user-centered design into scale agile process can be problematic. However, some papers present cases where the issues were resolved and user experience specialists were fully integrated into the process. (Chung and Drummond, 2009; Federoff and Courage, 2009) Unfortunately, Mun-Wai and Drummond only state that a team was able to integrate user experience work into Scrum, but do not expand on it (Chung and Drummond, 2009).

Federoff and Courage present a case where the initial resistance and skepticism of user experience personnel was overcome by introducing three changes. First, the number of Scrum teams per one user experience designer were reduced from four to two. The user experience designers also agreed to spend a couple hours per week supporting the teams that did not have a dedicated user experience designer. Second, the software development and design processes were parallelized so that the development work is not stalled by waiting for design or vice versa. Third, the usage of the Rapid Iterative Test and Evaluation (RITE) process was used to evaluate the designs. The RITE process allowed for faster reaction to issues in the user interface whereas traditional usability testing would have required surveying all the participants. (Federoff and Courage, 2009)

Moore and Spens have identified several important qualities of high performing teams in their case organization. High performing teams were thinking and operating outside the team walls, trusting others to make good decisions and being vocal and raising concerns up to the program level, for example. (Moore and Spens, 2008)

Paasivaara et al describe how value workshops were an effective practice for creating a common vision for a distributed organization. In addition, the workshops helped in connecting stakeholders across the geographically distributed sites. (Paasivaara et al., 2014b)

Heikkilä et al suggest that organizing a Release Iteration Planning event is beneficial as it brings together the development organization and the product management. According to them, the face-to-face communication in the planning events enables fast dependency identification and management. (Heikkilä et al., 2015)

Type of success factor	Described in
Communities of practices	Atlas (2009); Evans (2008); Rodríguez et al. (2013); Silva and Doss (2007); Vlaanderen et al. (2012)
Change leadership	Atlas (2009); Bang (2007); Cowan (2011); Goos and Melisse (2008); Maples (2009); Mencke (2008); Ryan and Scudiere (2008)
Coaching	Atlas (2009); Benefield (2008); Brown (2011); Chung and Drummond (2009); Cowan (2011); Gat (2006); Goos and Melisse (2008); Hanly et al. (2006); Long and Starr (2008); Mencke (2008); Rodríguez et al. (2013); Schatz and Abdelshafi (2005); Seffernick (2007); Silva and Doss (2007); Smits and Rilliet (2011)
Training	Cowan (2011); Fecarotta (2008); Gat (2006); Hanly et al. (2006); Long and Starr (2008); Mencke (2008); O'Connor (2010); Seffernick (2007)
Using pilot projects to test and evaluate changes	Brown (2011); Chung and Drummond (2009); Cloke (2007); Fecarotta (2008); Gat (2006); Lee (2008); Maples (2009); Nielsen and McMunn (2005); O'Connor (2011); Prokhorenko (2012); Ranganath (2011); Schnitter and Mackert (2011); Silva and Doss (2007)
Synchronizing practices of teams	Brown (2011); Evans (2008); Farrow and Greene (2008); Ryan and Scudiere (2008)
Pragmatical customization of agile model	Federoff and Courage (2009); Long and Starr (2008); Maples (2009)
Structured approach to facilitate inter-team communication	Farrow and Greene (2008); Gat (2006); Lee (2008); Ryan and Scudiere (2008); Schnitter and Mackert (2011); Vlaanderen et al. (2012); Moe et al. (2014)

Continued on next page

Table 2.3 – *Continued from previous page*

Type of success factor	**Described in**
Focus on getting product owner role right	Fecarotta (2008); Hansen and Baggesen (2009); Prokhorenko (2012); Paasivaara et al. (2014a)
Communication of changes	Mencke (2008); Prokhorenko (2012); Ranganath (2011); Ryan and Scudiere (2008); Seffernick (2007); Smits and Rilliet (2011)

Table 2.3: Success factors in implementing and maintaining scaled agile software development

Chapter 3

Background

3.1 Rally Software

Rally Software is a multinational corporation that focuses on agile software products and corporate agile training. Rally has two software engineering offices in the United States and one in Finland. Besides these offices, Rally Software also has several sales offices across the globe. Rally employed over 500 people as of June 2015. The overall organization structure of Rally Software is presented in Figure 3.1.

Rally Software's core product is the Agile Lifecycle Management (ALM) platform which covers various aspects of managing an agile company, ranging from agile project management to agile product portfolio management. Rally also offers other complementary products like Flowdock, a team collaboration software product.

In addition to the above software products, Rally provides consultation services for enterprises seeking to adopt agile methods. These services consist of organization transformation consulting and agile training and coaching. For enterprises going through agile transformation, Rally recommends Scaled Agile Framework as a solution to supporting agile software development within the enterprise.

3.1.1 Organization structure

At the beginning of the case study, Rally was a product line organization with several different product lines. The largest product lines were built around existing products and their supporting functions. These three product lines were ALM, Team Collaboration (Flowdock) and Product Portfolio Management (PPM). The Insights product line was focused on developing new products which utilize a customer's own data from the ALM platform to bring

them additional value. Besides these product lines, Core Services and Operations existed to provide common infrastructure and services to support the other product lines.

During the case study, the product lines were removed by the new Rally management. Instead, the organization was restructured around the main Agile Lifecycle Management product in order to eliminate communication silos between the teams.

Rally Software has been organized according to the Scaled Agile Framework. Rally chose the Scaled Agile Framework due to being a fairly large software development organization with multiple products. The rationale behind choosing SAFe was to align the organization according to a common vision while still supporting agile values and practices.

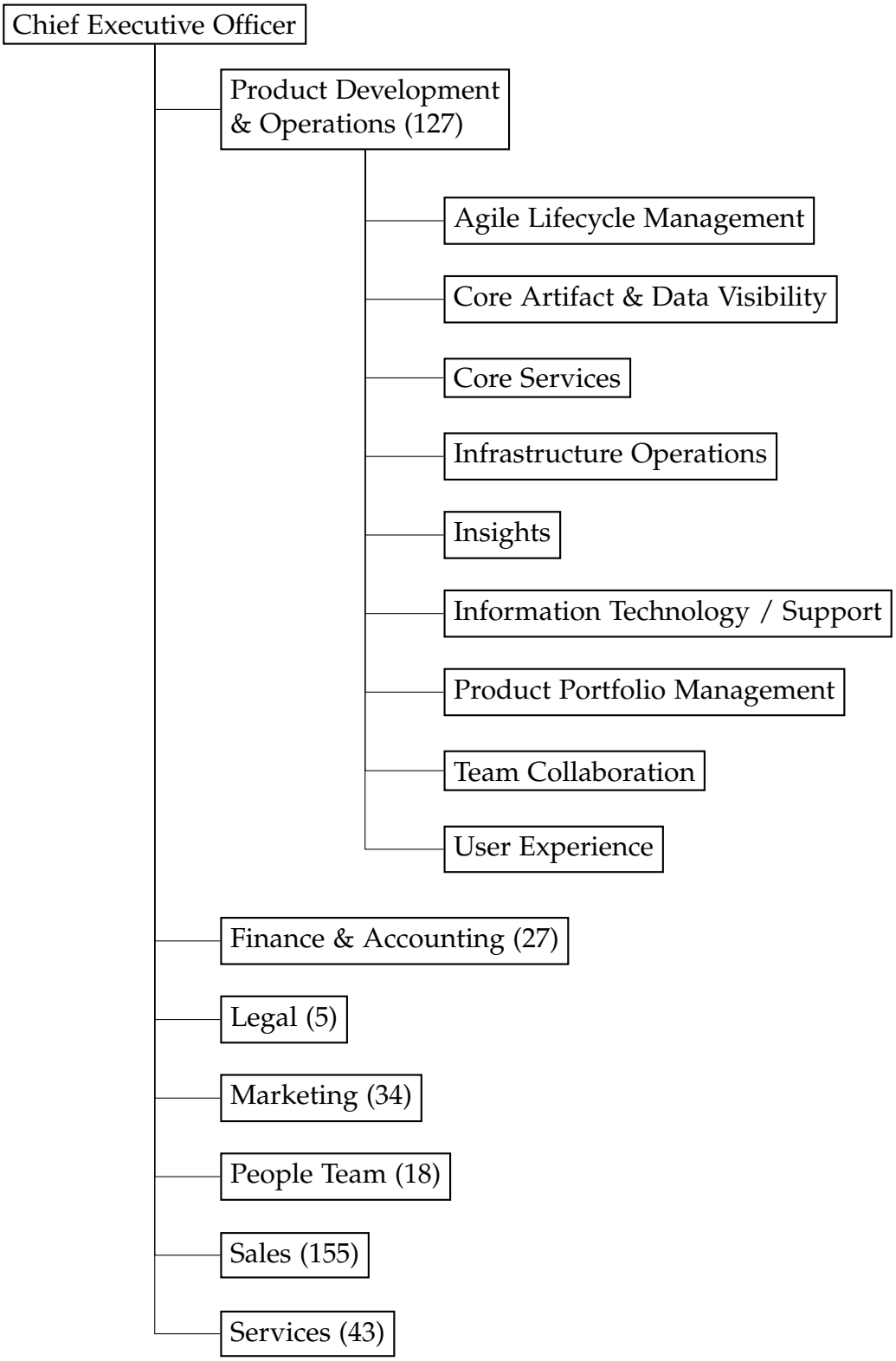

Figure 3.1: The organization chart of Rally Software as of May 2014. The number in parentheses is the number of employees in the particular branch. This thesis focuses on the product development and operations branch.

3.2 Agile survey

In order to evaluate its effectiveness as an agile organization, Rally Software employees participated in a survey during the spring of 2014. The survey was performed by an internal agile transformation consultant using the Six Boxes Model (Performance Thinking Network). The interviews ranged from 30 to 60 minutes in length. The Six Boxes Model divides survey questions into the following categories:

1. Expectations and feedback
2. Tools and resources
3. Consequences and incentives
4. Skill and knowledge
5. Selection and assignment
6. Motives and preferences

In the survey, the interviewees were divided into three groups: execution, planning and tertiary. The execution and planning groups divided the engineering organization employees into two groups based on the employee's primary function. The tertiary group included employees from human resources, accounting and legal department.

3.2.1 Survey results

The survey identified several issues in the organization and practices. First, there was very little cross-product vision. The communication on cross-cutting initiatives was limited. In addition, the initiatives tended to be product-specific rather than spanning across the portfolio of products. These boundaries between product lines were described as unhealthy for future collaboration across the product lines and with larger scale initiatives. The survey also found a lack of focus on goals beyond the very short term.

Second, the product line organization structure was not what it claimed to be. The survey described the existing product lines as neither products nor lines of business. Consequently, the structure of product lines had not produced any desirable results in the organization. Moreover, there was considerable management overhead in Product development and Operations according to the survey results.

Third, agile activities had degraded to the lowest common denominator. This was influenced by limited agile experience in product development. Being a company that provides agile consultancy services and software products, employees tended to overestimate their own agile expertise. In addition, the aggressive hiring of new employees had contributed to the degradation of agile practices. However, some teams had restored their agile practices to a state that was very close to the textbook version of Scrum. Those teams also had the highest success rate in delivery.

Fourth, the focus on improvement was slipping. Linked to the gradual fading of agile practices, the activities for process and practice improvement were in decline. Retrospection was limited and often not adding any value. The survey also identified resistance to improving processes and practices.

Fifth, the survey found elements that hindered innovation in the organization. For example, there was a communication gap between user experience and product development teams. This resulted in missed innovation and customer opportunities.

3.2.2 Recommended actions

The survey results were analyzed by the same agile transformation consultant who performed the interviews and gathered the data. This analysis suggested multiple actions for transforming and improving organization at different levels. The recommendations for team, program and portfolio levels were as follows.

At the team level, the analysis suggested that software development teams should move back to basics in their agile practices. The teams were using some agile practices, but had discarded complete process models over the time. The analysis suggested implementing Scrum or a similar iterative process model in full, and gradually customizing it according to the team's needs. The Scrum Masters should be tested and receive additional training if needed. In addition, the teams should pay attention to properly using retrospectives as an issue escalation path.

At the program level, the analysis recommended keeping separate roadmaps for products and using initiatives for cross-cutting features. These roadmaps should be coordinated on the portfolio level with initiatives derived from vision, customer feedback and cross-product collaboration. Furthermore, the planning activities of products should be consolidated and not just synchronized.

At the portfolio level, initiatives should only be used for cross-cutting issues. Consequently, any product-specific issues should be addressed in individual product roadmaps instead of creating initiatives for them. In general,

initiatives should be about improving the whole organization and its portfolio of products.

3.3 Agile transformation

Rally profiles itself as a company that embraces the agile values. However, the agile survey revealed that the organization was not fully working in an agile manner. A long lead time to market and the resulting market uncertainty were identified as issues in the current software development process.

External pressure was also among the reasons for a need for change. The company stock price had been in decline since late 2013 due to not meeting sales expectations of the market. Moreover, during this time, the competition had been able to increase their sales.

Due to these internal and external motivators, Rally Software decided to improve its product development process drastically. To achieve this goal, Rally launched an agile transformation program in late spring of 2014. The goal of the transformation process was to address the issues found in the survey, like degraded agile practices of software development teams and a suboptimal organization structure.

As an immediate change, all future quarterly planning and steering events were to be held in person with both engineering and business representatives. All the stakeholders would get together at the company headquarters in Boulder, Colorado in order to ensure effective planning activities.

This thesis follows Rally Software's progress from the beginning of the transformation to the post-transformation stage.

Chapter 4

Scaled Agile Framework at Rally Software

Scaled Agile Framework (SAFe) is a framework for adopting agile practices at enterprise scale. The framework was created by Dean Leffingwell and described in his book Agile Software Requirements (Leffingwell, 2010). Leffingwell claims it has been successfully applied in programs with 50-100 people and in enterprises employing thousands of software developers (Leffingwell LLC, a).

The framework describes how the enterprise should be organized to support effective agile software development. It aims to address different concerns of product portfolios, individual products and teams by placing them into separate, vertical levels. These levels are described in depth in the following sections of this chapter and compared to the implementation in the case organization. The information presented in this thesis is based on the version 2.5 of the framework. The big picture view of the framework is illustrated in Figure 4.1.

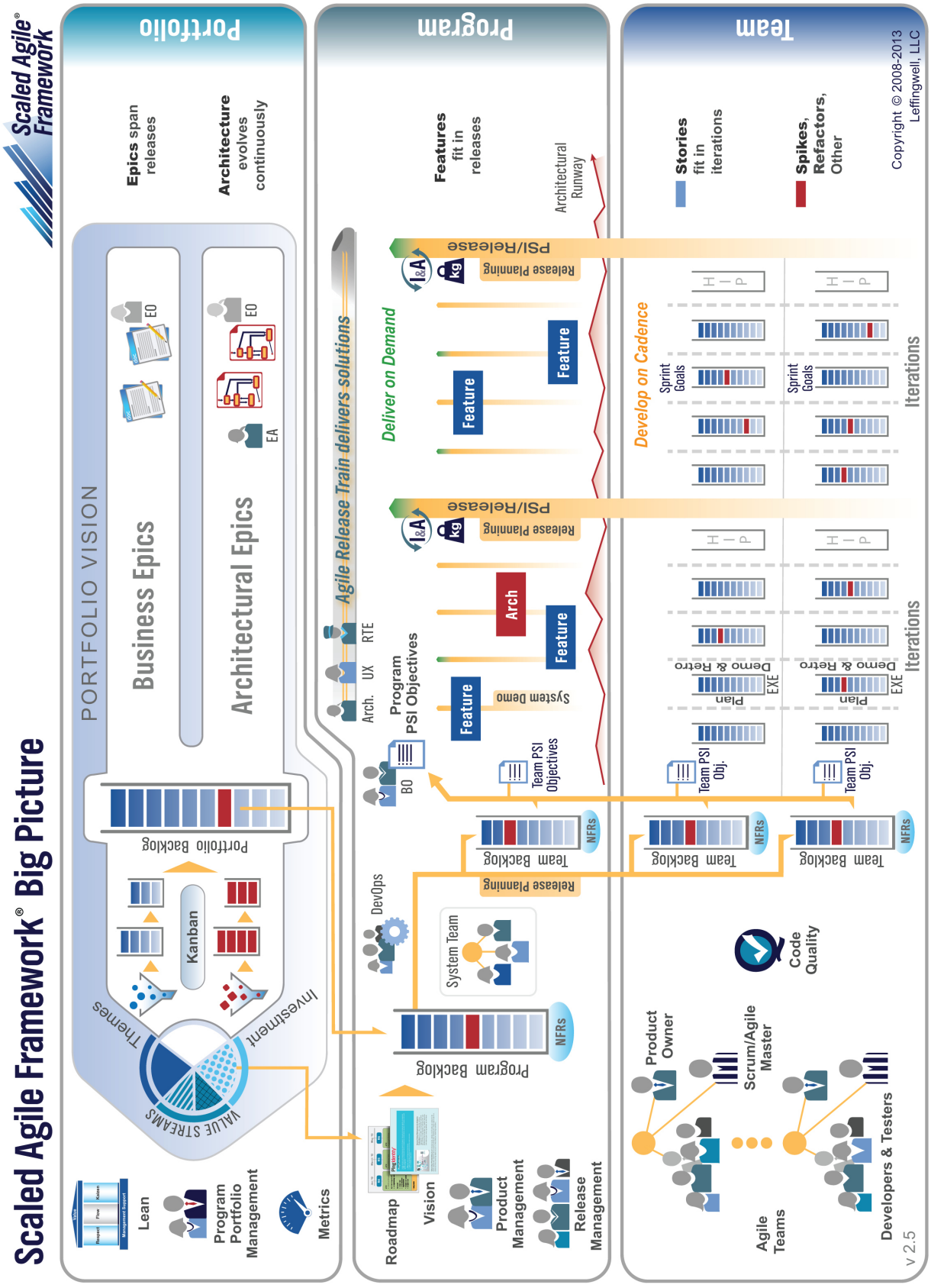

Figure 4.1: The big picture view of SAFe (version 2.5). Figure reproduced with permission.

4.1 Portfolio level

The topmost level is the Portfolio level. This level represents a portfolio of related software engineering programs and the enterprise business strategy that is used to align them. The overall business strategy is outlined in the Portfolio Vision.

4.1.1 Portfolio vision

The Portfolio vision is the steering force at the highest level of the framework. It represents the set of principles and decisions which drive an instance of SAFe within the enterprise. To manage the portfolio in a lean and agile manner, the Portfolio vision is divided into five additional constructs using practices from both methodologies. These constructs are described below.

4.1.2 Value stream

A value stream represents software engineering effort that generates value to the business or the customer. The set of value streams within an enterprise directs investment decisions on products and services. The lean-management method of Value Stream Mapping can be used to analyze and improve value streams.

In the past, the case organization's value streams were its product lines. As the product lines were dismantled during the summer of 2014, the value streams now originate directly from a single product that comprises of the Agile Lifecycle Management tool and Flowdock. The implication of this change is that there is no longer a portfolio in the sense that Scaled Agile Framework defines it. Instead, there is only one product and one Release train for it.

4.1.3 Investment themes

Investment themes represent set of initiatives within a value stream that drive the investments of the enterprise. Investment themes are strongly connected to the value propositions of individual products or services. Value propositions describe competitive advantages and marketplace differentiation of products and services. Thus value propositions provide a set of potential investment themes for the enterprise.

Rally Software starts the portfolio steering process by evaluating market opportunity to determine its investment themes. Rally does this through annual planning session. The investment themes are also aligned to the budget. The themes are inspected and adapted in quarterly company steering sessions.

4.1.4 Epics

Epics are enterprise initiatives that have business and technology impact and potential return on investment. Epics span releases and can affect multiple programs in the portfolio. These initiatives are typically used to enhance business value across the whole portfolio.

SAFe defines two types of epics: architecture and business epics. Architecture epics focus on improving the technological solutions in the portfolio. They are used to support current and future business needs in the architecture of the portfolio. Business epics are customer-facing initiatives. They aim to realize certain business benefit through new development.

Rally uses both architecture and business epics in its portfolio-level process. The epics are called initiatives in the case organization's vocabulary.

4.1.5 Portfolio backlog

The Portfolio backlog is the highest-level backlog in the framework. It contains a prioritized list of epics that are waiting to be pulled into Kanban systems.

4.1.6 Kanban systems

Kanban is a software development process model with an emphasis on just-in-time delivery (Anderson, 2010). In Kanban, the team members pull work tasks from a backlog while respecting the limit on work-in-progress. In order to increase understanding among stakeholders, the work and the work flow is visualized in Kanban using a physical board, for example.

SAFe suggests using two separate Kanban systems to manage epics. Architecture and business epics are fundamentally different types and have different stakeholders. Therefore, it is logical to split them into two separate Kanban systems.

As suggested in the framework, Rally Software uses Kanban for management of the portfolio-level epics. There are two separate Kanban systems operating simultaneously at the portfolio level; one for architecture and one for business.

4.2 Program level

The second level is Program level, where the efforts of multiple agile teams are integrated to create larger value for the enterprise. On this level, the framework aims to scale value, teams and time boxes.

4.2.1 Release train

A Release train is a long-term unit that consists of multiple agile teams. It is responsible for delivering value in the Program level. Using the resources dedicated to it, the Release train continuously defines, builds, tests and delivers value to one of the value streams in the enterprise. It aligns the teams to a common mission, schedule and development rhythm.

Release trains are organized around the value streams in the enterprise. The development effort is thus used in an optimal way to deliver value and create economic benefit.

At Rally, each product line – or Release train – used to create a program vision and roadmap that ultimately drove their backlog. The product lines were mostly independent, but operated on the same cadence when it came to iterations. After the product line structure was removed, all teams started working on a single product as one Release train. The teams in the Release train are connected through a daily Scrum of Scrums meeting.

4.2.2 Roles in the agile program

In addition to the agile software development teams, SAFe defines five extra roles for the Program level. First, a Product manager is responsible for the feature content in the release train. Second, a Release management team evaluates and approves built solutions for release to customers. Third, a System team is responsible for system integration. It uses continuous integration, automated test suites and end-to-end testing to provide timely feedback for developers and stakeholders. Fourth, a Release train engineer is responsible for keeping the release train on tracks. Fifth, DevOps is included into the Release train to ensure successful deployment and operation of built software. DevOps consists of personnel from the Operations team.

4.2.3 Program backlog

The Program Backlog consists of all the anticipated future work of the program. The backlog contains future features for end users as well as architectural items to enable development of upcoming features. It is driven by both customer feedback and epics in Portfolio Backlog.

4.2.4 PSIs and releases

A Potentially shippable increment (PSI) is a milestone in the software's lifecycle. It is a deliverable increment of functionality that has been fully tested and

integrated. A PSI can also be a release when it is made available to external users.

4.2.5 Release planning

Release planning is the synchronization point of the release train. It aims to establish a common vision and commitment to release objectives for the next period. The main outputs of this planning process are Team PSI objectives for individual teams, Program PSI objectives and a PSI plan identifying milestones during the period.

At Rally Software, a R&D steering meeting is held quarterly in order to establish a prioritized backlog of Rally product enhancements for the next quarter. The updated backlog informs the following PSI planning session. In the PSI planning session, the teams break down items from the prioritized backlog into user stories for the next quarter.

4.2.6 Inspect and adapt workshop

A program conducts an Inspect and adapt workshop at the end of each release or PSI. The purpose of this workshop is to find ways of improving the program and thus maintain continuous improvement on the program level.

Inspect and adapt workshops are currently not organized in the case organization. However, the quarterly steering meetings have a dedicated part for discussing lessons learned and reflecting on the past quarter. The outputs of the steering meeting provide valuable information for continuous improvement on the program level. Thus a separate workshop for continuous improvement has been deemed unnecessary.

4.3 Team level

The third level is the Team level, which is made up of individual software engineering teams and their practices. The framework provides the agile teams with organization, role and process models. The teams are responsible for executing agile project management and technical practices, which are mostly adopted from Scrum and XP.

4.3.1 Agile team

An Agile team consists of a maximum of 7 to 9 members. Each team member has one of the three distinct roles defined for an agile team in the framework.

The first role is the Product owner, who is responsible for maintaining the product backlog by eliciting user requirements and prioritizing them. The second role is a Scrum/Agile master, who facilitates agile software development and helps the team maximize its productivity. The final role, Developers and testers, form the backbone of the agile team. They define, build, test and deliver user stories for the product.

At Rally Software, most teams use Scrum as their process model after the transformation. Scrum practices are also promoted by the external coaches that have been helping Rally restore agile practices in the engineering organization. However, previously most teams used Kanban and one team is still using a Kanban-inspired model instead. All the teams have a Product owner and Scrum master as suggested by the framework. The number of members in a team ranges from 4 to 9 at the case organization.

4.3.2 Iterations

Agile teams use time boxed development cycles to develop and deliver new functionality. These cycles are called iterations in the framework and they are comparable to sprints in Scrum. All the teams in the framework share the same iteration length and start and stop their iterations synchronously.

Iterations repeat the same standard pattern to produce value as increments of functionality. The iteration starts with planning and committing to build some functionality. The bulk of the iteration is used to build functionality and test it. At the end of the iteration, the team demonstrates created functionality to the stakeholders. Before repeating the cycle, the team holds a retrospective meeting to reflect on the past iteration and how to improve future iterations.

At Rally Software, the Scrum teams have set their iterations to two weeks and the cycles are synchronized. The one team using Kanban is aligned with the two week sprint cadence of the Scrum teams.

The whole company gathers together for iteration demos on every other Friday. In the iteration demo, each team can present what they have delivered in the past iteration or during several iterations. The teams are encouraged to concisely demonstrate only those features that are of importance to the large audience of stakeholders present.

4.3.3 User stories

User stories describe customers' requirements for the product. The agile teams use user stories to implement functionality that provides value to the user. User stories are the smallest units that flow in value streams, eventually reaching the customer.

SAFe recommends using the user voice form of expression in user stories: "As a ⟨user role⟩ I can ⟨activity⟩ so that ⟨business value⟩".

At the case organization, user stories are written in a free-form manner. The user stories are stored and managed in Rally's own instance of their Agile Lifecycle Management product.

4.3.4 Team backlog

The Team backlog contains all the user stories that the team has planned to implement. Team backlogs can also include defects, refactoring and infrastructure work that the team has identified. Each team has their own backlog and it is managed and prioritized by the Product owner of the team.

At Rally Software, the team backlogs are managed by the respective Product Owners using the Agile Lifecycle Management product.

Chapter 5

Research methods

The three most commonly used research methods are quantitative, qualitative and mixed methods research. Creswell defines qualitative research as an approach for exploring and understanding the meaning individuals or groups attribute to a social or human problem. In contrast, quantitative research is an approach for testing objective theories while examining the relationship among variables. (Creswell, 2013)

Denzin and Lincoln define qualitative research as focusing on the qualities of entities and on processes and meanings that are not experimentally examined or measured in terms of quantity, amount, intensity, or frequency. In contrast, quantitative studies emphasize the measurement and analysis of causal relationships between variables, not processes. (Denzin and Lincoln, 2011)

In this thesis, we seek answers to open-ended questions about the case organization, and do not possess any theories about the case organization. Thus, the qualitative method was chosen for the empirical part of the thesis. We chose interviews as the specific qualitative method since they give the interviewer an opportunity to ask follow-up questions and allow the interviewee to provide detailed answers.

We conducted two rounds of interviews at Rally Software. The first interview round was performed amidst the organizational changes in July 2014. The goal of the first interview round was to establish a view on current issues in the case organization. After the transformation was completed, we performed the second round of interviews in April 2015. This round of interviews was aimed to evaluate how the transformation had affected the case organization.

5.1 Research process

The research process of this case study roughly follows the case study protocol described by Yin (Yin, 2013).

We have chosen a holistic single case study for this thesis. The rationale for this decision is that we could observe the case organization at two points of time, thus enabling a longitudinal case study. In addition, the case organization represents a rather typical large software development organization that has been using scaled agile methods for a long time.

The list of questions used in the interviews was derived from a case study of agile transformation. The transformation case study focused on change from a waterfall to an agile process. Thus, we omitted questions related to the waterfall to agile change and selected questions that were applicable to the case organization's status as a mature agile organization. In addition, a few questions about agile at the organizational level were added to the list. For the second round of interviews, the questions were slightly modified to focus on changes since the last interview.

We chose to approach the product owners of six teams working on the Agile Lifecycle Management tool. Five of the product owners agreed to be interviewed. We conducted both rounds of interviews with the same participants. The first round of interviews was performed face-to-face in the Boulder office. The second round of interviews was conducted via teleconference calls. In order to maintain a chain of evidence, all the interviews were recorded and the answers to the questions were summarized in textual form.

The presented case study results were obtained by using pattern matching and explanation building techniques in the analysis (Yin, 2013). The results of the analysis were also reviewed by the interviewees.

5.2 Interviews

We performed two sets of interviews to gain a longitudinal view into the case organization's evolving scaled agile process. The first set of interviews was held in July 2014 and the second one in April 2015. The interview questions for both interview rounds are included in the appendices.

The interview results were obtained through the following steps. First, we summarized the recorded interviews as text. The summarization process produced a text version of the main points of the interviewee's answers as well as some relevant quotes. Next, we grouped all the text answers into a single answer sheet along with information on who is the author of the particular answer. In this stage, we also identified answers that were relevant to other

questions and appended those answers under other relevant questions. Finally, we performed pattern matching on the answers of different interviewees and grouped similar ones into a larger theme. The results are presented in the next chapter.

5.2.1 Interviewees

Five employees from the engineering department of Rally Software were interviewed. These employees represent five of the six teams working on the ALM product. Thus the interviewees are likely to provide a solid overall view into the development of the ALM product. In addition, the interviewees interact with the other functions of the organization, which should give them a perspective on larger scale issues within the organization.

The interviews were roughly one hour long, depending on the the interviewee. Each of the interviewees answered the interview questions listed in the appendices. The interviewees were also given time to address any additional issues they had on their mind.

The roles and backgrounds of the interviewees are presented in Figure 5.1.

Role	**History at Rally**	**Education**
Product Owner	2 months	Degree in Television production
Product Owner	1 month	B.Sc. degree in Information Science
Product Owner	2.5 years	B.Sc. degree in Computer Science
Product Owner	3 years	B.Sc. degree in Computer Science
Product Owner	6 years	M.Sc. degree in Computer Science

Figure 5.1: Backgrounds of the interviewees

Chapter 6

Interview results

The results of the Rally Software employee interviews are presented in this chapter. The findings of the interviews have been grouped into sections in order to provide a comprehensive view on larger themes.

6.1 Results of the first interview round

The first interview round was performed during the transformation process. The findings are grouped into 5 themes that are presented in the following sections. The issues are summarized in the Figure 6.1.

Viewpoint	Found issues	Improvement suggestions
Agile practices of teams	• Degraded practices • Lack of discipline in following practices	• External agile coaches • Enforcing practices
Scaled agile at the organizational level	• Constant change • Product line organization structure that created communication silos • Lack of discipline in following the Scaled Agile Framework • Fitting work into three month (quarterly) time boxes is suboptimal • Business stakeholders and the product organization are not connected enough	• Smaller, incremental changes • Measuring the effect of changes • Removing product lines • Enforcing SAFe • Recognizing that features rarely match to three month time boxes
Trust	• Lack of trust between teams • Skepticism about the ability of other teams to deliver • Constant changes and drifting priorities create distrust towards management	• Reduce rate of change • Give teams more autonomy to make decisions outside of planning meetings • Trust teams' ability to deliver • Enable teams to give each other honest feedback
Feedback	• Lack of cross-team feedback mechanisms	• Enable teams to give each other honest feedback
Technology	• Difficulty in delivering customer value and improving aging technology foundations at the same time	• Give engineering more resources to improve the foundations of the product

Figure 6.1: Summary of results from the first round of interviews

6.1.1 Agile practices of teams

The teams of the interviewees used Kanban and Scrum as their software development processes. Of the five teams represented in the interviews, four teams used Kanban and one team used Scrum.

Rally provided all the software development teams with external Scrum Masters in June 2014. The purpose of these Scrum Masters was to restore degraded agile practices and help the teams to get back to the basics of agile software development. Even though most teams use Kanban, ”Scrum Master” was chosen as the term for this facilitator role. The Scrum Master is responsible for removing impediments to the team’s productivity and challenge them to continuously improve, as defined in the Scrum process model.

According to the interviewees, introducing external Scrum Masters and enforcing disciplined agile practices have been mostly positive changes for their teams. However, one interviewee reported that their team has shown a fair amount of friction towards these changes. In this team, changes proposed by the team itself received very little resistance whereas external propositions faced lots of pushback.

Some of the interviewees expressed concern over teams forgetting the value of certain agile practices. This had even led to some teams discarding those practices completely, deteriorating agile practices in the process. One interviewee described the end result as the team being unsuccessful with agile and having overconfidence in their own excellence. One example of deteriorating practices was one team’s retrospectives that produced no action items, thus being unproductive. When discussing team practices, one interviewee said that ”[Rally’s agile coaches] would never advocate doing this in customer organizations”.

On the other hand, another interviewee explained that the team was not trying to stick with textbook methods, and had instead let the methods evolve over time. One interviewee suggested that while engineers know the textbook answers to why certain practices are done, they need to revisit why practices are actually useful. The interviewee explained that ”if practices are not explicitly stated, they tend to fade away. You need to enforce the practice to make it work.”

According to one of the interviewees, test-driven development was not practiced much in the organization. The interviewee explained that some teams resisted and discarded the practice whereas the teams that sticked with it had less quality problems. Another interviewee mentioned that developers in the team sometimes push back. The developers might even decline to implement a task presented by the Product Owner.

One of the interviewees explained that teams are only able to work on

certain areas of the product, which causes bottlenecks.

6.1.2 Scaled agile at the organizational level

All interviewees identified constant change in processes and business priorities as a serious challenge. The interviewees felt that the organization was constantly moving and making major corrections to its course. Not having a clear status quo made it difficult to evaluate the effects of changes and whether they were beneficial or justified. The interviewees fully supported the agile value of embracing change, but also felt that introducing huge changes constantly is harmful and in contradiction with the agile values. For example, one interviewee stated that the organization is ”almost too agile”. One interviewee explained that there is tendency towards ”throwing everything away instead of keeping the good stuff” in the organization. Another interviewee described that the organization is obsessed about getting better. According to the interviewee, this is manifested in the need to experiment and have lots of meetings.

The interviewees provided several improvement suggestions for resolving the issue. The most popular suggestion was introducing smaller incremental changes in the organization instead of changing the course completely. The effect of the change should be quantified and measured against a baseline before the change is rolled out across the organization. The interviewees believed these practices would help reduce constant steering of the whole organization and provide more stability for the product development organization.

The organization structure of Rally Software received a fair amount of criticism from the interviewees. The product line based organization structure was especially regarded as suboptimal by most of the interviewees. They described product lines as harmful due to a lack of trust and communication silos between the product lines. In addition, product lines encourage focusing on a narrow set of functionality that is specific to the product line. Thus teams are not encouraged to be cross-functional, which is a key requirement for agile teams.

Some interviewees also addressed the use of the Scaled Agile Framework at Rally Software. One interviewee said that the Scaled Agile Framework was followed at a philosophical level, but not in a disciplined manner. Moreover, according to this interviewee, there was a lack of commitment from the organization as a whole. Another interviewee explained that the Scaled Agile Framework is good as a reference and working well so far, but it is more like a guideline than a strict framework.

The interviewees had some ideas on how to improve the Scaled Agile Framework implementation at Rally Software. The middle layer of the frame-

work, the Program level, could use a more concrete feature and initiative roadmap (according to one interviewee). This would provide the teams with a vision. Moreover, this interviewee explained that a longer cadence at the Program level could provide more stability to business decisions and thus avoid asking the engineering department to correct course every two weeks. Addressing the issue, the interviewee stated that "we could do better in communicating and actually having a vision for integration [of features]". As an improvement suggestion, the interviewee suggested making feature and initiative roadmaps available.

Another interviewee provided insight into issues in the planning and steering process of the organization. The PSI planning process brings out problems in the organization and forces to focus on the pain points. On the other hand, it provides vision and timeboxing for large initiatives, even though committing to delivering three months worth of work beforehand is not the most agile approach.

According to several interviewees, the organization could do a better job of connecting the business stakeholders and the product organization. For example, it is not obvious if delivered increments of work are actually helping the customers, as stated by one interviewee. Another interviewee felt that the business organization was driving the product a little more than necessary and the software development teams were not empowered enough. The interviewee suggested that teams should be allowed to make more autonomous decisions outside the planning meetings.

This interviewee also described the benefits and drawbacks of the current steering process in detail. The interviewee stated that the process is really painful, but helps in identifying dependencies between product features and improves communication in the engineering organization. The voting process clarified what people wanted to have in the product. As an improvement suggestion, the interviewee suggested that after voting, teams could have already started working on an implementation plan on their own instead of doing it with everyone in the meeting.

According to the interviewee, another suboptimal aspect of the steering process was trying to fit work into a three month time box (quarter). The interviewee suggested recognizing that the work does not really fit into such time boxes and trying to force that does not add value. Instead, the teams and developers should be trusted to get to the goal. The estimates are not going to be accurate for such a long period of time, but they give a rough idea about the feasibility of fitting a feature into a quarter.

6.1.3 Trust

All interviewees agreed that trust within teams is on a good level. Team members seem to generally trust each other based on regular one-on-one discussions between team members and the development lead of the team. All in all, the interviewees did not see issues in trust between individuals.

However, most of the interviewees had seen a lack of trust between teams. According to several interviewees, a major reason for the lack of trust between teams was skepticism towards other teams being able to deliver. For example, some team members regarded a dependency to another team's work as a major risk, and were skeptical about their own ability to deliver given the dependency. One interviewee stated that "there is a lot of distrust between product lines", which affects trust between teams working in different product lines. Another interviewee described lack of transparency and tribal culture among teams as significant reasons for distrust towards other teams. According to the interviewee, an absence of written documentation was one reason for transparency issues among the teams. Also, it was difficult for other teams to understand why delivering some increments took so much time for another team.

Part of the trust issues between teams seemed to be a discrepancy between communicating problems and getting them resolved. All interviewees felt that problems are easy to communicate, whereas getting them resolved was much more difficult. One interviewee explained that escalated problems had no clear owners and thus getting them resolved was challenging.

The interviewees expressed slightly mixed feelings about trusting management. Recently, there were significant personnel changes within the management of the engineering organization. Given the personnel changes, some interviewees felt that trust would have to be regained over time.

Constant changes in the company's course and drifting priorities were also seen as sources of friction by the interviewees. Some interviewees suggested that these factors were likely to reduce trust in management within the teams. Besides that, the interviewees did not report any significant lack of trust in management from their teams. However, one interviewee mentioned that there is distrust between the sales and engineering organizations.

To build trust across different parts of the organization, the interviewees presented several improvement suggestions. The most popular suggestion was allowing teams to make more autonomous decisions outside planning meetings and thus empowering them to do what they think is best. Additionally, management should listen to engineers more often and value their thoughts on what is important in the long term. Teams should be allowed to refactor when they see it is necessary for achieving long-terms goals.

6.1.4 Feedback

All interviewees felt that the members of their teams received enough individual feedback. Every team had a practice of having one-on-one meetings at least once a month. In these meetings, the development manager of the team and a team member give each other feedback about work-related issues. However, one interviewee suggested that one-on-one meetings are not optimal for providing honest feedback, as it is difficult to give negative feedback. For example, the interviewee asked "can someone criticise my leadership?"

Each of the teams had also implemented a retrospective meeting for gathering feedback. The retrospectives were held every two weeks in all the teams. The retrospective practices were not identical across teams, but each retrospective produced new action items from issues, and reviewed action items from the previous retrospective. All the interviewees felt that these retrospective meeting were producing valuable feedback for the whole team.

Escalating issues that were beyond the team's scope was not seen as problematic by the interviewees. According to them, the product leadership meeting was sufficient for escalating any larger issues. Some interviewees felt that they could easily escalate issues to their superior. All interviewees agreed that problems are easy to communicate, but most of them also stated that the problems are not solved as easily.

However, according to the interviewees, there was no mechanism in place for cross-team feedback. This was seen as one of the causes behind lack of trust between teams. Most of the interviewees felt that without honest feedback across teams, it was not possible to build trust.

6.1.5 Technology

Some interviewees focused on describing pressing technology challenges at Rally Software. According to those interviewees, the key technological challenge for Rally Software was overcoming a "crossroads situation" with the ALM product. These interviewees explained that the product is large, aged and has a significant amount of complexity. One interviewee said that engineering is faced with the challenge of "breaking from the legacy monolithic application architecture". The interviewee also suggested that engineers should understand why certain design patterns, like Service-oriented Architecture (SOA), are beneficial and not just blindly following them.

Given these unfavourable attributes, the teams were described to be struggling with moving to the next generation of the product. Most of the interviewees agreed that delivering customer value while still improving the technology foundations of the product was challenging (to say the least). For example,

one interviewee mentioned that it is difficult to justify investments when the results may only be visible after several quarters. The interviewee also stated that the business organization feels it is investing enough, but the software developers still don't have enough time for improving the foundations of the product.

6.2 Results of the second interview round

The second interview round was performed after the transformation was completed and the organization had stabilized. We analyzed the answers of the interviewees and grouped them under common, larger themes (as in the first round of interviews). These themes are described in detail in the following subsections.

6.2.1 Agile transformation

In general, the interviewees had assumed more responsibility in the organization. Two of the interviewees were now Product Owners for two teams and the others had moved on to other upper management roles.

All the interviewees felt that the teams were now doing significantly better than during the first round of interviews. The velocity of the teams had increased during this period, and the rest of the organization had more confidence towards the engineering department.

Most interviewees identified the back to basics transformation in the team level as a success factor. The additional focus on implementing and using agile practices – like Scrum – in a disciplined way had helped teams increase their velocity and consistency in delivering features. Nevertheless, one interviewee explained that the level of engagement in teams had decreased significantly as a result of introducing more discipline and taking away autonomy. Especially the teams and individuals who viewed the transformation as command and control struggled.

Most interviewees expressed concerns over the quality of the delivered software. The number of defects had been accumulating since the first interview round. Additionally, the increased feature delivery rate tended to introduce more new defects than could be fixed during the same time. These interviewees identified lack of consistent team quality practices as one factor behind the quality issue. One interviewee stated that ”we are struggling with defects and quality in general”.

All interviewees mentioned that the greatest change in team-level agile practices was requiring the use of Scrum practices. The teams must now report to the Program Level as Scrum teams using two-week sprints. However, one interviewee explained that one team is still using Kanban with similar output as the Scrum teams.

The interviewees pointed out several benefits from the increased use of Scrum practices. One interviewee explained that there was now less uncertainty of progress after switching from Kanban to Scrum. In addition, one team

had managed to drastically increase their feature acceptance rate after identifying the problem through team metrics and taking on a more disciplined approach to finishing features. Another interviewee mentioned that teams were now using standardized estimation, and were producing more consistent and accurate estimates.

6.2.2 The Scaled Agile Framework

Based on the interviews, the most significant change in Scaled Agile Framework practices was the stabilization of the PSI planning process. According to the interviewees, the teams are better prepared for the main event, which has yielded a more efficient and smoother experience when everyone attends the main planning event. For example, teams were already discussing trade-offs and inter-team dependencies before the planning event, according to one interviewee.

The PSI planning process is now continuously improved through small changes, whereas it was heavily modified each time at the time of the first interviews. All the interviewees agreed that the stability in the process was a success factor in the better planning results. Some interviewees still described the planning event as painful, but also felt that the small improvements were continuously steering the process in the right direction.

One interviewee explained that due to increased cross-functionality, teams able to work on more areas. The interviewee stated that ”in PSI planning, more teams are able to pick up features to work on”. Moreover, it enabled delivery of the highest value features.

One interviewee explained some suboptimal aspects of using the Scaled Agile Framework in the organization. First, the Scaled Agile Framework prescribes practices like Scrum, as all the ceremonies are build around them. However, it’s fair to ask if Scrum is the best process model for teams.

Second, the three-month release cadence of the Scaled Agile Framework has some waterfall flavor to it. According to the interviewee, this causes the tendency to plan features that fit into a release and don’t span release boundaries. In addition, if a feature is finished before the end of a release cycle, some filler work will be done instead of starting a new feature.

Third, according to the interviewee, the time boxes can cause bad decisions. For example, the time boundary can be used as an excuse for refusing to do something that is time-consuming, or introducing new technology, even if it would be the better approach. The interviewee explained that such time boxes can be particularly harmful for long-term decision making.

Another interviewee stated that the organization was using the Scaled Agile Framework better than during the first interviews. However, the interviewee

said that the case organization was ”still not doing SAFe as well as other companies”.

Most interviewees agreed that the Scaled Agile Framework is a good starting point for practicing scaled agile. They felt that the framework can be customized as the organization gains more experience in using it. One interviewee suggested that the Scaled Agile Framework provides a clear and efficient way to steer Program Level work. The people in the organization understand what they are going to and hope to do in the next quarter. Additionally, there is a clear way to estimate progress with the framework.

A couple of interviewees pointed out that the framework is very useful for companies that have little experience with agile. In addition, the Scaled Agile Framework provides clear roles for everyone in the agile organization. However, one of these interviewees explained that incorporating user experience designers and some other roles was slightly problematic.

6.2.3 Change resistance

Some interviewees explained that there was some resistance towards estimation (which was introduced as a part of the back to basics transformation). Another area that had triggered some resistance was the adoption of new technologies.

Several interviewees mentioned that introducing new technologies and frameworks was creating pressure for teams. One of these interviewees explained that the whole build, deploy and test process had changed over the last 8 months, since the previous interview. The interviewee stated that ”we are going through a change in technology and how it should be tested”. On the other hand, the interviewees identified this as an achievement as well. One interviewee suggested that the rate of innovation should slow down to give the teams time to properly adjust.

6.2.4 Achievements

Several interviewees described the recent organizational transformation as an achievement. One key indicator of the success was restored trust in the engineering organization to deliver value. One interviewee explained that ”we have gone from a place where engineering had lost its way to where it has delivered, in a meaningful way, for Rally and customers”.

Many of the interviewees explained that removing product lines from the engineering organization was a success factor. Without product lines, teams were really working on the same product and forced to look at dependencies.

One interviewee had addressed the growing quality issue by bringing up several improvements to the flow of defects. In addition, the interviewee had changed the defect assignment process from push to pull and introduced quarterly team quotas for pulling defects.

Focusing on a couple of key initiatives was seen as a success factor by some interviewees. Previously, there were lots of initiatives, but the amount was reduced to a maximum of three. One of those interviewees explained that initiatives have helped to solidify what is most important for the company and what to focus on.

6.2.5 Improvement suggestions

One interviewee explained that improvement suggestions will be implemented as long as the teams really believe in them. According to the interviewee, there is a strong culture at the organization against telling people what to do. The interviewee explained that ”running around and telling people what to do doesn’t work [at Rally]”. For example, the forced Scrum process was not received well unless the reasoning was clearly explained. When teams see how it connects to the Program Level work, people tend to adopt it better.

One interviewee explained that the organization should focus on introducing less defects and fixing escaped defects efficiently. Moreover, the interviewee suggested that the culture must change as well. The engineering teams needs to write test cases on the team and Program level.

Another interviewee suggested that the organization should return autonomy to teams. As the level of engagement has been declining since the transformation and removal of autonomy, restoring autonomy should have a positive impact on engagement on team and individual levels. Since the teams now understand what the organization’s expectations are, they can figure out alternative ways to fulfill these expectations. For example, addressing the quality issue could be delegated to the team level.

One interviewee addressed the definition of features in the Scaled Agile Framework planning. According to the interviewee, features are not really boxes with start and end dates (as they are simplified in the planning process). There is hidden work after the box is done, after eliciting customer feedback, for example. The interviewee explained that there are four stages for features: exploration, building, market test and adjustment. The engineering organization should more specifically communicate the stages of the features to the business organization.

Another interviewee addressed the same issue from the viewpoint of large initiatives. According to the interviewee, two initiatives were being finished in

the current quarter, but user interface consistency work was still ongoing and thus they were not really finished yet.

According to one interviewee, the work as a product organization has only begun. The people in the leadership team are still figuring out their roles and are somewhat focused on their own developer teams. Ideally, the product organization should work as one large team striving for a common goal.

6.2.6 Trust

All the interviewees agreed that trust within teams is still at a good level. One interviewee explained that this trust is demonstrated by the fact that people rarely want to change teams.

According to some interviewees, on the inter-team level, trust has improved. One of them mentioned that after the product lines were removed, the teams are part of the same product, which has increased trust. However, according to another interviewee, a lack of communication on how reusability and integration of components is handled has sometimes created distrust between teams. Some interviewees mentioned the historical ability to deliver and geographical distribution were still reasons for distrust between teams.

The interviewees had mixed views on trust in management. Several interviewees explained that their trust in management had increased since the last interview. On the other hand, a couple of interviewees described some trust issues. One interviewee explained that teams have less trust in management due to not being asked their opinion in the transformation process. Another interviewee suggested that the recent layoffs and some people leaving the company had created tension and uncertainty, which had a negative impact on trust in management. The interviewee said that ”it’s been going on too long and people are on their toes”.

Most interviewees were satisfied with how easily problems are communicated in the case organization. However, a couple of interviewees explained that the organization talks more about problems than successes. Only discussing issues is not very motivating. According to one interviewee, constructive criticism could be used more in communicating problems.

One interviewee explained that the product steering meeting does not feel like a safe room to tell others about problems. On the other hand, another interviewee said that everyone has to share in the product steering meetings, which makes it easier to tell about their own issues when the others talk about their issues too.

6.2.7 Agile at the organizational level

One interviewee suggested that some parts of the organization are still struggling with working in an agile way. The interviewee explained that integrating functions like sales, legal, marketing and accounting into the scaled agile process is difficult. For example, when an initiative is presented by engineering, marketing has to figure out how to market it and sales has to understand the value in order to be able to sell it. According to the interviewee, there have been both successes and failures in this process. The interviewee explained that one way to evaluate this is by asking ”is our the sales team enabled in such a way that they understand what the software is and can leverage it in the deals that they are making?”.

A couple of interviewees suggested that constant care is needed to keep the organization working in an agile way. Keeping the ceremonies and focusing on constantly improving their ways of working were important factors. Once things start going well and the organization obtains the initial gains, there is a tendency to stop investing in the agile process. The organization must therefore stay alert. One of these interviewees pointed out that the organization is no longer as willing to change as during the initial interview. Instead, now the organization has a more stabilized planning process and opts for small improvements instead of throwing everything out.

One interviewee explained that there is a tendency to focus too much on one’s own domain in the organization. According to the interviewee, people need to understand how the whole system works and optimize towards it rather than optimizing for themselves.

6.2.8 Summary

This subsection presents a summary of changes in the organization that were described in the second round of interviews. The changes are summarized in following Figure 6.2.

Viewpoint	Effects of transformation
Agile practices of teams	• Increased discipline in following agile practices • Somewhat decreased engagement of teams and individuals
Agile at organization level	• Stabilized PSI planning process • Increased feature delivery rate • Increased defect rate
Trust	• Increased level of trust between teams • Increased trust in the engineering organization • Improvement in trust in management

Figure 6.2: Summary of results from the second round of interviews

Chapter 7

Discussion

In this chapter, we compare the results of the literature review and the interviews performed at Rally Software. We will also present causes and effects drawn from the interview results and suggest next steps for the case organization. Finally, we discuss the threats to the validity of this work and estimate the potential limitations.

The causes and effects are summarized in the following figure 7.1.

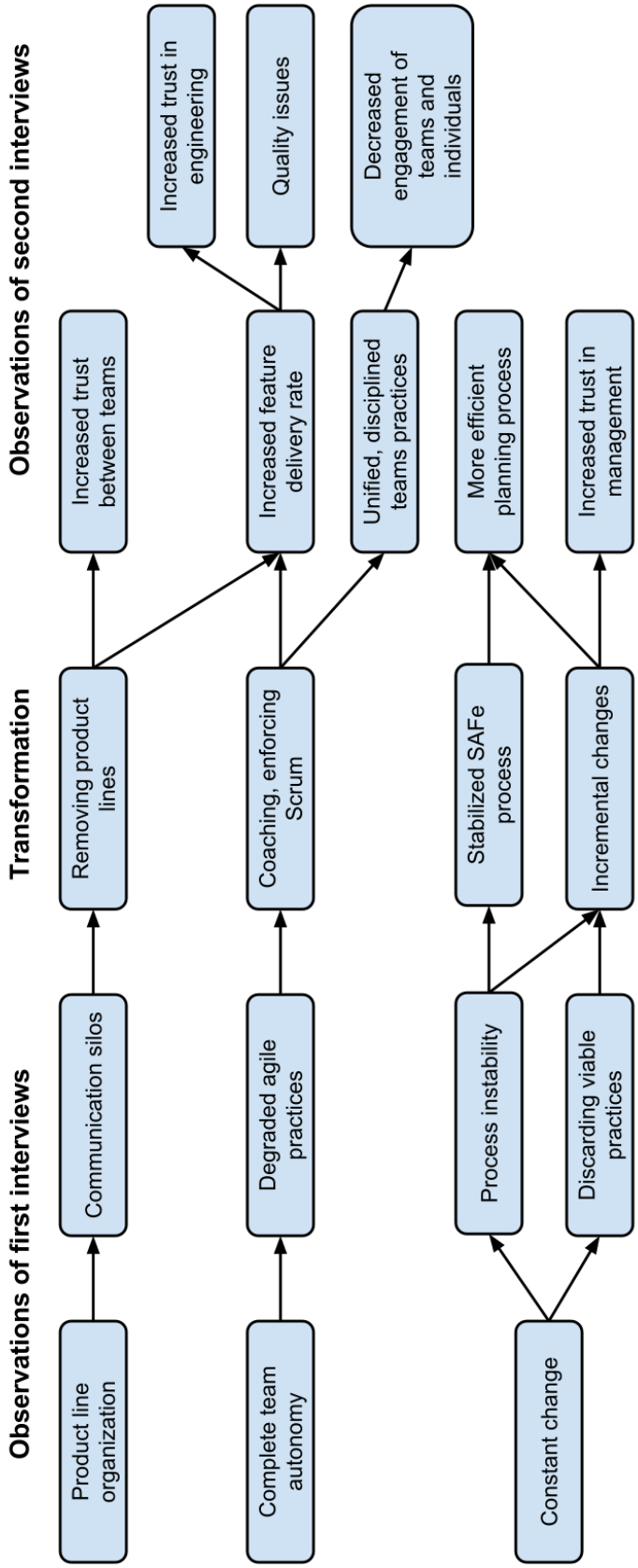

Figure 7.1: Summary of causes and effects found in this case study.

7.1 Initial state of the organization

This section presents a comparison of results from the literature study and the observed status of the case organization based on the first round of interviews. The comparison is divided into subsections based on themes.

7.1.1 Lack of trust between teams

The interview results suggest that there is a lack of trust between teams in the case organization. Most interviewees agreed that there is at least some distrust between teams and it has a negative effect on the scaled agile process. The interviewees also explained that the communication and collaboration between teams was at a suboptimal level.

The interviewees also explained that there was a lack of inter-team feedback in the case organization. The feedback mechanisms in the organization were mostly focused on individuals getting feedback from their team leaders and vice versa. Most of the interviewees felt that without honest feedback across teams, it was not possible to build trust between them.

This was not, however, prominent in the cases described in the literature. In fact, only one paper in the literature study described team-centricity and distrust between teams (Hansen and Baggesen, 2009). However, another paper regarded trusting others to make good decisions as a trait of a high-performing team (Moore and Spens, 2008). The small number of papers addressing this aspect could be explained by the focus on practices and processes in the case studies. Still, trust is an integral part of organizational culture and thus has an effect on software development processes.

7.1.2 Erosion of agile practices

The interviewees expressed concerns over teams forgetting the value of certain agile practices and even discarding them altogether. This has led to deteriorating agile practices in the engineering organization as the value behind a practice is not understood. One interviewee even described the end result as the team being unsuccessful with agile and having overconfidence in their own excellence.

The erosion of agile practices was also evident in the literature study results. Many of the papers explained how the divergence of agile models between teams had become a challenge (Benefield, 2008; Chung and Drummond, 2009; Goos and Melisse, 2008; Vlaanderen et al., 2012; Schnitter and Mackert, 2011; Vlaanderen et al., 2012; Paasivaara et al., 2014b,b). Some papers also described how the lack of understanding resulted in a tendency to strip away

some practices completely (Chung and Drummond, 2009; Lewis and Neher, 2007). These issues correlate very well with the findings of the interviews.

The case organization began providing agile coaching to the teams during the interviews. According to the interviewees, this was beneficial for the teams. The coaching helped in re-enforcing the core agile values and fixing the deteriorated practices. The literature study results seem to correlate with this assumption. In the analyzed papers, coaching was the most frequently mentioned success factor for establishing and maintaining a scaled agile process (Atlas, 2009; Benefield, 2008; Brown, 2011; Chung and Drummond, 2009; Cowan, 2011; Gat, 2006; Goos and Melisse, 2008; Hanly et al., 2006; Long and Starr, 2008; Mencke, 2008; Rodríguez et al., 2013; Schatz and Abdelshafi, 2005; Seffernick, 2007; Silva and Doss, 2007; Smits and Rilliet, 2011).

7.1.3 Training

Many papers attribute some part of the success to providing training about agile methods (Cowan, 2011; Fecarotta, 2008; Gat, 2006; Hanly et al., 2006; Long and Starr, 2008; Mencke, 2008; O'Connor, 2010; Seffernick, 2007). However, all of these case studies focused on organizations transforming from waterfall or other models into scaled agile. Arguably, these results may not be directly applicable for organizations that have been maintaining scaled agile for an extended period of time, such as the case organization of this thesis.

The case organization provides all the new employees with agile training during their first few days. Existing employees had not received much additional agile training. However, the case organization recently decided to organize a training for the engineering department to create a common baseline on practices across the research and development organization.

7.1.4 Coaching

All the interviewees agreed that introducing external coaches had a positive effect on teams. According to them, the coaching helped teams focus on a disciplined implementation of practices. This is strongly supported by the literature study results, which indicate that coaching was one of the most common success factors.

According to some papers, one of the benefits of coaching was that the coaches were able to react immediately to any anti-patterns in the team's behavior (Chung and Drummond, 2009; Rodríguez et al., 2013). In some cases, external coaches were able to provide valuable objective feedback (Schatz and Abdelshafi, 2005; Silva and Doss, 2007).

The main benefit of coaching seems to be the ability to guide the scaled agile implementation towards optimal performance. While training can help in establishing agile principles, the ongoing implementation process is where the actual learning happens. Having experts guide the teams and the organization in this process seems to be very beneficial.

7.1.5 Customization of the agile model

The interview results suggest that constant change in processes and business priorities is a serious challenge for the organization. According to the interviewees, not having a clear status quo made it difficult to evaluate the effects of changes and whether they were beneficial or justified. Embracing change is one of the core agile values, but too much change can be a burden on the whole organization.

The literature study results include only one similar case where constant change was seen as a challenge. Paasivaara et al present a case study where the organization seemed to have very similar issues with constant change (Paasivaara et al., 2014a). However, only a few of the papers describe organizations that have used agile methods for an extended period of time. This issue could very well be specific to organizations with an established scaled agile process.

Nevertheless, some papers did recommend introducing small, gradual improvements to the process instead of a complete overhaul. Letting the process evolve through clearly motivated and justified customizations was seen as a success factor in these cases. (Federoff and Courage, 2009; Long and Starr, 2008; Maples, 2009)

7.1.6 Inter-team communication and collaboration

All interviewees felt that the product line based organization structure was causing communication silos. In addition, interviewees mentioned aversion towards having dependencies to other teams' work, and skepticism about other teams' ability to deliver. These factors suggest that inter-team communication is suboptimal at the case organization.

The lack of inter-team communication and collaboration was regarded as a major problem in several papers (Beavers, 2007; Berczuk and Lv, 2010; Chung and Drummond, 2009; Cloke, 2007; Farrow and Greene, 2008; Gat, 2006; Hansen and Baggesen, 2009; Laanti, 2008; Lee, 2008; Lewis and Neher, 2007; Moore and Spens, 2008; Ranganath, 2011; Spayd, 2003; Moe et al., 2014; Paasivaara and Lassenius, 2014; Sekitoleko et al., 2014). The lack of collaboration resulted in an array of problems, such as managing dependencies and the integration of components (Chung and Drummond, 2009; Cloke, 2007; Gat, 2006; Laanti,

2008; Lewis and Neher, 2007; Moore and Spens, 2008), unclear responsibilities between teams (Beavers, 2007), duplicate work (Lee, 2008), the inability to see the bigger picture (Berczuk and Lv, 2010; Farrow and Greene, 2008; Moore and Spens, 2008), a low level of collaboration (Lewis and Neher, 2007; Moore and Spens, 2008; Ranganath, 2011; Rodríguez et al., 2013; Spayd, 2003; Moe et al., 2014) and distrust between teams (Hansen and Baggesen, 2009). The team centricity described in one paper bears a strong resemblance to the case organization's situation (Hansen and Baggesen, 2009).

It seems that the lack of communication and collaboration is related to trust issues between teams in the case organization. However, it is not clear which is the cause and which is the effect. These factors seem to create a vicious cycle where growing distrust hinders collaboration and communication, and vice versa.

7.1.7 Experiences with the Scaled Agile Framework

The outline of the first interview round did not include any specific questions about the Scaled Agile Framework, but some interviewees mentioned it when talking about agile at the organizational level. Some interviewees explained that the Scaled Agile Framework was followed at a philosophical level, but not in a disciplined manner. One interviewee wanted more visibility to the Program level of the framework. All in all, the framework did not seem to have a significant effect on the teams or their internal processes and practices.

The literature study did not find any cases that present experiences with a fully implemented Scaled Agile Framework. Some papers describe similar practices that were used in the studied organizations (Díaz et al., 2014; Heikkilä et al., 2013, 2015). All in all, there seems to be a lack of literature about the Scaled Agile Framework and its applicability to scaling agile software development. According to the Scaled Agile Framework website (Leffingwell LLC, a), the process model has been introduced in large enterprises like John Deere. However, the case studies presented by the website haven't been published in any academic publications. Thus it is difficult to estimate how well the case organization's implementation of the Scaled Agile Framework has performed and how the framework suits scaled agile development in general.

7.1.8 Explanation building

The lack of trust between teams was mentioned by most interviewees. It seems to be a central symptom of underlying issues in the case organization. Several interviewees explained that the product line organization structure cre-

ated communication silos. This is a likely cause for the lack of trust between teams. In addition, most interviewees described that there were no cross-team feedback mechanisms in the organization. Those interviewees felt that honest feedback was essential for building trust between the teams.

In the past, the case organization has let teams decide their practices themselves. Some interviewees described that the agile practices of teams have deteriorated over time and teams have forgotten the value of the practices. There could be a cause and effect relationship between these two issues.

As all interviews identified constant change as a serious challenge, it is important to understand what is triggering changes in the organization. One interviewee suggested that the organization's interest in continuous improvement was a driver behind the constant changes. Some interviewees mentioned that the changes in management personnel and drifting priorities were decreasing trust in management.

Some interviewees described a disconnect between the business and the product organizations. The interviewees did not provide an explanations for the disconnect. However, one interviewee suggested that the Scaled Agile Framework was not followed in a disciplined way. As the framework intends to connect the stakeholders of the whole organization, not following it fully may explain the feeling of disconnect.

One interviewee focused on describing technological issues. According to them, software development is hindered by the monolithic legacy architecture and the complexity that the product accrued over the years. The interviewee went on to describe their difficult situation of balancing technology improvements and delivering customer value at the same time. In addition, the business and product organizations have different opinions on whether enough investment is put into improving the foundations of the product. This could be a sign of the disconnect between the business and the product organizations that was described above.

7.1.9 Summary

The observed initial state of the organization exhibits many similarities to the organizations of the case studies found in the literature study. First, inter-team communication and collaboration seems to be a common challenge among organizations maintaining scaled agile. Second, erosion of agile practices was visible in Rally Software as well as many of the case study organizations. Third, coaching proved to be an important success factor for case study organizations as well as the teams at Rally Software. Coaching helped by mitigating the negative effects of the erosion and re-enforcing teams' agile practices.

Most of the case studies were focused on organizations that were adopting

scaled agile or had recently introduced it. Rally Software, on the other hand, has been practicing scaled agile for several years. This difference may explain some of the disparities in the challenges found in the literature study and in the interviews. For example, the fast-paced process evolution and inter-team trust issues may be easily observable in organizations that have used scaled agile for an extended period of time.

Rally Software was also the only organization that had chosen to adopt the Scaled Agile Framework in full. Other organizations in the case studies mostly used custom process models utilizing Scrum or Kanban on the team level and auxiliary practices like Scrum of Scrums to connect the teams. However, some organizations did use practices that were inspired by the Scaled Agile Framework.

7.2 The state of the organization after the transformation

This section presents the observed status of the case organization after the transformation and the second round of interviews. The comparison is divided into the following subsections based on different themes.

7.2.1 Agile practices at the team level

The interviewees agreed that the agile practices of the teams were significantly improved and unified as a result of the transformation. The teams were delivering more features than before the transformation. In addition, the level of trust between teams had improved as they were now working together on one product, according to some interviewees.

However, one interviewee expressed concerns over teams and individuals losing their engagement as a result of the forced transformation. Since the teams had to conform to the expectations of management, the level of autonomy within teams was reduced drastically. The interviewee suggested restoring autonomy to teams now that management expectations were clearly defined, letting the teams figure out ways to meet those expectations on their own. Given that the transformation was more or less a top-down order, enabling teams and individuals to innovate and continuously improve on their own seems like a logical approach to bring back agile values on the team level.

7.2.2 Agile at the organization level

Previously, most interviewees identified constant changes as a significant challenge in the organization. However, during the second round of interviews, the interviewees explained that the planning process had stabilized and did not express any concerns about excessive change in the organization. According to several interviewees, the changes are now smaller and focus on continuously improving by addressing specific issues.

The previous product line organization received a fair share of criticism in the first round of interviews. In the second interviews, the interviewees regarded teams working together on one product as a success factor behind the increased velocity. It seems that the product line organization was not an optimal structure for the case organization.

A couple of interviewees expressed concern about the usage of the Scaled Agile Framework in the organization. One interviewee explained that the case organization was not using the framework as efficiently as other companies. Another interviewee described the drawbacks of the framework, such as release time boxes that cause suboptimal behavior and the framework's focus on a single process model (Scrum). None of the literature review papers addressed these aspects of the Scaled Agile Framework.

7.2.3 Current challenges

The interviewees identified several challenges in the second round of interviews.

First, the most frequently mentioned issue was maintaining quality at an acceptable level. The increased feature delivery rate had also increased the rate of new defects. A couple of interviewees mentioned that the teams did not have common quality practices. One of the interviewees suggested that the quality practices of the teams should be unified in order to address the quality issue. Incorporating quality assurance was also found to be problematic in several of the literature review case studies (Beavers, 2007; Gat, 2006; Maples, 2009; Schnitter and Mackert, 2011; Smits and Rilliet, 2011).

Second, according to one interviewee, the teams and individuals were less engaged than before as a result of the top-down transformation. The interviewee explained that the transformation had removed teams' autonomy to decide their own ways of working. The transformation also meant a strong change in organization culture and decision-making, which had been previously been even excessively collaborative. As a solution, the interviewee suggested restoring autonomy to teams now that the management's expectations are well-defined and the teams know how to meet them. The literature

study papers did not describe similar reactions to increased discipline in agile practices. However, many papers suggested that a top-down approach creates change resistance (Abdelnour-Nocera and Sharp, 2007, 2008; Evans, 2008; O'Connor, 2010; Spayd, 2003). Therefore, avoiding further top-down changes could have positive long-term effects on the engagement of teams.

Third, a couple of interviewees suggested that constant care is needed to keep the organization working in an agile way. According to these interviewees, there is a tendency to stop investing in the agile process after the initial gains. As a solution, one interviewee suggested keeping the ceremonies and focusing on improving all the time. Several papers also emphasizes the importance of pragmatical customization of the agile process (Federoff and Courage, 2009; Long and Starr, 2008; Maples, 2009).

Fourth, one interviewee described challenges in integrating other functions of the organization into the agile process. The interviewee explained that integrating functions like sales, legal, marketing and accounting into the scaled agile process is still difficult. This challenge is very well described in literature. The literature review found case studies that described issues in integrating marketing (Abdelnour-Nocera and Sharp, 2007, 2008; Beavers, 2007; Benefield, 2008; Maples, 2009; O'Connor, 2011; Rodríguez et al., 2013; Schnitter and Mackert, 2011; Smits and Rilliet, 2011), sales (Maples, 2009; Smits and Rilliet, 2011), legal (Maples, 2009) and finance (Maples, 2009) into the agile process. The Scaled Agile Framework does not specifically prescribe how these functions should be integrated into the process. The framework provides input points for stakeholders and visibility into the engineering process, but it is largely the responsibility of the other functions to self-organize around the framework. The interviewee explained that when an initiative is presented by engineering, marketing has to figure out how to market it and sales has to understand the value of it in order to be able to sell it. According to the interviewee, this approach has produced mixed results.

7.2.4 Summary

All of the interviewees felt that the transformation in the organization was a success. According to the interviewees, the organization achieved the goals set for the transformation: unifying teams under one product and vision, and thus increasing the feature delivery rate to match and outperform competitors.

The set of identified challenges is completely different than in the first round of interviews. As a result of the transformation, the case organization has evolved and needs to address these new issues. The literature provides little insight into the new challenges. Case studies about issues in using the Scaled Agile Framework are especially scarce. In the next section, we present

potential next steps for the case organization.

7.3 Next steps for the case organization

In this section, we present potential next steps for the post-transformation case organization. The interviewees provided several improvement suggestions that are worth considering.

First, the engagement of teams and individuals seemed to have decreased as a result of the top-down transformation process. To address this, one interviewee suggested giving autonomy back to the teams. This would let the teams decide their own ways of working – as long as they can provide the current output to the Program level of the Scaled Agile Framework.

As the teams are now familiar with the expectations from management, having the same process model across all the teams may not be required to keep up the increased delivery rate. Some papers suggest that pragmatical customization of the process is a success factor (Federoff and Courage, 2009; Long and Starr, 2008; Maples, 2009). However, in the initial state before the transformation, the teams had very diverged practices and were not following them in a disciplined way. Given this, it might be challenging to find a balance for the right amount of team autonomy so that the organization does not drift back to the suboptimal state.

In the second round of interviews, quality-related issues were mentioned by all interviewees. Thus, one important aspect of the team-level agile process model is incorporating quality assurance. The interviewees had different opinions about how the quality issue should be addressed. One interviewee proposed introducing unified quality practices for teams, such as writing test cases. We found some papers in the literature study that regarded synchronized practices as a success factor (Brown, 2011; Evans, 2008; Farrow and Greene, 2008; Ryan and Scudiere, 2008). However, another interviewee suggested letting teams address the quality issue on their own. According to the interviewee, this could increase the engagement of the teams and individuals. As long as the process customization is pragmatic, there is some support for modifying practices in the literature (Federoff and Courage, 2009; Long and Starr, 2008; Maples, 2009). However, several papers list divergence of agile practices as a challenge (Chung and Drummond, 2009; Goos and Melisse, 2008; Lewis and Neher, 2007; O'Connor, 2011; Ryan and Scudiere, 2008; Schnitter and Mackert, 2011; Vlaanderen et al., 2012; Paasivaara et al., 2014a).

All in all, the issue boils down to finding the optimal amount of freedom and discipline in the team-level process model. The case organization has experimented with the opposing ends of the spectrum, moving from complete

autonomy to a top-down mandated process model. The current disciplined approach seems to yield a faster feature delivery rate and boost confidence in the engineering department. Thus, trying to maintain these benefits while addressing drawbacks – such as decreased engagement – seems like a sensible approach. This could be achieved by the suggested increase in team level-autonomy. However, at the same time, the teams should not lose their current disciplined approach to following practices or change practices without clear justifications and careful evaluation.

According to the interview data, other functions of the organization were not fully integrated into the scaled agile process. The software engineering organization was aligned with the Scaled Agile Framework, but other functions like sales, legal, marketing and accounting were at least somewhat disconnected from the framework. Some papers found in the literature review suggest approaches for integrating marketing into requirements management in large scale agile processes (Beavers, 2007; Gat, 2006; Maples, 2009). However, the Scaled Agile Framework already provides input points for marketing and other functions. Thus, the issue could be addressed by encouraging the other functions to fully embrace the framework and its ways of working. After all, the transformation process succeeded in aligning the engineering department with the framework and producing positive results in terms of productivity and trust.

Now that the transformation is complete, it is important to have a retrospective on what could be done better if the need for changes arise in the future. According to the interviewees, the top-down approach for introducing changes triggered change resistance. This effect has been described in several other case studies as well (Abdelnour-Nocera and Sharp, 2007, 2008; Evans, 2008; O'Connor, 2010; Spayd, 2003). One interviewee reflected on past improvement suggestions and explained that having a buy-in from teams and individuals was essential in reducing friction caused by changes. According to the interviewee, when a trusted person explained the underlying motives for the changes, there was very little resistance from the teams. Similarly, several papers highlight the importance of investing in communicating changes appropriately (Mencke, 2008; Prokhorenko, 2012; Ranganath, 2011; Ryan and Scudiere, 2008; Seffernick, 2007; Smits and Rilliet, 2011).

7.4 Threats to validity

The following subsections will address the validity of this thesis as a case study. The validity of this case study is assessed with four tests: construct validity, internal validity, external validity and reliability. These tests are described by

Yin in his book Case Study Research: Design and Methods (Yin, 2013).

7.4.1 Construct validity

Construct validity refers to the extent to which a case study investigates what it claims to investigate. In other words, when establishing the construct validity of a procedure, one needs to evaluate to which extent it leads to accurate observations of reality. (Denzin and Lincoln, 2011)

The main concern for construct validity was the accuracy of the descriptions of the interviewees. To increase construct validity, we interviewed multiple people with the same Product Owner role in the organization. The interviewees' teams represent around half of the engineering department at Rally Software. However, the number of interviewees is fairly low, and thus can cause bias towards certain issues. Moreover, all the interviewees have a management role, which may affect their perspective. For example, the teams' software developers might have different answers for the interview questions.

Case study research literature suggest using multiple sources of evidence and maintaining a chain of evidence (Yin, 2013). In this case study, we used interviews as the only source of evidence. We maintained a chain of evidence by recording all the interviews and producing text summaries of the answers. Additionally, the key informants were given a draft of the case study to review.

7.4.2 Internal validity

Internal validity is mainly the concern of explanatory case studies where the researcher tries to explain how and why one event lead to another (Yin, 2013). As this case study is not explanatory, this aspect of internal validity is not relevant. However, another aspect of internal validity is the inferences a researcher makes every time an event cannot be observed directly. In this case study, we rely on the descriptions of the interviewees and infer that some events resulted from earlier occurrences that were described by the interviewees. To increase internal validity, we have performed pattern matching and used explanation building.

7.4.3 External validity

External validity means establishing the domain for generalization of the results of the case study (Yin, 2013). As this case study focuses on a single organization, the domain for generalization is mainly limited by the characteristics of the case organization.

Rally Software is a fairly large organization with around 500 employees. However, roughly half of the employees are working outside of the engineering organization, which is responsible for software development. Thus, while these findings may apply for an organization with a couple of hundred engineers, an organization of a thousand or more engineers is likely to yield different results. Additionally, the case organization has a long history of both employing agile practices and developing agile software tools. This puts Rally Software on a completely different level when compared to organizations that are transitioning to or have just adopted agile methods.

7.4.4 Reliability

Yin defines reliability as repeatability of operations of the case study. We applied the sample case study protocol suggested by Yin in this thesis. (Yin, 2013)

All the procedures followed in this case study are documented in Chapter 5. We stored all the case study data including recordings of the interviews and the text summaries of the answers for later reference and repeatability.

Chapter 8

Summary and conclusions

This chapter presents the conclusions of this thesis and suggests possible future work based on the findings. The conclusions are divided into the following four sections. These sections present answers to the research questions.

8.1 Agile at Rally Software

Rally Software uses the Scaled Agile Framework as its process model for scaled agile at the organizational level. Individual software development teams use Scrum or Kanban as their process model. In addition, the case organization uses the Scrum of Scrums practice to connect the teams working on the Agile Lifecycle Management (ALM) product. The product owners of these teams are also connected via product leadership meetings.

During the case study, Rally Software focused on restoring disciplined agile practices across the software development teams. Consequently, every team working on the product has been assigned an external Scrum Master. The Scrum Master role is similar to the one defined in the Scrum process model, but in this case is especially focused on overseeing that practices are not too far removed from textbook definitions. The interviewees identified the focus on disciplined agile practices as a success factor in the increase feature delivery rate.

The case organization arranges three regular planning meetings that involve the whole organization. First, there is annual planning meeting that sets the course for a year. Second, quarterly steering meetings define what features are going to be implemented during the quarter. Third, the teams prepare for PSI planning meetings by breaking features into user stories and identifying dependencies to the work of other teams. In the actual PSI planning meeting, the individual planning efforts of teams are combined into a common release

plan.

In addition, Rally Software uses their own ALM tool for managing the work of teams distributed across several locations.

8.2 Common issues of scaling agile

The literature study of this thesis identified several common issues of scaling agile methods. These issues are listed below in descending order according to the number of case studies that mention the issue.

1. Inter-team communication, team coordination, dependency management
2. Requirements management with large/complex products
3. Integrating other functions of the company
4. Not understanding the agile values behind practices
5. Diverged agile models between teams
6. Lack of literature guidance on scaling agile practices
7. Incorporating quality assurance and performance testing into the scaled agile process
8. Teams are too focused on own goals instead of the bigger picture
9. Top-down approach creates resistance towards change
10. Including user-centered design processes into scaled agile

8.3 Issues in the case organization

8.3.1 First round of interviews

The first round of interviews that were conducted at Rally Software revealed various issues in the software development processes of the organization. The findings are grouped into the following four major themes.

First, the interviewees suggested that teams' agile practices had eroded over time. The teams had diverged from textbook versions of practices, and had even discarded practices completely. While process evolution is part of agile, the interviewees felt that the results had not been optimal.

Second, agile at the organizational level received a fair share of criticism from the interviewees. The most frequently mentioned challenge was the high level of change in processes and the business priorities of the organization. In addition, the product line organization structure was creating communication silos. Some interviewees also described the quarterly steering and planning process as too time-consuming.

Third, according to the interviewees, there was a lack of trust between the teams. The level of cross-team feedback and collaboration was not sufficient for an organization that aims to excel at scaled agile software development. Some interviewees explained that their trust in management was slightly reduced due to the recent personnel changes.

Fourth, interviewees with extensive experience in developing the product said that the technology foundations of the ALM product were in need of refactoring and improvement. According to the interviews, allocating resources for performance improvements and refactoring was problematic as there was also pressure to deliver more customer value in every iteration. Balancing between these two interests was difficult as long-term maintenance work would produce concrete customer value only after several quarters worth of development.

8.3.2 Second round of interviews

The second round of interviews were conducted after the case organization had undergone a transformation that aimed to address the previously identified issues. This round of interviews found four major challenges, which are described next.

First, as a result of the increased velocity of teams, the overall quality of the delivered software had decreased. The defect rate was on the rise and the teams lacked common practices for addressing software quality.

Second, the top-down transformation process had decreased engagement in teams and individuals. In addition, the transformation meant a strong change in organization culture and decision-making, which had previously been excessively collaborative.

Third, continuous effort is needed in maintaining the positive trend after the transformation. According to the interviewees, organizations tend to stop investing in improving the agile process after the initial gains.

Fourth, integrating other functions of the organization into the agile process was still challenging. Functions like sales, legal, marketing and accounting were not fully aligned with the rest of the agile organization.

8.4 Approaches to resolving the current issues

Incorporating quality assurance into the scaled agile process was seen as problematic in several other case studies as well. However, none of the case studies suggested approaches for resolving the issues. During the interviews, one interviewee suggested unifying the quality practices of teams. On the other hand, another interviewee suggested that giving autonomy back to the teams would allow teams to address the issue themselves and thus increase their engagement as well.

Several case studies suggested that top-down changes create resistance. Likewise, the transformation in the case organization decreased engagement of teams and individuals as team autonomy was taken away. Now that the goals of the transformation were achieved, the case organization can avoid making further top-down changes to reduce change resistance. If change is necessary, investing in good communication and trying to achieve buy-in from teams can help in reducing change resistance.

Pragmatic customization of the agile process was seen as a success factor in some case studies. Some interviewees also suggested keeping the ceremonies and continuously improving the agile process. These approaches could help in preventing the agile process from becoming stagnant or degrading.

The case studies from the literature study did not yield much insight into how other functions of the company should be incorporated into the agile process. The interviewees did not have suggestions for resolving this issue either. The Scaled Agile Framework provides clear input points for stakeholders outside the engineering organization. Thus, the issue could be addressed by encouraging those stakeholders to fully embrace the framework and its ways of working.

8.5 Future work

The literature study suggests that there are few studies on maintaining the scaled agile process. Most of the analyzed papers describe the initial agile transformation and issues related to it. However, as an agile process constantly evolves, it would be beneficial to follow the process evolution and evaluate how successful the transition is in the long term. Moreover, another interesting target for a case study is organizations that have maintained a large scaled agile process over a longer period of time.

Another potential research topic is the role of the Scaled Agile Framework in scaling agile. Only a few of the papers found in the literature review described experiences with the framework. The interviews in the case organi-

zation yielded some insights into the benefits and issues of the framework in large scale agile software development. However, further case studies of the framework are needed to produce data that is not tied to a single organization and its implementation of the framework.

Bibliography

J. Abdelnour-Nocera and H. Sharp. Adopting agile in a large organization: balancing the old with the new. Technical report, The Open University, Faculty of Mathematics and Computing, Department of Computing, 2007.

José Abdelnour-Nocera and Helen Sharp. Adopting agile in a large organisation. In Pekka Abrahamsson, Richard Baskerville, Kieran Conboy, Brian Fitzgerald, Lorraine Morgan, and Xiaofeng Wang, editors, *Agile Processes in Software Engineering and Extreme Programming*, volume 9 of *Lecture Notes in Business Information Processing*, pages 42–52. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-68254-7. doi: 10.1007/978-3-540-68255-4_5. URL http://dx.doi.org/10.1007/978-3-540-68255-4_5.

S.W. Ambler and M. Lines. *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*. IBM Press. Pearson Education, 2012. ISBN 9780132810104. URL http://books.google.fi/books?id=CwvBEKsCY2gC.

D.J. Anderson. *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010. ISBN 9780984521401.

A Atlas. Accidental adoption: The story of scrum at amazon.com. In *Agile Conference, 2009. AGILE '09.*, pages 135–140, Aug 2009. doi: 10.1109/AGILE.2009.10.

Tom J. Bang. Introducing agile methods into a project organisation. In *Proceedings of the 8th International Conference on Agile Processes in Software Engineering and Extreme Programming*, XP'07, pages 203–207, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-73100-9. URL http://dl.acm.org/citation.cfm?id=1768961.1769007.

P.A Beavers. Managing a large "agile" software engineering organization. In *Agile Conference, 2007. AGILE '07.*, pages 296–303, Aug 2007. doi: 10.1109/AGILE.2007.40.

Beck, K. and Beedle, M. and Bennekum, A. and Cockburn, A. et al. Agile manifesto. http://agilemanifesto.org/. Accessed: 2015-06-14.

M. Beedle. Enterprise Scrum. http://www.enterprisescrum.com/. Accessed: 2014-12-02.

G. Benefield. Rolling out agile in a large enterprise. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, pages 461–461, Jan 2008. doi: 10.1109/HICSS.2008.382.

S. Berczuk and Yi Lv. We're all in this together. *IEEE Software*, 27(6):pp. 12–15, 2010. ISSN ISSN 0740-7459.

B. Boehm and R. Turner. Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, 22(5):30–39, Sept 2005. ISSN 0740-7459. doi: 10.1109/MS.2005.129.

AlanW. Brown. A case study in agile-at-scale delivery. In Alberto Sillitti, Orit Hazzan, Emily Bache, and Xavier Albaladejo, editors, *Agile Processes in Software Engineering and Extreme Programming*, volume 77 of *Lecture Notes in Business Information Processing*, pages 266–281. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-20676-4. doi: 10.1007/978-3-642-20677-1_19. URL http://dx.doi.org/10.1007/978-3-642-20677-1_19.

Mun-Wai Chung and B. Drummond. Agile at Yahoo! From the Trenches. In *Agile Conference, 2009. AGILE '09.*, pages 113–118, Aug 2009. doi: 10.1109/AGILE.2009.41.

G. Cloke. GET YOUR AGILE FREAK ON! Agile Adoption at Yahoo! Music. In *Agile Conference, 2007. AGILE '07.*, pages 240–248, Aug 2007. doi: 10.1109/AGILE.2007.30.

C.L. Cowan. When the vp is a scrum master, you hit the ground running. In *Agile Conference, 2011. AGILE '11.*, pages 279–283, Aug 2011. doi: 10.1109/AGILE.2011.31.

J.W. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, 2013. ISBN 9781483321479. URL https://books.google.fi/books?id=EbogAQAAQBAJ.

N.K. Denzin and Y.S. Lincoln. *The SAGE Handbook of Qualitative Research*. Sage Handbook Of. SAGE Publications, 2011. ISBN 9781412974172. URL https://books.google.fi/books?id=qEiC-_ELYgIC.

Kim Dikert. Adopting agile software development in large organizations - A systematic literature review. Master's thesis, Aalto University, Finland, 2014.

Tore Dyba and T. Dingsoyr. What do we know about agile software development? *IEEE Software*, 26(5):6–9, Sept 2009. ISSN 0740-7459. doi: 10.1109/MS.2009.145.

Tore Dyb and Torgeir Dingsøyr. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50 (9–10):833 – 859, 2008. ISSN 0950-5849. doi: http://dx.doi.org/10.1016/j.infsof.2008.01.006. URL `http://www.sciencedirect.com/science/article/pii/S0950584908000256`.

Jessica Díaz, Jennifer Pérez, and Juan Garbajosa. Agile product-line architecting in practice: A case study in smart grids. *Information and Software Technology*, 56(7):727 – 748, 2014. ISSN 0950-5849. doi: http://dx.doi.org/10.1016/j.infsof.2014.01.014. URL `http://www.sciencedirect.com/science/article/pii/S0950584914000251`.

M. Evans. The FrAgile Organisation. In *Agile Conference, 2008. AGILE '08.*, pages 181–185, Aug 2008. doi: 10.1109/Agile.2008.80.

A Farrow and S. Greene. Fast & predictable a lightweight release framework promotes agility through rhythm and flow. In *Agile Conference, 2008. AGILE '08.*, pages 224–228, Aug 2008. doi: 10.1109/Agile.2008.83.

J. Fecarotta. Myboeingfleet and agile software development. In *Agile, 2008. AGILE '08. Conference*, pages 135–139, Aug 2008. doi: 10.1109/Agile.2008.72.

Melissa Federoff and Catherine Courage. Successful user experience in an agile enterprise environment. In MichaelJ. Smith and Gavriel Salvendy, editors, *Human Interface and the Management of Information. Designing Information Environments*, volume 5617 of *Lecture Notes in Computer Science*, pages 233–242. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-02555-6. doi: 10.1007/978-3-642-02556-3_27. URL `http://dx.doi.org/10.1007/978-3-642-02556-3_27`.

I Gat. How bmc is scaling agile development. In *Agile Conference, 2006. AGILE '06.*, pages 6 pp.–320, July 2006. doi: 10.1109/AGILE.2006.33.

J. Goos and A Melisse. An ericsson example of enterprise class agility. In *Agile Conference, 2008. AGILE '08.*, pages 154–159, Aug 2008. doi: 10.1109/Agile.2008.24.

H. Hajjdiab, AS. Taleb, and J. Ali. An industrial case study for scrum adoption. *Journal of Software*, 7(1):237–242, 2012.

S. Hanly, L. Wai, L. Meadows, and R. Leaton. Agile coaching in British Telecom: making strawberry jam. In *Agile Conference, 2006. AGILE '06.*, pages 9 pp.–202, July 2006. doi: 10.1109/AGILE.2006.13.

M.T. Hansen and H. Baggesen. From CMMI and Isolation to Scrum, Agile, Lean and Collaboration. In *Agile Conference, 2009. AGILE '09.*, pages 283–288, Aug 2009. doi: 10.1109/AGILE.2009.18.

Ville T. Heikkilä, Maria Paasivaara, Kristian Rautiainen, Casper Lassenius, Towo Toivola, and Janne Järvinen. Operational release planning in large-scale scrum with multiple stakeholders – a longitudinal case study at f-secure corporation. *Information and Software Technology*, 57:116 – 140, 2015. ISSN 0950-5849. doi: http://dx.doi.org/10.1016/j.infsof.2014.09.005. URL http://www.sciencedirect.com/science/article/pii/S0950584914002043.

VilleT. Heikkilä, Maria Paasivaara, Casper Lassenius, and Christian Engblom. Continuous release planning in a large-scale scrum development organization at ericsson. In Hubert Baumeister and Barbara Weber, editors, *Agile Processes in Software Engineering and Extreme Programming*, volume 149 of *Lecture Notes in Business Information Processing*, pages 195–209. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-38313-7. doi: 10.1007/978-3-642-38314-4_14. URL http://dx.doi.org/10.1007/978-3-642-38314-4_14.

Industrial Logic, Inc. Industrial XP. http://industrialxp.org/. Accessed: 2015-06-14.

J.P. Kotter. *Leading Change*. Harvard Business School Press, 1996. ISBN 9780875847474. URL http://books.google.fi/books?id=ib9Xzb5eFGQC.

M. Laanti. Implementing program model with agile principles in a large software development organization. In *Proceedings of the 32nd Annual IEEE International Computer Software and Applications, 2008. COMPSAC '08.*, pages 1383–1391, July 2008. doi: 10.1109/COMPSAC.2008.116.

E.C. Lee. Forming to performing: Transitioning large-scale project into agile. In *Agile Conference, 2008. AGILE '08.*, pages 106–111, Aug 2008. doi: 10.1109/Agile.2008.75.

Dean Leffingwell. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley Professional, 2010.

Leffingwell LLC. Scaled Agile Framework website. http://scaledagileframework.com/, a. Accessed: 2015-06-14.

Leffingwell LLC. Scaled Agile Framework - Case studies. http://www.scaledagileframework.com/case-studies/, b. Accessed: 2015-06-14.

A. Leszek and C. Courage. The Doctor is "In" – Using the Office Hours Concept to Make Limited Resources Most Effective. In *Agile Conference, 2008. AGILE '08.*, pages 196–201, Aug 2008. doi: 10.1109/Agile.2008.46.

J. Lewis and K. Neher. Over the Waterfall in a Barrel - MSIT Adventures in Scrum. In *Agile Conference, 2007. AGILE '07.*, pages 389–394, Aug 2007. doi: 10.1109/AGILE.2007.45.

K. Long and D. Starr. Agile supports improved culture and quality for healthwise. In *Agile Conference, 2008. AGILE '08.*, pages 160–165, Aug 2008. doi: 10.1109/Agile.2008.61.

C. Maples. Enterprise agile transformation: The two-year wall. In *Agile Conference, 2009. AGILE '09.*, pages 90–95, Aug 2009. doi: 10.1109/AGILE.2009.62.

R. Mencke. A product manager's guide to surviving the big bang approach to agile transitions. In *Agile Conference, 2008. AGILE '08.*, pages 407–412, Aug 2008. doi: 10.1109/Agile.2008.65.

Nils Brede Moe, Darja Šmite, Aivars Šāblis, Anne-Lie Börjesson, and Pia Andréasson. Networking in a large-scale distributed agile project. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '14, pages 12:1–12:8, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2774-9. doi: 10.1145/2652524.2652584. URL `http://doi.acm.org/10.1145/2652524.2652584`.

E. Moore and J. Spens. Scaling agile: Finding your agile tribe. In *Agile Conference, 2008. AGILE '08.*, pages 121–124, Aug 2008. doi: 10.1109/Agile.2008.43.

Jeff Nielsen and Dave McMunn. The agile journey. In Hubert Baumeister, Michele Marchesi, and Mike Holcombe, editors, *Extreme Programming and Agile Processes in Software Engineering*, volume 3556 of *Lecture Notes in Computer Science*, pages 28–37. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-26277-0. doi: 10.1007/11499053_4. URL `http://dx.doi.org/10.1007/11499053_4`.

Christopher P. O'Connor. Letters from the edge of an agile transition. In *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*, OOPSLA '10, pages 79–84, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0240-1. doi: 10.1145/1869542.1869557. URL http://doi.acm.org/10.1145/1869542.1869557.

C.P. O'Connor. Anatomy and physiology of an agile transition. In *Agile Conference, 2011. AGILE '11.*, pages 302–306, Aug 2011. doi: 10.1109/AGILE.2011.25.

M. Paasivaara and C. Lassenius. Deepening our understanding of communities of practice in large-scale agile development. In *Agile Conference, 2014. AGILE '14.*, pages 37–40, 2014. doi: 10.1109/AGILE.2014.18. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84910028133&partnerID=40&md5=a354a054ea2ce4b0afae4f11998020e9.

M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen. Towards Rapid Releases in Large-Scale XaaS Development at Ericsson: A Case Study. In *Proceedings of the IEEE 9th International Conference on Global Software Engineering (ICGSE), 2014*, pages 16–25, Aug 2014a. doi: 10.1109/ICGSE.2014.22.

M. Paasivaara, O. Väättänen, M. Hallikainen, and C. Lassenius. Supporting a large-scale lean and agile transformation by defining common values. *Lecture Notes in Business Information Processing*, 199: 73–82, 2014b. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84917708476&partnerID=40&md5=1e9541e09be2f2961423fd770ef9462e.

Performance Thinking Network. Six Boxes Model. http://www.sixboxes.com/. Accessed: 2015-02-28.

S. Prokhorenko. Skiing and boxing: Coaching product and enterprise teams. In *Agile Conference, 2012. AGILE '12.*, pages 191–196, Aug 2012. doi: 10.1109/Agile.2012.30.

P. Ranganath. Elevating Teams from 'Doing' Agile to 'Being' and 'Living' Agile. In *Agile Conference, 2011. AGILE '11.*, pages 187–194, Aug 2011. doi: 10.1109/AGILE.2011.40.

A.M. Razavi and R. Ahmad. Agile development in large and distributed environments: A systematic literature review on organizational, managerial and cultural aspects. In *Proceedings of the 8th Malaysian Software Engineering Conference (MySEC), 2014*, pages 216–221, Sept 2014. doi: 10.1109/MySec.2014.6986017.

P. Rodriguez, J. Markkula, M. Oivo, and K. Turula. Survey on agile and lean usage in finnish software industry. In *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2012*, pages 139–148, Sept 2012. doi: 10.1145/2372251.2372275.

Pilar Rodríguez, Kirsi Mikkonen, Pasi Kuvaja, Markku Oivo, and Juan Garbajosa. Building lean thinking in a telecom software development organization: Strengths and challenges. In *Proceedings of the 2013 International Conference on Software and System Process*, ICSSP 2013, pages 98–107, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2062-7. doi: 10.1145/2486046.2486064. URL http://doi.acm.org/10.1145/2486046.2486064.

J.J. Ryan and R. Scudiere. The price of agile is eternal vigilance. In *Agile Conference, 2008. AGILE '08.*, pages 125–128, Aug 2008. doi: 10.1109/Agile.2008.38.

B. Schatz and I. Abdelshafi. Primavera gets agile: a successful transition to agile development. *IEEE Software*, 22(3):36–42, May 2005. ISSN 0740-7459. doi: 10.1109/MS.2005.74.

Joachim Schnitter and Olaf Mackert. Large-scale agile software development at sap ag. In LeszekA. Maciaszek and Pericles Loucopoulos, editors, *Evaluation of Novel Approaches to Software Engineering*, volume 230 of *Communications in Computer and Information Science*, pages 209–220. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-23390-6. doi: 10.1007/978-3-642-23391-3_15. URL http://dx.doi.org/10.1007/978-3-642-23391-3_15.

T.R. Seffernick. Enabling agile in a large organization our journey down the yellow brick road. In *Agile Conference, 2007. AGILE '07.*, pages 200–206, Aug 2007. doi: 10.1109/AGILE.2007.23.

N. Sekitoleko, F. Evbota, E. Knauss, A. Sandberg, M. Chaudron, and H.H. Olsson. Technical dependency challenges in large-scale agile software development. *Lecture Notes in Business Information Processing*, 179 LNBIP:46–61, 2014. doi: 10.1007/978-3-319-06862-6. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84904563762&partnerID=40&md5=c81820c17170fbc798e0465d94ebf93f. cited By 0.

Mali Senapathi and Ananth Srinivasan. Sustained agile usage: A systematic literature review. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, EASE '13, pages 119–124, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1848-8. doi: 10.1145/2460999.2461016. URL http://doi.acm.org/10.1145/2460999.2461016.

K. Silva and C. Doss. The growth of an agile coach community at a fortune 200 company. In *Agile Conference, 2007. AGILE '07.*, pages 225–228, Aug 2007. doi: 10.1109/AGILE.2007.56.

H. Smits and K. Rilliet. Agile experience report: Transition and complexity at cisco voice technology group. In *Agile Conference, 2011. AGILE '11.*, pages 274–278, Aug 2011. doi: 10.1109/AGILE.2011.14.

M.K. Spayd. Evolving agile in the enterprise: implementing xp on a grand scale. In *Proceedings of the Agile Development Conference, ADC 2003.*, pages 60–70, June 2003. doi: 10.1109/ADC.2003.1231454.

J. Sutherland and R. Frohman. Hitting the wall: What to do when high performing scrum teams overwhelm operations and infrastructure. In *Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS), 2011.*, pages 1–6, Jan 2011. doi: 10.1109/HICSS.2011.222.

Kevin Vlaanderen, Peter van Stijn, Sjaak Brinkkemper, and Inge van de Weerd. Growing into agility: Process implementation paths for scrum. In Oscar Dieste, Andreas Jedlitschka, and Natalia Juristo, editors, *Product-Focused Software Process Improvement*, volume 7343 of *Lecture Notes in Computer Science*, pages 116–130. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-31062-1. doi: 10.1007/978-3-642-31063-8_10. URL `http://dx.doi.org/10.1007/978-3-642-31063-8_10`.

R.K. Yin. *Case Study Research: Design and Methods*. SAGE Publications, 2013. ISBN 9781483322247. URL `https://books.google.fi/books?id=0gyqBAAAQBAJ`.

First interview outline

1. **Interviewee**
   (a) Background (shortly)
   (b) Education
   (c) History at Rally
   (d) Role and tasks in ALM product line and/or team
2. **Overview**
   (a) How do you see the ALM teams' current situation in general? How are you doing?
   (b) How do you see your own team's current situation? How are you doing?
   (c) What's good / bad?
   (d) What is the direction you see you should go? What's the goal?
3. **Agile methods**
   (a) How individuals have understood the agile principles?
   (b) What practices do you use? How are those practices demonstrated in your daily work?
   (c) Are teams using text-book methods or have you modified practices to fit better to teams' needs?
   (d) What have you modified? Why? How? How well does it work?
   (e) Is there resistance towards change? Why? Any particular things people resist?
4. **Sprints and quarterly initiatives**
   (a) What actions are performed in sprint phases? (e.g. sprint planning, sprint review, retrospective)

   (b) What release practices do you have?
   (c) How are quarterly initiatives and sprints monitored?

5. **Challenges**
   (a) What are the biggest challenges at the moment?

6. **Improvement suggestions**
   (a) If you could decide what improvements to make, what would they be? Why?

7. **Trust**
   (a) Do people trust each other?
      i. In your team?
      ii. Do teams trust each other?
      iii. Do you trust management?
   (b) How easily are problems communicated?
   (c) Are individuals getting enough feedback on their work? Are individuals able to give feedback? (e.g. about organizational/team issues/etc)

8. **Agile at the organizational level**
   (a) What are the challenges in maintaining organization wide agile?

Second interview outline

1. **Interviewee**
   (a) Role and tasks in ALM product line and/or team (if changed)
2. **Overview**
   (a) How do you see the ALM teams' current situation in general? How are you doing?
   (b) How do you see your own team's current situation? How are you doing? What's good / bad?
   (c) What is the direction you see you should go? What's the goal?
3. **Agile methods**
   (a) Have you made any adjustments since the last interview? What have you modified? Why? How? How well does it work?
   (b) Is there resistance towards change? Why? Any particular things people resist?
4. **Challenges**
   (a) What are the biggest challenges at the moment?
5. **Improvement suggestions**
   (a) Have you brought up any improvement suggestions since the last interview? Have your improvement suggestions been implemented?
   (b) If you could decide what improvements to make, what would they be? Why?
6. **Trust**
   (a) Do people trust each other?

   i. In your team?
   ii. Do teams trust each other?
   iii. Do you trust management?

(b) How easily are problems communicated?

(c) Do you think there has been changes in trust since the last interview? Why?

7. **Agile at the organizational level**

(a) How do you see Rally's situation now? Has anything changed since the previous interview?

(b) What are the challenges in maintaining organization wide agile?