

Organization-Wide Agile Expansion Requires an Organization-Wide Agile Mindset

Hidde van Manen and Hans van Vliet

Computer Science Department, VU University Amsterdam, The Netherlands
hiddevanmanen@hotmail.com, hans@cs.vu.nl

Abstract. While agile methods are widely used, large organizations still struggle with the implementation thereof throughout the whole organization. The objective of our study is to identify factors that affect the expansion of agile software development in large organizations. We performed a multiple-case study to do so. We found agile software development in large organizations is more than implementing Scrum. In particular, we identified "agile mindset" as a crucial topic that deserves attention when expanding agile methods in large organizations.

Keywords: agile expansion, Scrum, agile mindset.

1 Introduction

Since their origin in February of 2001, agile software development methods have become immensely popular. While Forrester research from 2010 indicated that agile methods were practiced in more than a third of all projects [1], this percentage has kept growing strongly in the last few years and is now well over eighty percent of all projects [2]. Surely this cannot be the case in every organization? No. A quick internet search teaches us that large, international organizations all over the globe are still in the process of transforming their development organization towards agile software development – and that this change is not realized overnight. An example is the U.S. Postal Service, which has spent well over three years rolling out agile in its organization and recently announced that agile has officially replaced the waterfall methodology in March 2013¹.

Agile Methods, when referred to in this paper, encompass software development methodologies characterized by a continuous readiness to rapidly realize change, pro-actively or reactively embrace change, and learn from change while contributing to customer value. In particular, Scrum, the development methodology used in the environments studied in this paper, fits this definition. A focus on working code right from the beginning, delivery cycles of 2-4 weeks (so-called Sprints), and having business representatives on the team are but three practices to achieve the above benefits [3].

¹ <http://fcw.com/articles/2013/06/13/usps-agile-development.aspx> - accessed on June 25, 2013.

The introduction and adoption of agile methods within organizations is a popular subject of research into agile methods [4]. The number of anecdotal and qualitative studies into the challenges of such adoption is large and studies have been performed both over short and long periods of time [5–13]. The findings of these studies are not always consistent [7]. On the other hand some factors are commonly accepted as being critical to the success of agile adoption, such as "management support" and "customer collaboration" [7, 9, 11].

One of the items on the current research agenda is to study agile software development within organizations that have left the so-called adoption phase [14]. Recently a few publications on this topic have seen the light, e.g. [15], but a lot of questions remain open. There is an increasing need for knowledge about this topic, as *"many organizations have completed the adoption stage and agile methods start to become well-established processes of these organizations"* [16]. The expansion of agile methods in large organizations is another one of the issues to be addressed and this is also the topic of our research.

Already in the early days of agile, some publications report practitioners struggling with scaling agile methods, e.g. [17]. What they describe is the demand to balance the new agile methods with the document-driven (waterfall) approach that large organizations require. Boehm, Beck and Turner [18, 19] were involved in a discussion on balancing agile methods with traditional document-driven methods in large organizations, in order to achieve the benefits of both. They state: *"both approaches have shortcomings that, if left unaddressed, can lead to project failure. The challenge is to balance the two approaches to take advantage of their strengths and compensate for their weaknesses"* [18, 19]. It seems rather hard to adopt agile methods on an enterprise-wide scale in large organizations. Agile methods in general, including Scrum, focus more on the team level and less on organizational issues. When large organizations try to adopt agile methods, they face a myriad of issues that are different compared to individual teams adopting agile and which make the organizational adoption a complex and hard journey. In every organization agile methods are implemented in a unique context that is the result of a combination of organizational, process, human and technological factors. Also, the challenges concerning the use of agile methods change when agile transforms from being a small experiment to being the main method used company-wide.

When we speak of agile expansion or scaling, we mean that agile methods are for example extended from one organizational unit to other organizational units, or from an initial small project to larger and more complex projects, which results in challenges of a new kind [15, 20]. Multiple authors describe and suggest that there are many issues and challenges that arise when agile is implemented and adopted at the organizational level, such as synchronization of agile and non-agile (document-driven) functions [15, 16, 21, 22].

Beyond Budgeting is complementary to the agile way of working. It is a performance management method oriented to fast changing environments, rather than strict control mechanisms [23]. Traditional management adopts a command-and-control way of thinking, whereas Beyond Budgeting adopts a sense-and-control

way of thinking. The complementarities between Beyond Budgeting and Agile software development are discussed in [24].

Agility can also be linked to Lean Software Development. Lean manufacturing was introduced in the 1990's in the Japanese car industry. The concepts of lean manufacturing and agile software development were combined by Mary and Tom Poppendieck [25]. Lean software development is built around seven principles: eliminate waste, build quality in, create knowledge, defer commitment, deliver fast, respect people, and optimize the whole. Williams [26] gives an elaborate comparison of agile and lean production, while [27] gives an experience report on the application of lean approaches in agile software development.

The research question we aim to answer is: *Identify factors that affect the expansion of agile development in large organizations*. We performed a multiple-case study to do so. We found agile software development in large organizations encompasses more than implementing Scrum. We identified a number of issues that deserve attention when expanding agile methods in large organizations. We grouped them in two broad categories: "agile mindset" and "contextual dependencies". For lack of space, we only discuss the "agile mindset" issues in this paper. Example contextual dependencies identified are: "agility of partner organizations", "governance procedures", and "top management agile vision".

2 Research Method

The research we have performed was a multiple-case study. The case study method is best suited to develop an understanding of the interactions among information technology innovations and organizational context [28]. Multiple-case designs enable generalizability and the ability to extend theory through cross-case analyses [29]. Some argue with this, because the characteristics of each case are unique and this limits the external validity of comparing cases and making generalizations [30]. However, Walsham [31] argues that generalization is not necessarily the primary goal of studying multiple cases. More important is the deep insight that enables the researcher to develop concepts and theory based on information from many sources and understanding of the context.

2.1 Data Collection and Analysis

Since the focus of our research was on the expansion of agile methods in large organizations, we needed to study organizations of a certain scale that were in the process of expanding agile software development activities within their organization. We selected organizations that had been working on this expansion for quite some time, so they could discuss their previous efforts, struggles and results with us, because we were unable to perform a longitudinal study. Stepping in and following their efforts and progress over time was not an option. Thus, we selected two large multi-nationals based in the Netherlands which had both been expanding agile methods for at least 1,5 years at the time of our research. Besides the fact that both companies met our requirements of scale and being in the

process of expansion, our selection of companies and sub-units was opportunistic. Due to time constraints, we have selected the first two companies that were positive about participating in our research. We selected two different types of organizations, a consumer electronics company and a bank, to increase diversity of the organizational and business context. Within the companies we limited the scope of our research to respectively a change program and a business line, because of the limited resources (one researcher and two months time) that were available for our research. The companies themselves selected the program and business line. The characteristics of the two companies are listed in Table 1.

Table 1. Case company characteristics

| | Company A | Company B |
|-----------------------------------|--|---|
| Company background | Consumer electronics | Bank |
| Unit type | Program | Business line |
| Size | 12 teams, 90 persons | 7-8 teams, 80 persons |
| Direct organizational environment | 100 teams, separated in three sectors and cross-sector | 400 employees, spread over five business lines |
| Type of system developed | Internal profitability measuring system for controlling department | Internal back-office systems for global financial markets |
| Location | Co-located teams spread over several floors | Co-located teams on one big floor |
| Development method | Scrum | Loose Scrum |
| Years of experience with agile | 1,5 years | Three years ago Open UP, one year ago "agile the next step" |

The main source of data collection for our research was through face-to-face interviews. In both organizations we interviewed participants that have different roles within the organization in order to collect information from different angles and perspectives. We selected participants who were part of the agile development teams themselves or were directly involved with the teams that went through the transformation. We interviewed eight persons in Company A, and ten in Company B. Table 2 lists the interviewees, their role, and their experience with agile. For Company B, the business line that was subject to our research adopted several agile practices over a period of two years, such as standup meetings and work boards, but not others, such as iteration development or planning meetings. For that reason, a clear starting point for the experience with agile often cannot be given. All participants are from Dutch sites, except for the Operations Team Lead of Company A. The interviews took place in Spring 2013.

The interviews were semi-structured and conducted following the same template. The interviews lasted between 42 and 82 minutes with an average duration of approximately 62 minutes. The interviewer prepared an interview guide containing topics to discuss with participants. The interview guide served as a

Table 2. Participants overview

| Company | Role | Experience with Agile |
|-----------------------|--|---|
| Company A | Delivery manager | 18 months |
| | Business analyst | 12 months |
| | Member agile work group, agile coach | 36 months (18 at another company) |
| | Business stakeholder | 18 months |
| | Scrum master, agile coach, product owner support | 18 months |
| | Account manager | n.a. |
| | Operations team lead | n.a. |
| Company B | Environment manager | 3.5 years (1.5 Scrum master at another company) |
| | Agile coach | 1 year |
| | Manager projects | Not clear |
| | Team manager support | Not clear |
| | Business manager | Not clear |
| | Team manager development | Not clear |
| | Lead business analyst | Not clear |
| | Change & release manager | 1 year Scrum master |
| | Tester | 1 year |
| Business line manager | Not clear | |

structure for the interview, as well as to make sure that the interviewer covered all topics and kept the right scope. What follows is a list of the topics that were on the interview guide:

- Description of the employees role in the organization, his/hers experience with agile software development and general information about the project and organization he/she is currently working on;
- Benefits of agile software development that the participant has (and has not) seen in the organization;
- Expansion of agile - what differences does the participant experience between agile at small (team) and large (organizational) scale;
- What challenges has the organization encountered in the expansion of agile. Possible topics: (organizational) processes, people, business, agile method, tools and (change) management;
- In case the participant had not made clear what he/she deemed to be most important, we asked that if the participant could suggest one change or give one advice to his/her CIO regarding the expansion of agile software development, what it would be.

The interview concluded with a summary of highlights from the interview, based on the interviewer's notes, on which the interviewee could make remarks. Then each participant was given the possibility to add anything to the record without

the researcher’s possible bias. The questions in the interview were mostly open-ended, to stimulate interviewees to tell a story in its context, which is important in the context of agile software development. The interview setting allowed the interviewer to explain questions or ask follow-up questions in case this was necessary and thus prevent inaccurate answers. Closed questions were used to confirm facts or statements that were made to the researcher in earlier interviews or to challenge initial analyses of the researcher.

In addition to the data gathered through personal interviews, the researcher has made on-site observations during daily stand-ups, retrospective and Scrum review meetings and walking around the office locations on several occasions. These observations and meetings were not recorded, but notes were taken. Also, we received some files from interviewed persons, including (but not limited to) organizational diagrams, documents about the agile vision, an agile maturity (evaluation) model, weekly report forms and a dashboard for reporting agile expansion KPI’s. The last source of data came from sketches made during interviews by interviewees, e.g. about the project initiation process or to illustrate a timeline of events.

From the initial transcripts of the interviews, analysis went through the following steps:

- Initial coding: assigning codes to all relevant statements, catching the essence of what was said. The result was close to 100 codes per company.
- Focused coding, developing ”concepts”: eliminating, combining, or subdividing codes, looking for repeating ideas and concepts. This approach was based on [32]. We first grouped codes per person, and next merged similar concepts from multiple interviewees.
- Pattern coding, developing ”categories”: This step aims to aggregate and summarize the previous coding, identifying themes across all.
- Constructing theory: based on a comparison of the results for the two cases, two findings emerged: the importance of an ”agile mindset” and ”contextual dependencies”. The researcher then went back to consult relevant literature on these two topics.

3 Results and Discussion

We identified two broad topics that are, according to participants of our research, important in order to successfully expand agile software development within a large organization: ”agile mindset” and ”contextual dependencies”. We next went back to the codes and concepts from our research to identify factors that are positively or negatively related to the topics agile mindset or contextual dependencies. In this section we will go through this analysis for the ”agile mindset” topic and relate our observations to existing literature.

Throughout our interviews participants in both companies deemed an agile mindset crucial to the successful expansion (i.e. the potential benefits of the agile way of working are obtained) of agile methods through the organization as a whole. At Company A participants mainly stated that although the development

teams had adopted agile quickly and started to think more and more in agile ways, the business client and partner companies had more trouble changing their mindset. In Company B we heard often that the general resistance employees had towards working agile was partially based on a lack of an agile mindset, which in turn was influenced by an extensive experience with non-agile methods. In both companies participants said that managers had too little knowledge of agile or lacked an agile mindset.

We analyzed our interview records to identify factors that participants described in relation to an agile mindset. In this analysis three issues emerged, which form an important part of the agile mindset. These are 'collaboration', 'trust' and 'continuous improvement'. In other words, if there is no collaboration, trust or wish for continuous improvement, then there exists no agile mindset within an organization.

Figure 1 shows the result of our analysis on how these three issues are influenced by organizational and cultural factors. The numbers between brackets indicate how often these factors were mentioned during our interviews (multiple instances per participant are counted separately) and the symbols along the arrows illustrate if a relation is positive (+) or negative (-). We only include factors that were mentioned more than once. We discuss each issue and its constituent factors:

Collaboration

- **Competing "partners" structure (-):** Having multiple competing partner companies in the development process negatively affects collaboration. These partners have a separate goal and that is to win more contracts than other partners. Also this construction demotivates knowledge sharing across the organization, as a participant from Company A explained to us: *"It would be stupid to share all our knowledge with [the other partner companies], because then what will our advantage over them be? Why would Company A choose us for the next project?"* On the other hand, a participant from Company B told us that having a "real" partnership with for example your support organization could boost collaboration. A "real" partnership means that you discuss both positive and negative things with each other. By doing this, the relationship is not only made up by discussions about what is going wrong, but can be enriched by talking about the progress you made together.
- **Serial work process (-):** A serial work process negatively affects collaboration, because during each step of the process, someone who is part of the process usually has less contact with other contributors of the process, but instead just waits until he gets something delivered from the person that precedes him in the chain.
- **Individual thinking (of people, teams, departments) (-):** Individual thinking was the factor most mentioned as influencing collaboration. When for example a team is focused too much on itself and not on the other teams they are working with to complete a product, this can lead to misalignment

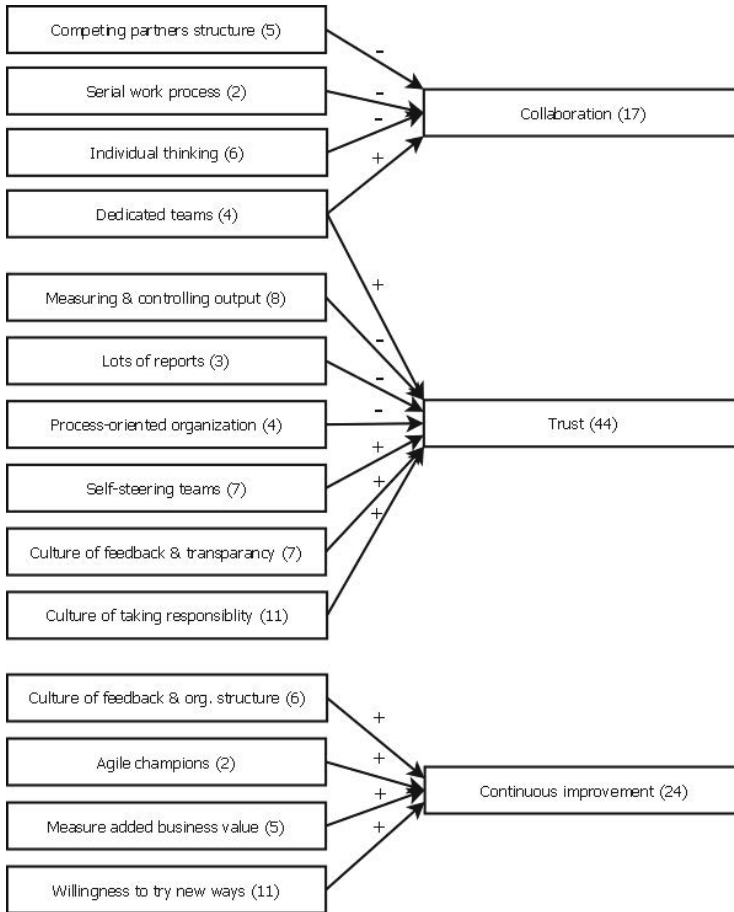


Fig. 1. Factors influencing agile mindset

of activities. Another example comes from Company A, where it is clear that the development organization and support organization are both pursuing their own goals, while they could improve their collaboration by working together as DevOps teams. One has to balance one's identity as a team member versus one's role in the organization at large [33].

- **Dedicated teams (with experience as one team) (+):** Having a team fully dedicated to one development stream increases collaboration, because team members are always available and do not have to switch between teams, environments and scopes. Also, experience as a team increases collaboration, because team members get to know each other's strengths and weaknesses. Participants from both companies condemned the first period of agile development, because they did not have dedicated teams then, as the following quote illustrates: *"After a few months of successful Scrumming, they dis-*

banded all the teams and spread all the members over different projects. We had to start all over again. That was a huge loss of knowledge." The reorientation required in a transition from individual work to a self-managing agile team is extensively discussed in [34].

Trust

- **Dedicated teams (with experience as one team) (+):** With increased dedication and experience as a team also comes increased trust, according to some participants. It is harder for team members to build trust if they only spend a fraction of their time together working on one collaborative product. A dedicated team is also more trustworthy for management, because the team need not spend time onboarding new team members at the beginning of each project. This also results in more stable output of work.
- **Measuring & controlling output (–):** Measuring the output (story points) of development teams and accounting teams on it, is a wrong idea according to multiple participants of both companies. *"Beforehand [management] states how many story points a team has to earn in each Sprint. If the team does not reach its goal, the partner company gets a fine. I do not think that is not very agile-minded. The team needs to earn at least 80% of your contract points, or the partner will not get paid. This has an effect on teams. [At the start of agile] people were enthusiastic at the end of a sprint, picking up new stories to work on. Now developers are more anxious about starting new stories, because if they cant finish them, they will not get paid. This has definitely affected peoples mindset.* Instead of focusing on points and giving the teams the idea that their performance is controlled, companies should focus on added business value and customer satisfaction. Part of trust is trusting teams that they will do the best possible job they can. Misra et al [10] also observed that a more qualitative control leads to more success in agile projects. In terms of the Beyond Budgeting philosophy [35]: *"The main goal of Beyond Budgeting is not to get rid of budgets [...] but it is more the budgeting mindset we need to get rid of.*
- **Lots of reports (–):** The amount of reports that a team has to file is a nice indicator for trust, say participants. *"The fact that I have to report to approximately ten different managers each week, does not give me the feeling that they trust our team"*, is how a participant explained the importance of reports to us.
- **Process-oriented organization (–):** If teams have to follow extensive processes and cannot change them this does not positively affect the feeling of trust in the organization. *Our organization needs a planning. When we file a project plan, we still need a Project Initiation Document (PID) with a decent reasoning on how long your project will take. And you cannot deviate from that too much. Another example is that IT development now wants to count with story points, but we are paying in euros, so you still need a translation back from story points, to working hours, to budgets. Although we are working agile, our planning is still waterfall, with all related phases. I*

believe it will stay that way. In an agile organization, there should be room to adjust processes and for teams to make their own choices, instead of having an extensive process in place for each particular situation.

- **Self-steering teams & facilitating management style (+):** On the other hand, if an organization allows teams to steer themselves and management has a facilitating attitude towards teams, this has a positive effect on trust in the organization. This is corroborated by [34] and [36], amongst others. Moe et al [37] identify several organisational barriers to self-management at the team level, such as the quest for organizational control and a culture of specialization. One important aspect of the facilitating management style is need for support and budget from the business to put qualified product owners on all development teams.
- **Culture of feedback & transparency (+):** Having a culture of feedback in the organization and being transparent across teams and departments positively influences trust. *We had our CIO at a product demo, and it was a Sprint in which not everything had gone right. I showed him [our problems] and he said he was happy to see that, because [he said:] all the demos that I go to, everything goes flawless. Apparently people do not dare and want to show what goes wrong. I think this is company culture. Everyone reports to his/her manager that things go well. For some reason, it is not done to report that something is not going well here, while I have seen this differently at other companies I've worked for.* One participant explained how her organization lacks this: *"I think that at this moment there is no attention for organizational impediments at higher levels, because these impediments are not brought up in the first place. A lot of people are anxious about reporting impediments, because they have experiences where bringing up these issues was ... not well received."*
- **Culture of taking responsibility (+):** The agile mindset element of trust relies heavily on employees taking responsibility, say participants. *Agile is a certain mindset. It means that you give responsibility to employees and [management] has to be open to that. You have to reduce the amount of controls, and give people empowerment to take action and responsibility. If you keep all your control structures and your blame culture in place that will conflict with agile. Only following the method is not enough, you have to look at the idea behind the method and adjust your controls accordingly.* Instead of hiding behind processes people should build ownership over development processes and take responsibility of emerging problems. This helps to solve these issues quicker and more effectively, while it also increases people's trust in each other. Creating such a culture is both seen as a challenge [33] and as a success factor [10]. Taking responsibility also is one of the leadership principles of Beyond Budgeting [24].

Continuous Improvement

- **Culture of feedback and organizational structure (+):** A culture of feedback accompanied by appropriate organizational structures to cultivate

this feedback and use it will positively affect continuous improvement. If these organizational structures are not in place, it can lead to disappointment among teams that spend time evaluating their work and providing constructive feedback about processes, say participants. Open constructive feedback is essential to the continuous improvement of the organization.

- **Agile champions (+):** The role of the business line management should be to address the resistance among employees, solve problems that hinder the agile expansion, change processes that do not fit the new way of working, train and coach people and facilitate the coordination with parties that the development depends on. However, multiple participants of our research mentioned that they miss a clear vision of the management. It is clear that agile is the way to go, but how, why, when and in what way the agile expansion should take place is unclear for them. Small changes are pervasive, but there is no coherent story. *There is no clear vision, no clear line. I think it is messy; some practices are picked and we are only doing agile partially. We are not rallying behind a choice and clearly going somewhere. We need someone to say: this is where we are going and this is how we are going to get there. I miss such a vision.* People that support the agile mindset and put effort in spreading it through the organization can have a lasting positive effect on continuous improvement[38].
- **Measure added business value (+):** Measuring added business value over costs focuses attention on continuous improvement of products, thus increasing the overall attention of continuous improvement in the organization. Continuous improvement is a key characteristic of both Beyond Budgeting and Lean Software Development.
- **Willingness to try new way of working (+):** Finally, the willingness of employees to try a new way of working also influences the continuous improvement of the organization. If employees are less willing to innovate their work processes, this reduces the possibilities for continuous improvement. The importance of continuous learning is also noted in [10, 39]. In the business line of one of the companies, there was a lot of resistance against agile among employees. This is partially a consequence of not making the change completely (but step-by-step), because this allows people to challenge the working method and its benefits. Improvements are not attributed to agile, while all issues give opponents more reason to complain. As a primary reason for the resistance, employees say that the why of the agile scaling was never clearly explained, as the following participant explained: *Change is not a problem as long as it is clear why it is needed. I hear a lot of people ask: Why? Did things go bad? Are there complaints? What is the goal of implementing agile? Is the goal to work agile, or are we trying to achieve certain benefits? That is not really clear.* To educate employees about agile and to create support, all employees had to follow an Agile Awareness training and make an Agile Foundation Exam, but they perceived these to be rather useless and not applicable in their work. There is disunity in teams, because some team members are opponents while other oppose agile. This resistance is not unwillingness; it is mostly habituation of the old way of working.

In our team we started to rollout agile. There are supporters and opponents. I have the idea that people are not yet convinced and that makes it hard to implement. () We have been discussing this for three weeks now, but if only the testers want to do it, we cannot do it. It only works when the whole team is behind it. Others do not think it is efficient, that it will not save them time. There also seems to be a difference between more and less experienced employees, where less experienced employees have a more positive attitude towards the agile expansion, which is also found in earlier research [40].

It is interesting to see that something intangible as "trust" seems to be at the core of the agile mindset, as is also observed by Moe [34], McHugh [41] and Strode [39]. The importance of trust was named even more than the other two main elements combined. More than the other two elements, trust is also affected by or reflected in organizational structures; one has to carefully balance control and flexibility [20]. This indicates that while the agile mindset is primarily a psychological issue, it is in fact quite related to organizational elements such as the reporting processes.

In order to successfully expand agile throughout the organization, the organization should foster an agile mindset among its employees. This involves developers, managers, and other stakeholders. Whitworth [42] researched the social nature of agile teams, and identified several characteristics of agile teams that stand out: (1) openness and respect, (2) a strong inclination for whole team consideration and involvement, and (3) highly value action, initiative and continuous improvement. Such is also reflected in the three core values of the agile mindset as identified in our research.

This agile mindset should exceed the agile team. Van Waardenburg [43] already mentions the importance of an agile mindset amongst business representatives. When the agile culture is limited to the IT department, it will only lead to frustration when working with other, non-agile, parts of the organization.

4 Limitations

Our research was more of a snapshot than an image of the organizations over time. A longitudinal study would have given us the chance to further validate our findings. This was not a possibility, as only limited time and resources were available to perform the research.

Both organizations that participated in our research used the Scrum method as their agile method. While Scrum is the most-used method in the world of agile, this limits the external validity of our researchs findings to organizations using other methods. Organizations that use other methods, which for example more directly address issues discussed in this thesis, may differ from our findings.

5 Conclusion

In this paper, we studied the expansion of agile methods in large organizations. We found the impact of agile on the organization goes much further than just the

development teams that practice Scrum (or a variant). If an organization wants to become completely agile and achieve all the potential benefits of an agile way of working, this requires a certain mindset of people throughout the organization and an adaptation of the agile implementation based on the contextual dependencies along three perspectives (The latter is not further elaborated in the present paper). This conclusion is in line with results from a recent global survey on agile development by VersionOne. When asked what barriers existed to further adoption of agile methods in the enterprise, ranked one (selected by 52% of the participants) was the "ability to change organizational culture", followed in third place (35%) by "trying to fit agile elements into a non-agile framework" [2].

Through analysis of both our participants thoughts on the agile mindset and existing literature on this subject, we claim that the agile mindset's main elements are:

- **Trust:** all employees should take responsibility for changes and issues, as they are empowered and trusted by the management to make their own decisions, while the organizational structure and processes reflect this trust;
- **Continuous improvement:** everyone in the organization strives for continuous improvement of all processes, people, and products, by maintaining an open attitude towards each others feedback, and
- **Collaboration:** all results and improvements are achieved through intensive collaboration of everyone in the organization.

Being a truly agile organization requires more than the implementation of an agile method such as Scrum. Being agile is a mindset based on trust, collaboration and continuous improvement. We would like to end our conclusions with a quote from Ivar Jacobson:

Agility should penetrate everything you do. It penetrates the way you should manage and do requirements, architecture, design, coding, integration and testing, and in the way that you should document and track what you do. () It reaches all levels in the company from upper level managers down to the developers. () Agile is an attitude that everyone must embrace. ²

Acknowledgement. We gratefully acknowledge the constructive feedback of Torgeir Dingsøy on an earlier version of this paper.

References

1. West, D.: Water-Scrum-Fall is the Reality Of Agile for Most Organizations Today. Forrester (2011)
2. VersionOne: 7th annual state of agile development survey (2013)

² Ivar Jacobson International, Scaling Agile WP, <http://www.ivarjacobson.com/resource.aspx?id=402>, accessed May 17th, 2013

3. Schwaber, K., Sutherland, J.: The Scrum Guide: The official rulebook. Technical Report (October 26, 1991), <http://www.scrum/scrumguides/>
4. Dybå, T., Dingsøy, T.: Empirical studies of agile software development: A systematic review. *Information and Software Technology* 50(9-10), 833–859 (2008)
5. Boehm, B.: Get Ready for Agile Methods, with Care. *Computer* 35, 64–69 (2002)
6. Bottani, E.: Profile and enablers of agile companies: An empirical investigation. *International Journal of Production Economics* 125, 251–261 (2010)
7. Chow, T., Cao, D.B.: A surevy study on critical success factors in agile software projects. *Journal of Systems and Software* 81, 961–971 (2008)
8. Cockburn, A., Highsmith, J.: Agile software development, the people factor. *Computer* 34, 131–133 (2001)
9. Livermore, J.: Factors that significantly impact the implementation of an agile software development methodology. *Journal of Software* 3, 31–36 (2008)
10. Misra, S.C., Kumar, V., Kumar, U.: Identifying some important success factors in adopting agile software development practices. *The Journal of Systems and Software* 82, 1869–1890 (2009)
11. Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of Migrating to Agile Methodologies. *Communications of the ACM* 48, 73–78 (2005)
12. Schmidt, C.T., Venkatesha, S.G., Heymann, J.: Empirical Insights into the Perceived Benefits of Agile Software Engineering Practices: A Case Study from SAP. In: *Companion Volume Proceeding 36th International Conference on Software Engineering (ICSE)*, pp. 84–92. IEEE Computer Society (2014)
13. Laanti, M.: Agile Methods in Large-Scale Software Development Organizations – Applicability and Model for Adoption. PhD thesis, University of Oulu, Oulu, Finland (2012)
14. Dingsøy, T., Nerur, S., Balijepally, V., Brede Moe, N.: A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software* 85, 1213–1221 (2012)
15. Senapathi, M., Srinivasan, A.: Understanding post-adoptive agile usage: An exploratory cross-case analysis. *Journal of Systems and Software* 85, 1255–1268 (2012)
16. Abrahamsson, P., Conboy, K., Wang, X.: ‘lots done, more to do’: The current state of agile systems development research. *European Journal of Information Systems* 18, 281–284 (2009)
17. Reifer, D.: Scaling agile methods. *IEEE Software* 20(4), 12–14 (2003)
18. Beck, K., Boehm, B.: Agility through discipline: A debate. *Computer* 36(6), 44–46 (2003)
19. Boehm, B., Turner, R.: *Balancing Agility and Discipline*. Addison Wesley (2002)
20. Mishra, D., Mishra, A.: Complex software project development: Agile methods adoption. *Journal of Software Maintenance and Evolution: Research and Practice* 23(8), 549–564 (2011)
21. Mangalaraj, G., Mahapatra, R., Nerur, S.: Acceptance of software process innovations – the case of extreme programming. *European Journal of Information Systems* 18(4), 344–354 (2009)
22. Moe, N.B., Dingsøy, T.: Research Challenges in Large Scale Agile Software Development. *ACM SIGSOFT Software Engineering Notes* 38(5) (2013)
23. Bogsnes, B.: *Implementing Beyond Budgeting: Unlocking the Performance Potential*. Wiley (2008)

24. Lohan, G., Conboy, K., Lang, M.: Beyond Budgeting and Agile Software Development: A Conceptual Framework for the Performance Management of Agile Software Development Teams. In: Proceedings of the 31st International Conference on Information Systems (ICIS 2010) (2010)
25. Poppendieck, M., Poppendieck, T.: Implementing Lean Software Development: From Concept to Cash. Addison-Wesley (2007)
26. Williams, L.: Agile Software Development Methodologies and Practices. In: Advances in Computers, vol. 80, pp. 4–44. Elsevier (2010)
27. Wang, X., Conboy, K., Cawley, O.: “Leagile” software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software* 85(6), 1287–1299 (2012)
28. Seaman, C.: Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering* 25(4), 557–572 (1999)
29. Miles, M., Huberman, A.: *Qualitative Data Analysis: An Expanded Sourcebook*. Sage, London (1994)
30. Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., Emam, K.E.: Preliminary Guidelines for Empirical Research in Software Engineering. *IEEE Transactions on Software Engineering* 28, 721–734 (2002)
31. Walsham, G.: Interpretive Case Studies in IS Research: Nature and Method. *European Journal of Information Systems* 4(4), 74–81 (1995)
32. Bogdan, R., Biklen, S.: *Qualitative Research for Education: An introduction to Theories and Methods*, 4th edn. Pearson (2003)
33. Moore, E., Spens, J.: Scaling Agile: Finding your Agile Tribe. In: Agile 2008 Conference, pp. 121–124. IEEE Computer Society (2008)
34. Moe, N.B., Dingsøy, T., Dybå, T.: A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology* 52, 480–491 (2010)
35. Bogsnes, B.: Keynote: Beyond Budgeting in a Lean and Agile World. In: Abrahamsson, P., Marchesi, M., Maurer, F. (eds.) XP 2009. LNIP, vol. 31, pp. 5–7. Springer, Heidelberg (2009)
36. Vidgen, R., Wang, X.: Coevolving Systems and the Organization of Agile Software Development. *Information Systems Research* 20(3), 355–376 (2009)
37. Moe, N.B., Dingsøy, T., Dybå, T.: Overcoming Barriers to Self-Management in Software Teams. *IEEE Software* 26(6), 20–26 (2009)
38. Benefield, G.: Rolling out Agile in a Large Enterprise. In: Proceedings of the 41st Hawaii International Conference on System Sciences (HICSS 2008), p. 462. IEEE Computer Society (2008)
39. Strode, D.E., Huff, S.L., Tretiakov, A.: The Impact of Organizational Culture on Agile Method Use. In: Proceedings of the 42nd Hawaii International Conference on System Sciences, pp. 1–9. IEEE Computer Society (2009)
40. Vijayarathy, L., Turk, D.: Drivers of agile software development use: Dialectic Interplay between benefits and hindrances. *Information and Software Technology* 54(2), 137–148 (2012)
41. McHugh, O., Conboy, K., Lang, M.: Agile Practices: The Impact on Trust in Software Project Teams. *IEEE Software* 29(3), 71–76 (2012)
42. Whitworth, E., Biddle, R.: The Social Nature of Agile Teams. In: Agile 2007, pp. 26–36. IEEE Computer Society (2007)
43. Waardenburg, G.V., van Vliet, H.: When Agile meets the Enterprise. *Information and Software Technology* 55(12), 2154–2171 (2013)