# Improving ISD Agility in Fast-Moving Software Organizations

Persson, John Stouby; Nørbjerg, Jacob; Nielsen, Peter Axel

*Published in:*
ECIS 2016 Proceedings

*Publication date:*
2016

*Document Version*
Accepted author manuscript, peer reviewed version

# IMPROVING ISD AGILITY IN FAST-MOVING SOFTWARE ORGANIZATIONS

*Research*

Persson, John Stouby, Aalborg University, Aalborg, Denmark, john@cs.aau.dk

Nørbjerg, Jacob, Copenhagen Business School, Copenhagen, Denmark, jno.itm@cbs.dk

Nielsen, Peter Axel, Aalborg University, Aalborg, Denmark, pan@cs.aau.dk

## Abstract

*Fast-moving software organizations must respond quickly to changing technological options and market trends while delivering high-quality services at competitive prices. Improving agility of information systems development (ISD) may reconcile these inherent tensions, but previous research of agility predominantly focused separately on managing either the individual project or the organization. Limited research has investigated the management that ties the agility of individual projects with the company agility characterizing fast-moving organizations. This paper reports an action research study on how to improve ISD agility in a fast-moving software organization. The study maps central problems in the ISD management to direct improvements of agility. Our following intervention addressed method improvements in defining types of ISD by customer relations and integrating the method with the task management tool used by the organization. The paper discusses how the action research contributes to our understanding of ISD agility in fast-moving software organizations with a framework for mapping and evaluating improvements of agility. The action research specifically points out that project managers need to attend to the company's agility in relating to customers, that company agility links to project agility, and that this requires light method and tool support.*

*Keywords: Agile ISD, action research, fast-moving software organizations*

# 1 Introduction

Rapid changes in markets and technologies force software organizations to make frequent changes to what they do and how they do it. This is particularly prevalent in the hyper-competitive markets introduced with the Internet boom (Lyytinen et al. 2010). Such software organizations are constantly on the move – not because they find this behaviour particularly attractive, but because their existence depends on constant adaptions to turbulent environments (Holmberg and Mathiassen 2001). The constant adaptive behaviour may also be the organizational objective, conceptualized as *strategic flexibility* (Hitt et al. 1998), *dynamic capabilities* (Elsenhardt and Martin 2000), or *response ability* (Dove 2002). In short, software organizations need to adapt.

Matching the management process to the structure of the company is a central challenge in the small and mid-sized enterprise segment (Turner et al. 2010; Turner et al. 2012) that includes most fast-moving software organizations. Agile methods appear as a solution to information systems development (ISD) managers in fast-moving software organizations. An agile ISD method implies a continual readiness "to rapidly or inherently create change, proactively or reactively embrace change and learn from change while contributing to perceived customer value (economy, quality and simplicity) through its collective components and relationships with its environment" (Conboy 2009). The literature on agile development methods emerged with few links to the literature on organizational agility that characterizes fast-moving software organizations (Mathiassen and Pries-Heje 2006). Fast-moving software organizations with a large number of small diversified projects are, furthermore, at odds with the underlying assumption of 'one team - one project' found in agile methods (Larman 2004).

Research has so far focused on how to improve agility of either the project or the company. Limited research has focused on improving agility of the ISD management that ties the agility of individual projects with the organizational agility of the company. Furthermore, it is particularly prudent to help fast-moving software organizations balance their potential for significant economic growth against the persistent risk of failure resulting from constant adaptions. Against this backdrop, we report on an action research study of ISD agility in *Adapt*, a successful and fast-moving software organization that develops e-commerce solutions based on open-source software. Through action research (Mathiassen 2002) with *Adapt*, we address the research question:

**How can ISD agility be improved in fast-moving software organizations?**

We answer the research question based on an analysis of the challenges in our client organization *Adapt* that we collaboratively addressed, with the aim to improve ISD agility. Our interventions were evaluated with practitioner assessments of how well they addressed the challenges, and theoretically, whether and how the changes contributed to ISD agility. Based on our study, we contribute empirical knowledge on (1) the importance of understanding the company's environment (2) the linkages between agility of the projects and the company; and (3) the lightness of methods and tools. The paper is structured as follows: The next section presents related research and the theoretical framing for the study. The following section summarizes our action research approach and subsequently, we present an analysis based on our improvement activities of ISD agility with *Adapt*. Three themes emerged from our analysis and we discuss how they contribute to our understanding of ISD agility in fast-moving software organizations.

# 2 Related Research

In the following, we present the theoretical foundation for explaining the fast-moving software organization and provide an overview of the research pertaining to ISD agility. The related research helped us understand, guide, and evaluate how the action research interventions contributed to ISD agility in the fast-moving software organization *Adapt*.

## 2.1 Fast-Moving Software Organizations

Holmberg and Mathiassen (2001) conceptualized the fast-moving software organization with their lessons on how to cope with a dynamic environment while simultaneously improving professional practices. They argued that from an organization's attempts to deal effectively with its environment, a culture emerges (Schein 1985) that we need to understand in order to improve their practice. The fast-moving software organization can be understood as an agile enterprise that relates more generally to flexible organizations for fast-moving markets (Volberda 1997), strategic flexibility (Hitt et al. 1998), dynamic capabilities (Elsenhardt and Martin 2000), and response ability (Dove 2002). In these organizations, agile operations are related to effectively responding to a changing environment while at the same time being productive. This concept of agility arose from flexible and lean manufacturing aiming for economy of scope rather than economy of scale (Dove 2002; Mathiassen and Pries-Heje 2006).

The software organizations in the age of the Internet that need to cope with rapid change (Baskerville et al. 2003; Pries-Heje et al. 2004) exemplify a demand for agility by a new scope of operations. The adoption of Internet technologies and cloud services by software organizations reflect a hyper-competitive environment demanding simultaneous and mutually complementary learning routines (Lyytinen et al. 2010). Baskerville et al. (2011) argue that dramatic changes in the market causes disruption of established practices, experimentation, and process adaptations followed by consolidation of lessons learnt into a new and once again relatively stable software development process. They historically situate early phenomena such as ''Internet Speed'' and ''Internet Time'' as pre-agility, which was an early form of agility that does not completely satisfy today's taxonomies (Conboy 2009), but instead helped to shape them (Baskerville et al. 2011). Today, ISD agility is widely disseminated with references to the *Agile Manifesto* (Beck et al. 2001) and development methods such as *Scrum* (Schwaber and Beedle 2002) and *Extreme Programming* (Beck 1999). The *Agile Manifesto* (Beck et al. 2001) reflected a reaction to issues in software organizations of rigid processes and tools, comprehensive documentation, contract negotiation, and following a plan. Fast-moving software organizations inevitably put much more value in individuals and interactions, working software, customer collaboration, and responding to change. These organizations may have small and diversified projects with little room for institutionalizing processes (Babb et al. 2014a; Basri and O'Connor 2010; Coleman and O'Connor 2008; Lester et al. 2010; Pedreira et al. 2007; Staples et al. 2007).

Post-agility for ISD may result from the organizational issues created by the boundaries between agile development teams and plan-driven personnel (Baskerville et al. 2011). Post-agile ISD proactively pursue the dual goal of agility and alignment through a diversity of means, for example through method components, and software tools as well as via new ways of organizing, specializing, communicating, and managing relationships. Baskerville et al. (2011) suggests a deep incorporation of agility into all modes of software development such that agile and plan-driven cease to be distinguishable.

## 2.2 ISD Agility

The body of research into ISD agility focuses on the practices of software developers within agile software development projects (Conboy 2009; Molnar and Nandhakumar 2009; Stacey and Nandhakumar 2008). Several studies compared the recommendations and techniques suggested in an agile method, such as; e.g. Scrum or XP, to the actual practices of the software developers and their managers. The studies demonstrated how agile methods are adapted to local conditions and constraints (Conboy and Fitzgerald 2010), and that many practices such as; e.g., having a customer on site, are omitted or tailored due to the customer's reluctance to commit the necessary effort or simply the absence of a identifiable customer (Conboy 2009; Hoda et al. 2011; Wang et al. 2012). Other practices such as daily meetings, retrospectives, or pair programming are also frequently omitted or adapted due to time constraints or management preferences (Babb et al. 2014b; Babb et al. 2014a; Hoda et al. 2013; Wang et al. 2012). Agile ISD will in practice mix and match elements from both agile and tradi-

tional methods (Boehm and Turner 2003; Conboy 2009). A project with fixed requirements may, for example, use Sprints and frequent releases to plan and control the project internally, while maintaining a phased and sequential process in the project's interactions with the customer. The adaptations and adoptions of agile practices with elements from traditional software development methods motivated studies into the breadth and depth of an organization's adoption of agile methods (Senapathi and Srinivasan 2012; Wang et al. 2012) as well as a debate about when local development and management practices cease to be agile (Conboy 2009; Lyytinen and Rose 2006). It is, however, problematic to assess agility of local practices through comparison with a specific method insofar different agile methods do not agree on which practices are important and may in fact contain contradictory advice (Conboy 2009). Instead, evaluation of the agility of a method, a practice, or a set of practices should be based on a clear definition of the term agility (Conboy 2009).

Based on a literature study of agility in areas other than ISD, Conboy proposes the following definition of ISD method agility emphasizing the core principles of agility of embracing change and providing customer value: *"[T]he continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change and learn from change while contributing to perceived customer value (economy, quality and simplicity) through its collective components and relationships with its environment*" (Conboy 2009 p. 340). The definition is translated into a formative taxonomy of ISD Agility (see Table 1) that outlines the goals an ISD method or a part of it must achieve to be agile. The taxonomy has three parts: the first and second parts refer to handling change and contribution to value. The third part emphasizes that an agile method component should be readily available; i.e.; not take too much time to prepare and use. A project plan, for example, is not readily available if it takes too much time to prepare and/or change it.

| | |
|---|---|
| 1. To be agile, an ISD method component *must* contribute to one or more of the following: | (i) creation of change |
| | (ii) proaction in advance of change |
| | (iii) reaction to change |
| | (iv) learning from change |
| 2. To be agile, an ISD method component *must* contribute to one or more of the following, and *must not* detract from any: | (i) perceived economy |
| | (ii) perceived quality |
| | (iii) perceived simplicity |
| 3. To be agile, an ISD method component must be continually ready, i.e., minimal time and cost to prepare the component for use. | |

*Table 1. Taxonomy of ISD agility (Conboy 2009 p. 341)*

Conboy (2009) suggests that the taxonomy can be applied to (1) test agility of commercially labelled agile practices, (2) show a practice is not agile in every instance, and (3) identify new agile practices. In this paper, we want to evaluate the results of interventions into the ISD management practices in a fast-moving software organization. We, therefore, use the taxonomy to evaluate the agility of the ISD management improvements that resulted from our action research interventions.

# 3 Research Approach

The research design was based on action research. This is an appropriate research approach when the research question addresses how professional practitioners take action and improve their action in an organization (Davison et al. 2004; Kock 2007; McKay and Marshall 2001). Our action research effort had the purpose of improving ISD agility by addressing the ISD management challenges in a fast-

moving software organization. This was agreed with the client organization as part of establishing the collaboration. The specific action research approach was Collaborative Practice Research (Mathiassen 2002) that assists in connecting the need to understand current practices with the aim to improve them. Collaborative Practice Research was supplemented with six criteria for evaluating action research: roles, documentation, control, usefulness, frameworks, and transferability (Nielsen 2007). Our approach, therefore, covers the same aspects as canonical action research (Davison et al. 2004), but it is more specific on creating useful interventions for professional practitioners to improve practice.

There were three action researchers, all with a background in software development and management, and with documented action research experiences ranging from 8 to 30 years. The action researchers went into the collaboration with an initial knowledge interest to study the company's agility and how that influenced ISD management. The company went into the collaboration with the researchers to attain an outside view on their current practices and to be assisted in overcoming some key problems they were facing. We conducted the action research over a period of six months through eight activities. These activities follow the structure laid out in (Iversen et al. 2004):

1. Establish agreement between the client organization and the actions researchers (Davison et al. 2004) and the organization of the collaboration (Mathiassen 2002).
2. Problem definition jointly with the client company (Nielsen and Persson 2010).
3. Appreciate the company's current project management practice in its organizational context using open-ended qualitative interviews (Patton 2005) and analyze the company's organizational culture (Cameron and Quinn 2011).
4. Diagnose the problems in detail and suggest actions to improve the current practice (Iversen et al. 1999).
5. Take actions to change current practice – these change actions should be iteratively organized to keep pace and direction with the agility of the company (Börjesson and Mathiassen 2005; Börjesson et al. 2006).
6. Evaluate the effects of the intervention against the understanding of the diagnosis and on the backdrop of the company's agility (step 4).
7. Take supportive action and return to step 5 if the effects are not yet satisfactory.
8. Elicit lessons learned and evaluate against the six action research criteria (Iversen et al. 2004; Nielsen 2007).

All encounters between the action researchers and the company's professional practitioners were documented through audio recordings, field notes and minutes. The minutes were sent to all participants for commenting. Following each encounter, a debriefing meeting (Spall 1998) was conducted among the action researchers. The premise, inference, and contribution of action research can be composed in different styles (Mathiassen et al. 2012). The premise style of this research is practical and not theoretical as we have investigated how practitioners in *Adapt* (a fast-moving software organization) can improve ISD agility. The inference style is inductive and not deductive as the arguments are based on data and evidence from the problem-solving where agility were worked with and then subsequently related more directly to a better understanding of the concepts from the research literature. The contribution style we seek is a field study that extends concepts in the literature about ISD agility and fast-moving software organizations.

# 4 Analysis

*Adapt* was established in 1998 and develops web-based solutions for both public and private organizations. It is a profitable company maintaining a top credit rating, and with 65 employees (as of March

2015). Their customers include Danish retail businesses of various sizes and companies for which online functionality and visibility is a central part of the business. *Adapt* has recently migrated from their own proprietary content management system to *Drupal*. The company considers itself a leader in this technology and is active in the Danish and international *Drupal* community contributing with code reviews and new modules. Thus, recruiting and keeping *Drupal* specialists are key priorities in their business strategy.

The software developers and project managers are organized into three functional groups: Project management headed by the chief project manager, back-end development headed by the chief technology officer, and front-end design and development headed by the chief design officer. Developers are divided into teams lead by a project manager, and assigned to several projects. Teams are frequently reorganized in order to balance fluctuating resource demands among projects. Developers and architects are self-organizing all operational tasks. The chief project manager divides her time between managing her own projects, resource allocation across projects, and supporting the two less experienced project managers in e.g. weekly coaching sessions. The project managers all have a background in media and communications and had worked with web design prior to coming to *Adapt*. They had between 1 to 8 years total work experience and had worked between 2 months and 5 years at *Adapt*. They had general project management experience prior to coming to *Adapt*, and two of them are certified in project management according to the International Project Management Association Competence Baseline (IPMA ICB). They described their project management training at *Adapt* as "learning by doing" under the supervision of a more experienced project manager.

There had been several recent changes to the project management group. One experienced project manager had left *Adapt* in 2013 and two new had been hired between Summer 2013 and February 2014. A newly hired project manager had only been two months at *Adapt*. Another project manager decided to leave the company and a replacement was hired in the spring of 2014. *Adapt* has also made numerous movements in response to their market. Some years ago they moved to the *Drupal* platform and prior to the action research project in February 2014 they initiated other moves: From midsized to large clients; from fixed price projects to long-term time-and-material contracts; from project-orientation to client orientation; numerous new employees – especially among project managers; and creating a subsidiary company for mobile applications.

At the start of the action research project with *Adapt*, we met a strong, technically competent, and self-managed group of developers several of whom had long company experience. The developers were organized into teams, but the team structure was loose and fluid in order to maintain high group coherence and minimal internal competition among the developers (according to the Chief Technology Officer). The organization's success and agility was founded on the competences of these developers. They had built the organization's initial technical platform, but had also – supported by new hires – been key drivers in *Adapt's* early and successful move to *Drupal*. The four project managers struggled with an increasing number of projects, customers, and frequent changes to project types and customer relations. We decided to focus the action research project on the problems of the project managers because they were clearly motivated to changes that would reduce the pressure upon them.

## 4.1 ISD Management Challenges in *Adapt*

During the problem diagnosis, we identified three major and current challenges faced by *Adapt*: (1) diverse project manager roles with different responsibilities, (2) diversified and changing project types, and (3) insufficient, dated, and redundant methods and tools for ISD management.

**Challenge 1:** The project managers were responsible for customer communication, project budget and planning, and task breakdown and completion. They would prioritize development tasks on projects and allocate them to developers on a weekly or bi-weekly basis. The task allocation became increasingly complex as the number of projects grew, and involved frequent negotiations and re-allocation of

developers between project managers. Furthermore, the chief project manager spent considerable time allocating and reallocating developers to projects. The project managers also reported that they were also expected to fill the roles of lead architect, test manager, and tester on projects. These tasks were not allocated to a specific function in *Adapt* and had, therefore, become part of the project manager's responsibilities: *"Testing is a developer responsibility but they don't do it properly. They need to learn."* (Project manager). The fluid borders between project management and development had worked well in the past, but the least experienced project managers and the chief project manager reported increasing time pressure, and delays: *"I don't like tasks that are just lying around. Some were over two years old and nothing had been done because of lack of time. ... I am beginning to do the same. Tasks are allowed to sit .. one month, two months."* (Junior project manager).

**Challenge 2:** *Adapt* was going through changes to customer relations and project types. A transition from a waterfall type process model with fixed time and budget, towards agile projects in 2013 was accompanied by a general change in customer relations from individual projects towards long-term relationships based on a Service Level Agreement with each customer. The transitions were not completed – and probably will not be for a while because of differences in customer relationships and preferences – so projects could vary across process type (agile vs. waterfall) and customer relationship (project vs. Service Level Agreement) resulting in four different types of projects of varying sizes. The project managers were struggling to understand the differences between these four types and how best to manage each. *"[The process] has been stable for several years. But the last year has been chaotic ... Every time I begin to describe the process, things change ... We grow so quickly and get so many new customers who want to work in a new way."* (Chief project manager)

**Challenge 3:** The chief project manager had revised the process descriptions in 2013, but her changes had already been made obsolete by the many changes in the company. *"This is a description of Adapt's method. It was made 6 months ago and it is already outdated "* (Chief project manager). The 13-page method description tailored to the company featured checklists and a contingency-based selection of either a traditional or an agile approach. The chief project manager was working on a new process description but it had not been completed due to lack of time. The project managers, therefore, worked with the old descriptions, templates, and checklists in their own way, and there was no common project management practice in *Adapt*. The project managers used several reporting and support tools for planning, resource allocation, time reporting, and billing, including their own personal spreadsheets. The lack of tool integration resulted in redundant data entry and poor overview of project status and resource allocation. *Adapt* was beginning to use *JIRA* for task allocation and tracking, and intended to eventually use the system to support project and resource management – including time reporting – as well. Use of the system for project management was, however, not mandated, nor was there any common guidelines: *"We were told to [use JIRA] in our own way ... How does that support the developers and the process? It is far too difficult for someone else to take over from me if I do everything my way instead of everyone doing it the same way."* (Junior project manager)

*JIRA* is an issue tracking system most commonly used for software bug tracking developed by Atlassian Corporation starting in 2002. Its advanced customization features make it suitable for other types of ticketing systems (work orders, help desks, etc.), and project management. These features make *JIRA* useful for managing even large-scale software development (Helming et al. 2009). *Adapt* used an extension for agile systems development (*JIRA Agile*) and described project tasks in terms of *Epics* and *User Stories*. A *User Story* is a requirement expressed in a few sentences and in a non-technical language (represented as an *Issue* in *JIRA*). An epic captures a large body of work that can be broken down into a number of smaller *User Stories*. The project managers used *JIRA* to manage the development tasks but not the management tasks.

## 4.2 Intervention at *Adapt*

Over the cause of the action research project, we came to consider challenges 1 and 2 as conditions for project management in *Adapt* since the market and environment for the company underwent major changes. The implication of this realization was that the project managers would have to improve their ability to adjust to changes in the company's environment by being flexible and agile. We therefore decided on a concerted effort to address challenge 3 by changing their methods and tools in such a way that they would better support the project managers in their diverse roles and in adapting quickly to changes in their environment, i.e., improve their agility.

On this background, we proposed to the project managers (1) to reformulate their method to better reflect their practice and environment, and (2) to integrate ISD planning and control into *JIRA*. As a particular thinking behind adapting to the environment, we suggested to define few and clearly distinct types of projects in order to better differentiate their processes to the needs of different customers. The intention was to improve agility by making their ISD management processes more externally oriented while still maintaining a contingency-based approach with a limited scope suitable for small and midsized enterprises. At the same time, they should increase brevity of the method presentation to cater for future changes to their methods in an agile manner. The integration of ISD planning and control into a tool, in this case *JIRA*, was proposed to consolidate understanding of their activities on a single platform. We specifically proposed they develop templates for ISD management activities for the different project types, and to integrate management tasks with the development tasks. The purpose was to improve agility by making ISD management processes more flexible and maintainable while aligning them with an existing IT infrastructure in the company. The rationality behind these suggestions can all be traced back to how we understand a fast-moving software organization and what agility is in a software company, cf. section 2.

In the action research project, the chief project manager was key in realizing the proposed changes. She stayed committed despite a tremendous growth of customers and employees, which put a very high workload on her shoulders during the intervention. To establish commitment to the changes, we (the action researchers) made a substantial effort in documenting and explaining the challenges in *Adapt* – not only to the chief project manager but also to all the project managers and to the company at large. In a seminar held with all project managers and the CEO, we illustrated how they could improve their 'planning of planning' by better distinguishing between customer relations (in types of ISD) and agile versus traditional ISD management in the method. Based on an analysis of features in *JIRA* we explained in detail, how to represent project types as templates with preloaded issues corresponding to the essential ISD management tasks. Each task would then contain checklists in the form of sub-tasks.

To initiate the improvement effort, the chief project manager rewrote their ISD management method based on our initial feedback and discussions with her. This new method description was then reviewed and feedback was provided in two iterations. The resulting method description was reduced from 13 to 6 pages. The generic ISD management model with an agile and a traditional variant was changed to three distinct models called *Project* (2 pages), *Service Level Agreement* (1 page), and *Supporter* (1 page). Each model contained a visualization of the process, descriptions of key activities, and a checklist. The remaining 2 pages were respectively an overview of the three approaches and definitions of the 9 roles as *customer representative, project manager, lead (developer), developer, information architect, designer, hosting, quality assurance,* and *coordinator*. The chief project manager also made changes to their use of *JIRA* concurrent with revising the ISD management method. She made specialized use for each of the three ISD types but had not implemented (only planned) templates and checklists at the time we made the final evaluation of the interventions. However, she started using *JIRA* for managing development resources in conjunction with the tasks, thus abandoning their previous resource management systems. Two out of the four teams made the transition to *JIRA* while the remaining two were planned to make the transition in the weeks following the evaluation.

## 4.3 Improving ISD agility

We evaluated the resulting changes at *Adapt* in two stages: In the first stage, we asked the practitioners at *Adapt* to assess how well the changes addressed the challenges they were facing. In the second stage, we used the taxonomy of ISD Agility (cf. Table 1) to analyze whether and how the changes contributed to agility in *Adapt*. We summarize our changes as: (1) *define ISD management by customer relations* and (2) *integrate ISD management with a tool (JIRA).*

**Evaluation 1:** The evaluation took place in an interview with the chief project manager and a seminar with the project managers and the CEO in June 2014. The chief project manager reflects on their preceding problems in relation to (1) *define ISD management by customer relations*. She points to the increasing technical competencies among their customers resulting in them taking larger management responsibility, which allows ISD to be more iterative and incremental: *"We have been running our projects like we only had 15 people in the company and we are close to 60 … Now we are making teams much more independent – plus we are establishing a shared method. We had a shared method in the past but it was divided in phases and was a regular waterfall. Now our customers have a completely different approach and they are much more competent in running projects. We are facing technical project managers – it is not just somebody from the marketing department running an IT project."* (Chief project manager)

The chief project manager further reflects on the second method improvement of integrating with *JIRA*. She emphasizes the importance of incrementally developing both their ISD management method and tool because of their fast-moving software company: *"We need to develop both our method and its supporting tools. I have mistakenly tried to run with a method and then put in the tools. The problem is that we barely finish before there is a new direction. Now we are doing it leaner with a little bit of method concurrently with a little bit tool and so forth. That has proven to run better the preceding 6 months."* (Chief project manager)

The constantly changing organization made it difficult for project managers to communicate their processes among each other and to the developers in their teams. The scalability of method is thus important to a senior project manager because it can bring stability: *"A lot of things are changing and we need to know what to communicate. I think we now have a method that is scalable enough to give us stability even though we continue the growth."* (Senior project manager)

The CEO also expressed satisfaction with the two method improvements in defining ISD management by customer relations and integrating ISD management with *JIRA*. In particular, he points to their changes in project management tool: *"We are now in a position with the right organization and the right tool – and we are about to have the right processes. We didn't have this earlier: we didn't have the right organizational structure; we didn't have the right tools; and thereby also some scruffy processes. The project managers ran their individual tracks with great frustration among many – especially the project participants. This also explains why we had: our own developed [tool]; Rally; Wrike; JIRA in three years … We have the right structure now."* (CEO)

**Evaluation 2:** While the practitioners at *Adapt* are content with the problem solving, our research concern of ISD agility is not evident in their statements. We thus used the lessons from *Adapt* to evaluate the effect on ISD agility according to Conboy's (2009) taxonomy (cf. Table 1). To be agile, an ISD method component *must* (1) contribute to one or more ways of handling change, (2) contribute to economy, quality, or simplicity without detracting from any, and (3) be continually ready for use. The results, summarized in Table 2, show that the changes contributed to agility in all three categories of the taxonomy: (1) handling change, (2) creating perceived value, and (3) continual readiness. The first and second categories require a contribution to at least one subcategory, leaving some empty fields in Table 2 with no identified contribution to agility. Our analysis also shows that the changes contributed to agility of both the projects and the company.

| | | **(1) Define ISD management by customer relations** | | **(2) Integrate ISD management with a tool (*JIRA*)** | |
|---|---|---|---|---|---|
| **Taxonomy of Agility** (Conboy 2009) | | Project | Company | Project | Company |
| 1. To be agile, an ISD method component *must* contribute to one or more of the following: | (i) creation of change | | | | |
| | (ii) proaction in advance of change | | | Shared overview of all the tasks and used resources helps act in advance of change. | |
| | (iii) reaction to change | Brief method descriptions contingent upon the overall task, which assumes and allows adjustments with the three ISD types as a shared staring point. | Brief method description makes itself easy to change, i.e. adding a new checkpoint or a new type of customer relationship. | Customers feeding tasks into *JIRA* and tracking their progress allows faster discovery and reaction to needs for change. | Checklists and templates are easier to modify (communicating change) for future projects compared to the previous underused method. |
| | (iv) learning from change | | A shared starting point in the three types helps conceptualize and discuss specific changes among the project managers and to others in the organization. | | The checklist and templates may support shared codifications of lessons to supplement their predominantly informal learning. |
| 2. To be agile, an ISD method component *must* contribute to one or more of the following, and *must not* detract from any: | (i) perceived economy | | | Time savings in setting up the ISD management environment and in having the developer team and the customers specify, manage, and coordinate tasks. | Efficient communication and integration of method changes within the ISD types and improved overview of developer resources. |
| | (ii) perceived quality | | | | |
| | (iii) perceived simplicity | A method scope that is lowered to their most basic practical needs specific to the different overall ISD tasks. | The method is tied to the external task rather than an internal focus on an agile versus traditional process. It also depends on and exploits the existing knowledge socialization. | | |
| 3. To be agile, an ISD method component must be continually ready i.e. minimal time and cost to prepare the component for use. | | The reduced method scope is easier to follow and legitimize additional questions to the experienced project managers. | The customer orientation and reduced scope ease communicating the method to existing and new project managers. | Setting up the ISD management tool requires less effort and helps the project manager remembering and tracking her own tasks. | Setting templates and checklists, *JIRA* requires some effort but its maintenance is simpler. |

*Table 2 Evaluation of the changes' contribution to ISD Agility*

In Table 2, *Define ISD management by customer relations*, for example, contributes to simplicity in the management of projects by providing a concise ISD management framework. The change also contributes to simplicity for the company based on the specific needs of *Adapt* and their customers, and by providing a common point of reference for the project managers.

Both changes contribute to the ability to react to change in both the projects and the company. A short and simple checklist of management tasks eases adjustments as conditions for changes to the ISD type. The short and simple descriptions are also easy to adjust to changes in *Adapt's* environment, e.g., a

new type of customer relationship. Integrating ISD management into a tool, in this case *JIRA*, similarly enables customers and project managers to discover and react to changes, but it also eases the documentation – and use – of changes to checklists and templates. Both changes, furthermore, contribute to learning from change in the company. The reactions to change in the projects are shaped by *JIRA*, but the explicit ties to the three types of customer relations help the project managers conceptualize and discuss specific changes among each other in general.

Both changes have linkages between the project agility and the company agility in terms of coherence and concurrence. The coherence is the connection and consistency between how a change contributes to agility across project and company. For example, *integrate ISD management with a tool (JIRA)*, contributes to perceived economy by saving time in both the projects and the company (cf. Table 2). The contributions are similar (yet not identical) and connected by being mutually enforcing and dependent on each other. We see such coherence in how the templates produced by the chief project manager reduces the project managers' effort when setting up new ISD projects, but thereby she also eases her maintenance of the method. The concurrence is the linkages' temporal dependency. For example, the contributions to perceived economy are emerging at a similar pace for both the projects and the organization. In this case, both the chief project manager and the project managers quickly experience time savings that can increase with the continued commitment of both parties.

# 5 Discussion

Our initial diagnosis of the challenges in *Adapt* showed that their existing ISD management approach did not sufficiently incorporate the fast-moving nature of the company. Time pressure, different and changing project types, and divergent practices inhibited the ability of the project managers to handle change in an agile manner. To improve this, we worked with them to develop a clear and simple distinction between types of ISD projects, and brief but comprehensive descriptions of how to manage each type. The descriptions were supported by a tool (in this case *JIRA*) in the form of checklists for the ISD management tasks. The particular way we have used the concepts, dimensions, and theory underpinning Table 2 has showed how useful the resulting framework is. The resulting framework is summarised in Figure 1.

| Agility | | Define mgt. | | Tool support | |
|---|---|---|---|---|---|
| | | *Proj.* | *Comp.* | *Proj.* | *Comp.* |
| 1. Contribute to: | (i) Create change | | | | |
| | (ii) Proactive | | | | |
| | (iii) React to change | | | | |
| | (iv) Learn f rom change | | | | |
| 2. Contribute to and not detract from: | (i) Economy | | | | |
| | (ii) Quality | | | | |
| | (iii) Simplicity | | | | |
| 3. Continually ready: | | | | | |

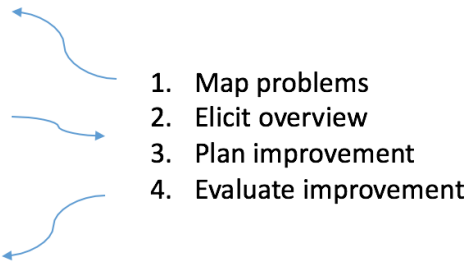


*Figure 1: The resulting framework*

The resulting framework on the left in Figure 1 stems partly from (Conboy 2009) and it is also extended with and generalised to the four columns on 'Define ISD management by relationship with customers' and 'Tool support'. The four steps to the right in Figure 1 generalises how we have used the framework. The framework is fundamentally a construct for mapping the specifics of a company's

agile practices and for utilising this mapping in observing problematic areas, eliciting an overview of the situation in the company, and for planning what will improve the situation. It is also a construct for evaluating to what degree changes have improved agility.

The main contribution of this action research can be found in the usefulness of the framework in Table 2 now generalised into Figure 1. The framework contains the theory of agility found in (Conboy 2009); but it also extends this by establishing the two other dimensions of the framework, i.e., (1) the distinction between defining ISD types based on customer relations and integrating the method with a tool; and (2) the distinction between the projects and the company. While the resulting framework reached its final format in Table 2 during the evaluation, we used all three dimensions as action researchers during all phases of the action research. The three dimensions were useful both separately, as well as when combined. They informed us as action researchers and consequently also the practitioners in *Adapt* about how to understand the current situation in *Adapt* and how to proceed with improvement from this mapping and overview. The framework also informed the evaluation of the improvement effort. The framework presented in Table 2 was useful for understanding, action planning, and evaluation of ISD management in *Adapt* as a fast-moving software organization. Based on this, we suggest that it may be just as useful for other fast-moving software organization wishing to improve ISD agility when we generalise it by removing what is specific to Adapt in Figure 1. In the following, we discuss three specific contributions of the action research using the framework (cf. Figure 1).

**Company environment:** The framework explicitly addresses how a fast-moving software company must relate to its environment; In our case how *Adapt* needs to be agile in how it relates to its different customers and how the company in differing ways must attend to customer needs. For *Adapt* this was achieved by defining ISD management by type of customer relation. We worked with *Adapt's* project managers to develop clear and simple distinctions between ISD types and guidelines on how to manage each type. As a result, they abandoned their previous distinction between traditional vs. agile projects. The initial aim was to improve agility by making their ISD management processes more externally oriented (Cameron and Quinn 2011) while still maintaining a contingency-based approach (Conboy and Fitzgerald 2010). The external orientation should help the project managers handle the company's constant adaptions to turbulent environments (Holmberg and Mathiassen 2001) and sharpen their aim for economy of scope rather than economy of scale (Dove 2002; Mathiassen and Pries-Heje 2006). It suggests that company agility as it is described by previous research, and more specifically how a particular company's environment influences the company, has to be understood by ISD managers. How project managers understand the economy of scope of their company influences ISD and they cannot limit their own scope of attention to merely understanding the goals and conditions of the projects they are managing.

This also relates to the discussion of *post agility* for ISD and specifically on how it may be accomplished (Baskerville et al. 2011). The framework highlights some practical knowledge on how the dual goal of on the one hand agility and on the other hand alignment (of planning, people, and tasks) can be achieved within relationships in ISD. In our study, the ISD relationships are between developers, project managers, and a fast-moving organization. Specifically we demonstrate an incorporation of agility into the project management such that agile and plan-driven ceases to be distinguishable (Baskerville et al. 2011). The choice between agile and plan-driven methods (Boehm and Turner 2003) was replaced with an increased focus on the external customer relations at *Adapt*. This change was based on an understanding of the customers, which *Adapt* currently have or want to have, rather than a theoretical and paradigmatic distinction between agile and plan-driven methods; and they are thus specific to *Adapt*. This change of focus, we suggest, is a particular way of detailing what (Baskerville et al. 2011) calls alignment by improving the agility of both the projects and the company.

**Linkages between the agility of the projects and the company:** The linkages between the project agility and the company agility are central to explaining the contribution of our changes using the framework. There were linkages between three parts of the agility dimension in Table 2; i.e.; (1) han-

dling change, (2) creating perceived value, and (3) continual readiness. The finding in this action research study suggests that for ISD management to be effective in a fast-moving software organization they must improve the agility of both the projects and the company.

This finding relates to the taxonomy of ISD agility in (Conboy 2009). The focus in (Conboy 2009) is on the effect of ISD method components or improvements on the agility of individual projects. However, in a fast-moving software organization like *Adapt* the agility of the company is of equal importance. While Conboy (2009) argues that the effect of an ISD method on agility is sensitive to the project context, the taxonomy does not explain the linkages between the agility of the individual project and the agility of the company. In fast-moving software organizations, the agility of the company is both an essential characteristic and a particular challenge (Holmberg and Mathiassen 2001). Thus, our action research in *Adapt* extends Conboy's (2009) taxonomy by showing that the linkages between agility of the company and the projects are fundamental to ISD agility. Thus, with this finding we address the call for a more holistic view on agility in IS research (Salmela et al. 2015). The implication of this is that ISD managers in fast-moving software companies must *simultaneously* strive to improve agility of the projects and of the company.

**Lightness of methods and tools:** The third theme emerged from working with *Adapt* on improving ISD agility by changing their tool support to better the efficiency and efficacy. The finding is that it was useful for *Adapt* to develop brief but comprehensive descriptions of how to manage each type of ISD that were then included in *JIRA* in the form of checklists for ISD management tasks. This improvement draws on previous research showing that a risk checklist helps software practitioners identify more risks than they would identify without the aid of a checklist (Keil et al. 2008). We transferred this finding of how checklists influence practitioners' perception and decision making to the general and essential project management tasks in *Adapt*. Furthermore, we reduced the already brief method description from 13 to 6 pages.

Our study contributes in this way to research on configuring ISD methods and tools for a dynamic context. The use of contextual factors to configure the method or process agility for ISD has received substantial research attention (Boehm and Turner 2003; Karlsson and Ågerfalk 2009; Kruchten 2013). However, little research has focused on the agility of managing the configuration activity itself. The combination of agile projects with a stable or slow-moving organization eliminates the need for management agility across both the projects and the company. In fast-moving software organizations, management agility of method and process configurations is a central challenge because their existence depends on constant adaptions to turbulent environments (Holmberg and Mathiassen 2001). Based on this action research study, we suggest that ISD management tools may play a substantial role in the frequent adjustments that are required to keep up with a dynamic environment. Our analysis also supports the principles of agile project management of minimal critical specification, autonomous teams, redundancy and feedback and learning (Dybå et al. 2014). However, we also add the importance of tool support for structured feedback and learning that is not only 'integral to the project's execution and the project's interaction with the environment' (Dybå et al. 2014 p. 293), but is integral to the fast-moving software company.

# 6 Conclusion

This paper reported a collaborative action research study of how ISD agility can be improved in fast-moving software organizations. We worked with the fast-moving software organization *Adapt* on two distinct changes intended to improve agility. The first was to define ISD management by customer relations. The second was to integrate ISD management with *JIRA*. The practitioners at *Adapt* assessed that these changes positively affected the challenges they were facing. Second, we used Conboy's (2009) taxonomy to analyze whether and how the changes contributed to ISD agility. We discuss the contribution of the changes to agility in three general themes for improving ISD agility in fast-moving

software organizations: (1) the importance of understanding the company's environment (2) the linkages between agility of the projects and the company; and (3) the lightness of methods and tools.

An important limitation of our study is the changing nature of fast-moving software organizations. The frequent moves make it very difficult to establish a current understanding of the organization and evaluate the effects of changes. Every time we met with *Adapt* in intervals of a few weeks they had made new changes as part of the company's general problem solving. Thus, organizational moves quickly entangled with the problem solving of the action research project. Interventions in fast-moving software organizations cannot be discretely isolated and evaluated. Our action research ended after six months when we were able to evaluate the changes with the involved practitioners and Conboy's (2009) taxonomy of ISD agility. While the changes may feed-forward or be adapted yet again to changing circumstances, this concern is outside the scope of our action research project.

The potential of fast-moving software organizations for economic growth, but also the persistent risk of failure from the constant adaptions, underline a need for future research. In these organizations, management of ISD agility appears to be an important element of success. However, we still need to explore different approaches to improving agility across project and company. Specifically, we call for exploration and evaluation of the coherence and concurrence between the agility of the individual projects and the company. The action research reported in this paper shows that agility is a useful measure of success when addressing ISD management challenges in fast-moving software organizations. However, we need more and varied studies to further our understanding of ISD agility improvements.

## References

Babb, J. S., R. Hoda, and J. Nørbjerg. 2014a. "XP in a Small Software Development Business: Adapting to Local Constraints," *5th Scandinavian Conference on Information Systems (SCIS)* pp. 14-29.

Babb, J., R. Hoda, and J. Norbjerg. 2014b, "Embedding Reflection and Learning into Agile Software Development," *IEEE Software* (31:4), pp. 51-57.

Baskerville, R., J. Pries-Heje, and S. Madsen. 2011, "Post-Agility: What Follows a Decade of Agility?" *Information and Software Technology* (53:5), pp. 543-555.

Baskerville, R., B. Ramesh, L. Levine, J. Pries-Heje, and S. Slaughter. 2003, "Is Internet-Speed Software Development Different?" *IEEE Software* (20:6), pp. 70-77.

Basri, S., and R. V. O'Connor. 2010. "Understanding the Perception of very Small Software Companies Towards the Adoption of Process Standards," *17th EuroSPI Conference* pp. 153-164.

Beck, K. 1999, "Embracing Change with Extreme Programming," *Computer* (32:10), pp. 70-77.

Beck, K., M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, and R. Jeffries. 2001, "Manifesto for Agile Software Development," *Http://Agilemanifesto.Org/*.

Boehm, B., and R. Turner. 2003, "Using Risk to Balance Agile and Plan-Driven Methods," *Computer* (36:6), pp. 57-66.

Börjesson, A., F. Martinsson, and M. Timmerås. 2006, "Agile Improvement Practices in Software Organizations," *European Journal of Information Systems* (15:2), pp. 169-182.

Börjesson, A., and L. Mathiassen. 2005, "Improving Software Organizations: Agility Challenges and Implications," *Information Technology & People* (18:4), pp. 359-382.

Cameron, K. S., and R. E. Quinn. 2011. *Diagnosing and Changing Organizational Culture: Based on the Competing Values Framework*, John Wiley & Sons, San Francisco, CA.

Coleman, G., and R. O'Connor. 2008, "Investigating Software Process in Practice: A Grounded Theory Perspective," *Journal of Systems and Software* (81:5), pp. 772-784.

Conboy, K. 2009, "Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development," *Information Systems Research* (20:3), pp. 329-354.

Conboy, K., and B. Fitzgerald. 2010, "Method and Developer Characteristics for Effective Agile Method Tailoring: A Study of XP Expert Opinion," *ACM Transactions on Software Engineering and Methodology* (20:1), pp. 1-30.

Davison, R., M. G. Martinsons, and N. Kock. 2004, "Principles of Canonical Action Research," *Information Systems Journal* (14:1), pp. 65-86.

Dove, R. 2002. *Response Ability: The Language, Structure, and Culture of the Agile Enterprise*, John Wiley & Sons, New York, New York.

Dybå, T., T. Dingsøyr, and N. B. Moe. 2014. *"Agile Project Management," Software Project Management in a Changing World*, Günther Ruhe and Claes Wohlin (eds.), Springer, Berlin Heidelberg, pp. 277-300.

Elsenhardt, K. M., and J. A. Martin. 2000, "Dynamic Capabilities: What are They," *Strategic Management Journal* (21:1), pp. 1105-1121.

Helming, J., M. Koegel, and H. Naughton. 2009. "Towards Traceability from Project Management to System Models," *Proceedings of the 2009 ICSE Workshop on Traceability in Emerging Forms of Software Engineering* pp. 11-15.

Hitt, M. A., B. W. Keats, and S. M. DeMarie. 1998, "Navigating in the New Competitive Landscape: Building Strategic Flexibility and Competitive Advantage in the 21st Century," *The Academy of Management Executive* (12:4), pp. 22-42.

Hoda, R., J. Babb, and J. Norbjerg. 2013, "Toward Learning Teams," *IEEE Software* (30:4), pp. 95-98.

Hoda, R., J. Noble, and S. Marshall. 2011, "The Impact of Inadequate Customer Collaboration on Self-Organizing Agile Teams," *Information and Software Technology* (53:5), pp. 521-534.

Holmberg, L., and L. Mathiassen. 2001, "Survival Patterns in Fast-Moving Software Organizations," *IEEE Software* (18:6), pp. 51-55.

Iversen, J. H., L. Mathiassen, and P. A. Nielsen. 2004, "Managing Risk in Software Process Improvement: An Action Research Approach," *MIS Quarterly* (28:3), pp. 395-433.

Iversen, J., P. A. Nielsen, and J. Norbjerg. 1999, "Situated Assessment of Problems in Software Development," *ACM SIGMIS Database* (30:2), pp. 66-81.

Karlsson, F., and P. Ågerfalk. 2009, "Exploring Agile Values in Method Configuration," *European Journal of Information Systems* (18:4), pp. 300-316.

Keil, M., L. Li, L. Mathiassen, and G. Zheng. 2008, "The Influence of Checklists and Roles on Software Practitioner Risk Perception and Decision-Making," *Journal of Systems and Software* (81:6), pp. 908-919.

Kock, N. 2007. *Information Systems Action Research: An Applied View of Emerging Concepts and Methods*, Springer Science & Business Media.

Kruchten, P. 2013, "Contextualizing Agile Software Development," *Journal of Software: Evolution and Process* (25:4), pp. 351-361.

Larman, C. 2004. *Agile and Iterative Development: A Manager's Guide*, Addison-Wesley Professional.

Lester, N., F. G. Wilkie, D. McFall, and M. Ware. 2010, "Investigating the Role of CMMI with Expanding Company Size for Small-to Medium-sized Enterprises," *Journal of Software Maintenance and Evolution: Research and Practice* (22:1), pp. 17-31.

Lyytinen, K., and G. M. Rose. 2006, "Information System Development Agility as Organizational Learning," *European Journal of Information Systems* (15:2), pp. 183-199.

Lyytinen, K., G. Rose, and Y. Yoo. 2010, "Learning Routines and Disruptive Technological Change: Hyper-Learning in Seven Software Development Organizations during Internet Adoption," *Information Technology & People* (23:2), pp. 165-192.

Mathiassen, L. 2002, "Collaborative Practice Research," *Information Technology & People* (15:4), pp. 321-345.

Mathiassen, L., M. Chiasson, and M. Germonprez. 2012, "Style Composition in Action Research Publication." *MIS Quarterly* (36:2), pp. 347-363.

Mathiassen, L., and J. Pries-Heje. 2006, "Business Agility and Diffusion of Information Technology," *European Journal of Information Systems* (15:2), pp. 116-119.

McKay, J., and P. Marshall. 2001, "The Dual Imperatives of Action Research," *Information Technology & People* (14:1), pp. 46-59.

Molnar, W., and J. Nandhakumar. 2009. "Managing Projects in an Embedded System Development Context: An in-Depth Case Study from an Improvisational Perspective," *42nd Hawaii International Conference on System Sciences (HICSS)* pp. 1-10.

Nielsen, P. A. 2007. *"IS Action Research and its Criteria," Information Systems Action Research–An Applied View of Emerging Concepts and Methods,* N. Kock (ed.), Springer, pp. 355-375.

Nielsen, P. A., and J. S. Persson. 2010. "IT Management in Local Government: Engaged Problem Formulation," *Proceedings of the 31th International Conference on Information Systems (ICIS)* pp. 1-12.

Patton, M. Q. 2005. *Qualitative Research*, Wiley Online Library.

Pedreira, O., M. Piattini, M. R. Luaces, and N. R. Brisaboa. 2007, "A Systematic Review of Software Process Tailoring," *ACM SIGSOFT Software Engineering Notes* (32:3), pp. 1-6.

Pries-Heje, J., R. L. Baskerville, L. Levine, and B. Ramesh. 2004, "The High Speed Balancing Game: How Software Companies Cope with Internet Speed," *Scandinavian Journal of Information Systems* (16:1), pp. 11-54.

Salmela, H., T. Tapanainen, A. Baiyere, M. Hallanoro, and R. Galliers. 2015. "IS Agility Research: An Assessment and Future Directions," *The Twenty-Third European Conference on Information Systems (ECIS)* pp. 1-16.

Schein, E. H. 1985. *Organisational Culture and Leadership: A Dynamic View*, Jossey-Bass, San Francisco, CA.

Schwaber, K., and M. Beedle. 2002. *Agile Software Development with Scrum*, Prentice Hall, Upper Saddle River, NJ.

Senapathi, M., and A. Srinivasan. 2012, "Understanding Post-Adoptive Agile Usage: An Exploratory Cross-Case Analysis," *Journal of Systems and Software* (85:6), pp. 1255-1268.

Spall, S. 1998, "Peer Debriefing in Qualitative Research: Emerging Operational Models," *Qualitative Inquiry* (4:2), pp. 280-292.

Stacey, P., and J. Nandhakumar. 2008, "Opening Up to Agile Games Development," *Communications of the ACM* (51:12), pp. 143-146.

Staples, M., M. Niazi, R. Jeffery, A. Abrahams, P. Byatt, and R. Murphy. 2007, "An Exploratory Study of Why Organizations do Not Adopt CMMI," *Journal of Systems and Software* (80:6), pp. 883-895.

Turner, R., A. Ledwith, and J. Kelly. 2010, "Project Management in Small to Medium-Sized Enterprises: Matching Processes to the Nature of the Firm," *International Journal of Project Management* (28:8), pp. 744-755.

———. 2012, "Project Management in Small to Medium-Sized Enterprises: Tailoring the Practices to the Size of Company," *Management Decision* (50:5), pp. 942-957.

Volberda, H. W. 1997, "Building Flexible Organizations for Fast-Moving Markets," *Long Range Planning* (30:2), pp. 169-148.

Wang, X., K. Conboy, and M. Pikkarainen. 2012, "Assimilation of Agile Practices in Use," *Information Systems Journal* (22:6), pp. 435-455.