Presented to the Interdisciplinary Studies Program:

UNIVERSITY OF OREGON
APPLIED INFORMATION MANAGEMENT

Applied Information Management
and the Graduate School of the
University of Oregon
in partial fulfillment of the
requirement for the degree of
Master of Science

Implementing Agile Methodology: Challenges and Best Practices

CAPSTONE REPORT

Jeffrey Verret
Sr. Systems Administrator III
Bonneville Power Administration

University of Oregon
Applied Information
Management
Program

Spring 2018

Continuing and Professional
Education
1277 University of Oregon
Eugene, OR 97403-1277
(800) 824-2714

Approved by

_____

Dr. Kara McFall
Director, AIM Program

Implementing Agile Methodology: Challenges and Best Practices

Jeffrey Verret

Bonneville Power Administration

Abstract

Information Technology (IT) projects have a reputation of not delivering business requirements. Historical challenges like meeting cost, quality, and timeline targets remain despite the extensive experience most organizations have managing projects of all sizes. The profession continues to have high profile failures that make headlines, such as the recent healthcare.gov initiative. This research provides literary sources on agile methodology that can be used to help improve project processes and outcomes.

Keywords: agile, deployment, challenges, methodologies, best practices

Table of Contents

List of Figures

Introduction to the Annotated Bibliography

Problem Description

It is now more than 17 years since the authors of the Agile Manifesto (Beck et al., 2001) first declared and wrote down the principles that would become the de facto standards of agile development methodologies, and more than 15 years since the agile movement started to have a significant impact on the software development industry (Laanti, Solo, & Abrahamsson, 2011). Panditi (2018) noted in a 2018 software study that "Most organizations reported that they are embracing agile within the software development area" (p. 1). The results of the study specifically indicate that eight out of ten organizations had committed to adopting agile software development practices, 55 percent were in the midst of adoption, and 25 percent had already put agile into practice (Panditi, 2018).

The seventeen original authors and signatories of the Manifesto for Agile Software Development (Beck et al., 2001) were the first to codify and write the following tenets for the methodology:

- *Individuals and iterations* [emphasis added] over processes and tools.
- *Working software* [emphasis added] over comprehensive documentation.
- *Customer collaboration* [emphasis added] over contract negotiation.
- *Responding to change* [emphasis added] over following a plan.

While recognizing the value of the non-italicized items on the right, the authors of the Manifesto for Agile Software Development (Beck et al., 2001) consider the italicized items on the left to be of greater value to the successful and timely delivery of Information Technology (IT) projects. These four tenets support the following principles of agile software development (Beck et al., 2001):

- The highest priority is customer satisfaction through early and continuous delivery of valuable software.
- Changing requirements, even late in the development process, are welcome and can lead to competitive advantage.
- Working software will be delivered frequently and, where possible, within a short timeframe.
- Business groups and developer groups must work together daily throughout the project lifecycle.
- Motivated individuals are at every project's core; they must be provided with the environment, support, and trust needed to succeed.
- Face-to-face communication is the most effective means of communication to and within the team.
- The primary measure of progress is the delivery of working software.
- Agile processes lead to sustainable development. Developers, users and other stakeholders involved in development should be able to maintain a constant pace indefinitely.
- Agile processes are enhanced by good design and aided by sustained attention to technical excellence.
- Simplicity, defined as "the art of maximizing the amount of work not done" (para. 11), is necessary.
- Self-organizing teams can be expected to deliver the best architectures, requirements, and designs.

- Teams will reflect on how to become more effective at regular intervals and adjust behavior accordingly.

These tenants and principals were first developed in response to pressures experienced by IT teams when engaged in projects for organizations which were themselves under pressure to perform and rapidly adjust to changing conditions (Highsmith, 2001). An initial reading of the above principles may incorrectly indicate a preference for a lack of organizational structure; instead, the principles do provide structure but also result in a lack of organized guidance (Maruping, Venkatech, & Agarwal, 2009). Highsmith (2001) notes that the original creators of the agile movement address the dichotomy by stating that the agile movement is not anti-methodology:

> In fact, many of us want to restore credibility to the word methodology. We want to restore a balance. We embrace modeling, but not in order to file some diagram in a dusty corporate repository. We embrace documentation, but not hundreds of pages of never-maintained and rarely-used tomes. We plan, but recognize the limits of planning in a turbulent environment. (para. 9)

From a practical perspective, the above principles translate into software project practices that employ high user engagement to reach consensus and build decisions (Ramesh, Cao & Baskerville, 2010). In addition, agile methodologies embrace an iterative approach to development as opposed to a waterfall model that uses a relatively linear and sequential approach (Sureshchandra & Shrinivasavadhani, 2008). Several of the agile principles, such as emphasizing face-to-face communication, evaluating effectiveness, and engaging stakeholders, support the practice of daily standup meetings (Laanti et al., 2011). The continuous monitoring of

deliverables to meet the agreed upon requirements means that new requirements can be introduced at any time in the development process (Ramesh, Cao & Baskerville, 2010).

Laanti et al. (2011) conducted a large-scale examination of agile practices at Nokia; their findings revealed that respondents agreed with the benefits of agile usage, including higher satisfaction and effectiveness, increased quality and transparency, and earlier deletion of faults. Responses to the questionnaire further revealed that 60% of respondents would prefer to stay with the agile methods rather than return to their previous ways of working. Other noted benefits of using agile methodologies include high visibility of project details, increased team efficiency, greater ability to adapt to change, and greater ability to scale (Vijayasarathy & Turk, 2010). Agile methods place the emphasis on business ownership of products and then prioritize IT team efforts based on what is of the greatest benefit to the business (Serrador & Pinto, 2014). This approach aims to enhance shared understanding amongst all stakeholders and teams of the project goals (Tessem, 2014).

Agile approaches also offer other benefits to the IT teams that employ these methodologies (Chow & Cao, 2008). These benefits include making IT teams aware that they are expected to become self-sufficient and take corrective actions when needed rather than waiting for instructions to be mandated from above (Chow & Cao, 2008). Project managers employing agile methods coach the IT teams for ways to improve, identify challenges, and address other areas such as wider stakeholder engagement in order to minimize distractions to the IT team and better allow them to focus purely on delivery (Maruping, Venkatesh & Agarwal, 2009).

While the benefits of agile have been well documented, adoption of agile methods can prove to be challenging (Svensson & Host, 2012). Schwaber, Laganza and D'Silva (2007) reported that authors of research on the adoption of agile methods have noted that agile adopters

are often unaware of what agile adoption really means and implementers are often unaware of how broad of a change is actually required. Recent research into common challenges organizations may encounter when attempting to introduce agile systems into a non-agile environment have revealed several cautionary areas (Gregory, Barroca, Sharp, Deshpande & Taylor, 2016). Issues include a reluctance to make the necessary project processes changes that adoption of agile methodologies requires, misunderstanding by management and others within the organization of the term *Agile* and its associated processes, challenges with team communication, insufficient trust between management and IT teams, and disagreements when prioritizing requirements (Gregory et al., 2016). Additionally, Gregory et al. (2016) note that there are established challenges scaling Agile for use in large organizations.

Other challenges with agile adoption include limited customer availability, minimal application and project documentation, incorrect budget and time estimates resulting from change requests, contractual limitations which do not allow for unspecified project adjustments, and disagreements on functional requirements (Inayat, Salim, Marczak, Daneva, & Shamshirband, 2015) Agile project challenges specific to management have included the inability of managers to stop micromanaging and even the use of the term *Agile* as an excuse to push teams harder (Gregory et al., 2016). If left unaddressed, these challenges, questions, and uncertainties can lead to a far lower level of adoption of agile methods than might otherwise occur within an organization (Hohl, Münch, Schneider, & Stupperich, 2016).

Purpose Statement

The purpose of this research is to systematically review scholarly sources to describe, demonstrate, guide, and analyze the following topics: (a) the efficacy of agile processes as they relate to project success, (b) how agile methodologies applied specifically to Information

Technology projects may be better understood and successfully deployed throughout an organization, and (c) how agile methods may be maintained to achieve greater project stakeholder engagement and higher levels of project success.

Research Questions

Primary question: What are the established best practices for enabling an organization to successfully employ agile methods?

Sub-question: How might these best practices be leveraged within organizations with entrenched workforces that may be resistant to change?

Audience Profile

The primary intended audience for this annotated bibliography includes project managers, program managers, Project Management Office (PMO) leaders, and team managers. Some of the responsibilities of project and program managers and all others who operate out of the PMO will be to (a) develop their organizations' agile knowledge bases, (b) ensure project reporting is fully consistent across all projects, and (c) facilitate project communication and understanding amongst all project stakeholders (Gregory et al., 2016). Given the expected role of those within the PMO to facilitate and manage IT projects and programs across the whole of the organization, their familiarity with agile processes will be necessary to ensure a repeatable process. Additionally, members of the PMO staff are expected to interact closely with the IT staff and, in order to be successful, all project stakeholders must be aware of the expectations of their counterparts (Beck, et al., 2001).

The secondary audience will be department managers and chief information officers (CIOs). When change is introduced to an organization, engagement will often be driven from the top-down (Gandomani & Nafchi, 2014). Introducing agile methods to an organization in which

they have not previously been used requires managers to create a movement, not issue a mandate (Gandomani & Nafchi, 2014). Managers work most closely with their teams and will have the strongest impact in the adoption of agile management planning and successful execution within their teams (Tessem, 2014). This study may be able to serve as a blueprint or guide for managers to outline the advantages of adopting agile methodologies within their teams.

Search Report

Search strategy. I performed the initial search using the advanced search function of the Google Scholar search engine to take advantage of the date filter function. Unless the journal article can be considered a foundational text, it is likely that research performed more than ten years ago will be of lesser value than more recent articles. I used Google Scholar's advanced search which allows returns to be filtered so that searches will only return research published after a specified date.

Once I collected samples of three or four journal articles I saved them to Zotero, which has a tab listing the associated metadata tags of each journal article. Armed with this list of metadata tags, I began the initial search of the University of Oregon Library system. The initial step of using Zotero to identify a set of metadata tags was certainly not necessary, as a similar result could have been accomplished directly within the UO Library system, but at the time I deemed the method useful for uncovering previously unexplored metadata tags. I used a filter similar to that of the advanced Google Scholar to only return research published after a specified date. A significant advantage of the OU Library system is that it can now display a short list of journal articles which have also been accessed and reviewed by those who have previously read the article. This feature was extremely helpful in locating potentially relevant research.

Libraries and search engines. I accomplished research for this annotated bibliography using the following libraries and search engines:

- Google Scholar.
- UO Library system.
- Bonneville Power Administration Department of Energy (DOE) library system.

Databases. I accessed the following databases to search for reference materials for this annotated bibliography:

- Computer Source.
- CiteSeer.
- Academic Search Premier.
- Scholars' Bank.
- ScienceDirect.
- Web of Science.
- Computer Source.

Key words and phrases. It was important to have a search strategy to effectively locate relevant research before performing searches through the previously mentioned search engines and databases. One part of an effective search strategy is the search terms. Below is the list of such terms I used to find sources for this annotated bibliography:

- Agile methods.
- Critical success factors.
- Software development.
- Innovation adoption.
- Organization(al).

- Change resistance.
- Overcome.
- Introduce/introducing.
- Pitfalls.

Documentation Method

Documentation approach. I used Zotero throughout this project to record and organize research. I almost always performed research for this project on a two-monitor system. When I read potential reference sources on one monitor I used the other to take notes or create an outline of the research using Microsoft Word documents. I then added these documents to the stand-alone Zotero tool and attached them directly to the referenced research. Through the research process three distinct categories became apparent (see below). I created folders for these research categories within Zotero and appropriately sorted the entries for each research entry.

I recorded all references and citations manually in American Psychological Association (APA) format. Electronic aids, such as those offered by Zotero and Microsoft Word, are not considered reliable and so were not used.

Research categories. I organized research for this project into the following two categories:

1. Agile effectiveness within an organization.
2. Best practice for ensuring a successful initial agile deployment.

Evaluation criteria. I evaluated research sources using several criteria and methods outlined by material provided by the University of Florida's Center for Public Issues Education website (Center for Public Issues in Education, n.d.). The specific criteria identified in the Center

for Public Issues in Education reference material are *authority, timeliness, quality, relevance* and lack of *bias.*

Authority: I did not necessarily discard research sources if the authors lacked a PhD in the subject but having an advanced degree does contribute to an author's credibility as an authority in the subject matter. Most search engines and research databases can identify how many times a source has been cited in other sources; a person is considered a greater authority on a subject when the author's research is cited by others, and more citations lead to greater authority. I also sought sources from peer-reviewed, scholarly journals, as these articles have gone through a rigorous editing process.

Timeliness: I limited most of the material for this research to sources published within the last ten years. However, I made rare exceptions for material which is considered foundational to the discipline. For example, the Agile Manifesto is now over 17 years old but is still referenced due to its significance and relevancy to this topic.

Quality: I sought sources with few, if any, grammatical or spelling errors. I noted whether the material in candidate sources was clearly presented and well organized.

Relevance: I selected material that applies to the research topic and gives context to the problem description or aids in answering the stated research questions.

Lack of *bias:* I sought material that is objectively presented, without any apparent evidence of bias. I checked to ensure the author(s) did not use emotionally charged language. I also ensured the conclusions reached were corroborated by conclusions of similar research.

Annotated Bibliography

Introduction to the Annotated Bibliography

The following annotated bibliography contains fifteen references and is organized into two categories: *Agile Effectiveness within an Organization* and *Best Practices for Ensuring a Successful Initial Agile Deployment*. The selected references are designed to (a) provide an understanding of agile methodology for those who may be unfamiliar with the concept, (b) provide justification of the benefits to an organization should they decide to implement agile methods, and (c) provide an understanding of the challenges, impediments, pitfalls, and deployment strategies organizations should be aware of before initiating an agile methodology deployment.

Category A: Agile Effectiveness within an Organization

Campanelli, A. & Parreiras, F. (2015). Agile methods tailoring – A systematic literature review. *Journal of Systems and Software, 110,* 85-110. doi:10.101/j.jss.2015.08.035.

Abstract. *Background*: The software development industry has been adopting agile methods instead of traditional software development methods because they are more flexible and can bring benefits such as handling requirements changes, productivity gains and business alignment. *Objective*: This study seeks to evaluate, synthesize, and present aspects of research on agile methods tailoring including the method tailoring approaches adopted and the criteria used for agile practice selection.

Method: The method adopted was a Systematic Literature Review (SLR) on studies published from 2002 to 2014. Results: 56 out of 783 papers have been identified as describing agile method tailoring approaches. These studies have been identified as case studies regarding the empirical research, as solution proposals regarding the research type, and as evaluation studies regarding

the research validation type. Most of the papers used method engineering to implement tailoring and were not specific to any agile method on their scope.

Conclusion: Most of agile methods tailoring research papers proposed or improved a technique, were implemented as case studies analyzing one case in details and validated their findings using evaluation. Method engineering was the base for tailoring, the approaches are independent of agile method and the main criteria used are internal environment and objectives variables.

Summary. This article is a systematic review of existing research into methods of tailoring deployments of agile methodology, including a summary of the research and identification of trends and gaps in the research. To accomplish this goal three research questions were proposed and answered based on the literature review:

1. What are the methodological aspects of agile methods tailoring research?
2. How practical has agile methods tailoring research been?
3. What criteria have been used for research into agile methods tailoring?

Following a review of existing literature, the authors provide an in-depth review of the research methods and search strategy used. The search strategy was a five-step process which began with an initial identification of 783 research papers. Iterative search refinements filtered this number to the 56 papers which were used in the authors' analysis. From this analysis, the authors identified the top fifteen most adopted agile practices. The top five agile practices from this list and the percentages of sources indicating their adoption are:

1. Daily stand-ups – 85%
2. Iteration planning – 75%
3. Retrospectives – 75%
4. Unit testing – 72%

5. Release planning – 70%

The two most common methods of tailoring agile deployments were determined to be *contingency factors* and *method engineering*. An organization engages in the contingency factors approach by selecting multiple agile methods to be available to a software development project. Teams then select those methods best suited to the particular project based on factors such as impact, priority, and experience. Method engineering entails the in-house creation of organization or project-specific methods and not adopting methods from vendors or a larger community.

This article is relevant to this research because it concisely identifies the most widely deployed agile practices and it identifies several methods organizations can use to tailor their agile deployments in forms that work best for them.

Chow, T. & Cao D. (2008). A survey of critical success factors in agile software projects.

Journal of Systems and Software, 81(6), 961-971. doi:10.1016/j.jss.2007.08.020

Abstract. While software is so important for all facets of the modern world, software development itself is not a perfect process. Agile software engineering methods have recently emerged as a new and different way of developing software as compared to the traditional methodologies. However, their success has mostly been anecdotal, and research in this subject is still scant in the academic circles. This research study was a survey study on the critical success factors of Agile software development projects using quantitative approach.

Based on existing literature, a preliminary list of potential critical success factors of Agile projects was identified and compiled. Subsequently, reliability analysis and factor analysis were conducted to consolidate this preliminary list into a final set of 12 possible critical success factors for each of the four project success categories – Quality, Scope, Time, and Cost.

A survey was conducted among Agile professionals, gathering survey data from 109 Agile projects from 25 countries across the world. Multiple regression techniques were used, both at the full regression model and at the optimized regression model via the stepwise screening procedure. The results revealed that only 10 out of 48 hypotheses were supported, identifying three critical success factors for Agile software development projects: (a) Delivery Strategy, (b) Agile Software Engineering Techniques, and (c) Team Capability.
Limitations of the study are discussed together with interpretations for practitioners. To ensure success of their projects, managers are urged to focus on choosing a high-caliber team, practicing Agile engineering techniques and following Agile-style delivery strategy.

Summary. This article begins with an empirical analysis of the Critical Success Factors (CSFs) and Failure Factors (FFs) commonly identified throughout existing scholarly articles about agile methodology. The authors performed a literature review and identified 36 CSFs categorized into five dimensions: *organizational*, *people*, *process*, *technical*, and *project*. The authors also researched pitfalls and identified 19 FFs, which were categorized into four dimensions: *organizational*, *people*, *process*, and *technical*. From these two lists of possible factors that may affect the failure or success of agile projects, factors with shared similar characteristics were consolidated into a reduced list of 39 factors. The authors then performed reliability analysis on each of the 39 factors. Final results revealed the twelve CSFs believed to have the greatest impact on agile project success. These twelve CSFs created the twelve hypotheses of the study and spanned four success dimensions: *quality*, *scope*, *time*, and *cost*.

The researchers then employed a multi-national web survey to gather real-world data about these twelve CSFs. This source is relevant to this study because it includes findings from analyses of literature research on agile implementations and the authors' conclusions from the

survey data, which revealed the following three Critical Success Factors for agile software development:

1. A workable product delivery strategy.
2. A proper practice of agile software engineering practices.
3. A high-functioning and reliable team.

The authors also noted factors which may be critical to success in various dimensions, including the software development lifecycle, strategic-planning, and project estimations:

- A resilient agile project management process.
- A team environment conducive to agile methodologies.
- Customers who are engaged and involved.

Holh, P., Münch, J., Schneider, K. & Stupperich, M. (2016). Forces that prevent agile adoption in the automotive domain. In Abrahamsson, P., Jedlitschka, A., Ngyuen, D., Felderer, M., Amasaki, S., & Mikkonen, T. (Eds), *Product-focused software process improvement*, (pp. 468-477). New York, NY: Springer International Publishing. doi:10.1007/978-3-319-49094-6_32

Abstract. *Context***:** The current transformation of automotive development towards innovation, permanent learning and adapting to changes are directing various foci on the integration of agile methods. Although, there have been efforts to apply agile methods in the automotive domain for many years, a wide-spread adoption has not yet taken place.

Goal: This study aims to gain a better understanding of the forces that prevent the adoption of agile methods.

Method: Survey based on 16 semi-structured interviews from the automotive domain. The results are analyzed by means of thematic coding.

Results: Forces that prevent agile adoption are mainly of organizational, technical and social nature and address inertia, anxiety and context factors. Key challenges in agile adoption are related to transforming organizational structures and culture, achieving faster software release cycles without loss of quality, the importance of software reuse in combination with agile practices, appropriate quality assurance measures, and the collaboration with suppliers and other disciplines such as mechanics.

Conclusion: Significant challenges are imposed by specific characteristics of the automotive domain such as high quality requirements and many interfaces to surrounding rigid and inflexible processes. Several means are identified that promise to overcome these challenges.

Summary. This article reports on an interview-based investigation into the specifics of agile methodology that seeks to understand hindering forces which may be acting against agile adoption and identify potential solutions. Participants had to have between three and twenty years of employment with their company and already be familiar with agile practices for software development to be considered for the study. Analysis of the interviews revealed six forces at work on agile adoption; some forces were working for agile adoption and some forces were working against. The forces the authors identified as working for agile adoption are:

- Trigger – Events which initiate a change and push a team or organization towards agile. An example of a trigger events would be management exhibiting frustration with how long it takes to receive a demonstration prototype.
- Push – Forces pushing IT teams towards agile methodology based on problems or demands. Examples of push events might include high errors rates in acceptance tests or deliverable deadlines which are frequently missed.

- Pull – Forces pulling IT teams towards agile based on the perceived attractiveness of the methodology. An example of a pull event might be the hiring of a well-respected team leader who takes on the role of an agile advocate and coach for his or her team.

Forces working against adoption are:

- Inertia – Mindsets and existing corporate culture will often work against attempts to adopt and implement new practices. The perception is that current development processes are sufficient and there is little value to be gained by trying to learn new ways of working. However, it is noted that a misunderstanding of agile and how to apply agile methods within an organization is a major cause of inertia.
- Anxiety – IT teams were concerned management would not assign greater autonomy to agile teams, thereby lessening their own responsibilities. Managers were concerned about how to provide correct estimates for efforts when employing agile practices. Managers were also concerned that employees who did not wish to adopt agile practices may leave.
- Context – Company structure may create too many project stakeholders and make feature negotiation more difficult. Existing process dependencies may create additional challenges for agile adoption.

The researchers then reviewed methods interviewees used to overcome each of the forces which work against agile adoption. The authors investigated detailed challenges that exist within each of these forces and offered solutions for each challenge. These solutions are:

- Additional and effective communication is a key requirement to overcome organizational forces of inertia which act to maintain the status quo.

- Focus must be on the effective prioritization of features and not on providing stakeholders with accurate estimates.
- Engage and encourage team members by providing a sense of ownership in the process.
- Reduce process dependencies within the organization that impede agile adoption.

While the research presented in this article is narrow in scope, it does provide conclusions which may be supported by other, broader research.

Inayat, I., Salim, S., Marczak, S., Daneva, M. & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior,* 51, 915-929. doi:10.1016/j.chb.2014.10.046

Abstract. Unlike traditional software development methods, agile methods are marked by extensive collaboration, i.e. face-to-face communication. Although claimed to be beneficial, the software development community as a whole is still unfamiliar with the role of the requirements engineering practices in agile methods. The term "agile requirements engineering" is used to define the "agile way" of planning, executing and reasoning about requirements engineering activities. Moreover, not much is known about the challenges posed by collaboration-oriented agile way of dealing with requirements engineering activities. Our goal is to map the evidence available about requirements engineering practices adopted and challenges faced by agile teams in order to understand how traditional requirements engineering issues are resolved using agile requirements engineering. We conducted a systematic review of literature published between 2002 and June 2013 and identified 21 papers, that discuss agile requirements engineering. We formulated and applied specific inclusion and exclusion criteria in two distinct rounds to determine the most relevant studies for our research goal. The review identified 17 practices of agile requirements engineering, five challenges traceable to traditional requirements engineering

that were overcome by agile requirements engineering, and eight challenges posed by the practice of agile requirements engineering. However, our findings suggest that agile requirements engineering as a research context needs additional attention and more empirical results are required to better understand the impact of agile requirements engineering practices e.g. dealing with non-functional requirements and self-organizing teams.

Summary. The authors seek to close the knowledge gap between Requirements Engineering (RE) and the best methods for adopting agile methodology. To do so, the authors posed three research questions:

1. What are the adopted practices of agile RE according to published materials?
2. What are the challenges of traditional requirements engineering?
3. What are the challenges of agile requirements engineering?

To answer these questions the authors conducted a systematic review of the existing literature. Initial research revealed 543 studies; the authors subjected these studies to iterative refinements of the research criteria until identifying the final 21 most applicable papers reviewed by the authors.

Their analysis revealed seventeen agile practices most used by organizations when implementing agile methodologies. The top ten of these practices are: face-to-face communication, customer involvement, user stories, iterative requirements, requirement prioritization, change management, cross-functional teams, prototyping, testing, and modelling.

The authors' research also identified the top five challenges associated with traditional development practices which are alleviated by agile practices:

1. Communication issues – Agile focuses on frequent face-to-face communication between development teams and the customer.

2. Overscoping – IT teams receive the full project feature list and it is then left to the IT team to manage effective delivery.
3. Requirements validation – The customer prioritizes requirements every iteration and frequent product delivery aids in validating requirements.
4. Requirements documentation – Frequent face-to-face meetings reduces ambiguity and the need for overly cumbersome documentation.
5. Limited customer involvement – Customers' active participation ensures all project goals will be met.

The authors' analysis identified the challenges associated with agile: minimal documentation, customer unavailability, inappropriate or inadequate systems architecture, unrealistic budget and time estimation, neglecting non-functional requirements, unaligned customer expectations, contractual limitations, and requirements changes.

This article is relevant to this research because it identifies through a literature review the benefits agile methodologies may provide IT teams, how these agile methods may be better suited to address IT challenges than other traditional development methods, and challenges IT teams have faced when engaged in agile methodology.

Maruping, L., Venkatesh, V., & Agarwal, R. (2009). A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research*, *20*(3), 377-399. Retrieved from http://www.jstor.org/stable/23015471

Abstract. In this paper, we draw on control theory to understand the conditions under which the use of agile practices is most effective in improving software project quality. Although agile development methodologies offer the potential of improving software development outcomes, limited research has examined how project managers can structure the software development

environment to maximize the benefits of agile methodology use during a project. As a result, project managers have little guidance on how to manage teams who are using agile methodologies. Arguing that the most effective control modes are those that provide teams with autonomy in determining the methods for achieving project objectives, we propose hypotheses related to the interaction between control modes, agile methodology use, and requirements change. We test the model in a field study of 862 software developers in 110 teams. The model explains substantial variance in four objective measures of project quality—bug severity, component complexity, coordinative complexity, and dynamic complexity. Results largely support our hypotheses, highlighting the interplay between project control, agile methodology use, and requirements change. The findings contribute to extant literature by integrating control theory into the growing literature on agile methodology use and by identifying specific contingencies affecting the efficacy of different control modes. We discuss the theoretical and practical implications of our results.

Summary. This research is intended to contribute to existing literature by shedding light on managerial methods which support agile methodology and examining the efficacy of different agile control mechanisms in order to determine how managers can structure their team environments to maximize the benefits of agile methodology. The authors conducted a field study that involved 862 software developers in 110 teams to test their proposed model which hypothesized that teams that had the most autonomy experienced the most success. Agile methodologies are known to encourage the delegation of authority to team members, which may be seen as a challenge to traditional management styles. Agile also encourages team members to be responsible for managing their own processes and delegating task responsibility to fellow

team members, thus further weakening management's position with the agile software development team.

Data analysis showed a statistically significant correlation between the use of agile methodology and net positive impacts on project quality. The results also identified a need for agile teams to be responsive to changes in design requirements. Specific findings regarding management and team autonomy were that autonomy is most effective when granted to the team as a whole and not individually and when it is conditional upon teams reaching specific performance targets. This article is relevant to this research because it offers findings on how to most effectively manage agile teams.

Ramesh, B., Cao L., & Baskerville, R. (2010) Agile requirements and engineering challenges: An empirical study. *Information Systems Journal, 20*(5), 449-480. doi:10.1111/j.1365-2575.2007.00259.x

Abstract. This paper describes empirical research into agile requirements engineering (RE) practices. Based on an analysis of data collected in 16 US software development organizations, we identify six agile practices. We also identify seven challenges that are created by the use of these practices. We further analyze how this collection of practices helps mitigate some, while exacerbating other risks in RE. We provide a framework for evaluating the impact and appropriateness of agile RE practices by relating them to RE risks. Two risks that are intractable by agile RE practices emerge from the analysis. First, problems with customer inability and a lack of concurrence among customers significantly impact agile development. Second, risks associated with the neglecting non-functional requirements such as security and scalability are a serious concern. Developers should carefully evaluate the risk factors in their project

environment to understand whether the benefits of agile RE practices outweigh the costs imposed by the challenges.

Summary. The objective of the author was to better understand how and why requirements analysis of teams operating with agile methodologies differs from traditional requirement analysis. The following three research questions were intended to satisfy this objective:

1. What Requirements Engineering (RE) practices are adopted in software development in agile environments?
2. What problems are encountered when using agile practices?
3. Do agile practices mitigate or increase requirements risk?

The authors sought to answer these questions by examining RE in sixteen real-world agile project implementations in sixteen separate organizations that develop software using agile methodology. These case studies were categorized into two groups: (a) ten organizations that develop in agile software environments whose agile components were pieced together to best fit their needs rather than following any one particular agile "brand" and (b) the remaining six organizations that follow either Scrum or XP versions of agile methodology.

The authors' analysis revealed six shared practices and seven shared challenges.

Shared practices:

- Face-to-face communication is preferred over written specifications.
- Iterative requirements engineering, where requirements are not defined at the beginning of a project and instead emerge during the development process.
- Requirements prioritization:
    - Setting requirements early enables early delivery of valuable features.

    - Customers and developers understanding of the project improves during development and new requirements can be identified and added or existing ones modified.
    - Prioritization must keep up with changing requirements.
- Requirement changes are managed through constant planning; teams must be able to adapt and react quickly to changing requirements.
- Use of prototypes:
    - Prototype demonstration and experimentation allows requirements to be validated and refined.
    - Prototypes promote the ability to quickly deploy new versions of products.
- Review meetings and acceptance tests serve as a project progress report.

Shared challenges:

- Cost and schedule estimation:
    - Agile approach makes estimating more difficult.
    - Agile methodology embraces change, making estimates unreliable.
    - Schedule estimates are based on user stories, which often do not contain quantifiable data.
- System architecture is inadequate or inappropriate, and the inability to scale may require refactoring – a lengthy and expensive process.
- Neglect of non-functional requirements.
- Customer access and participation.
- Prioritization is based on a single criterion, such as satisfying a business value or focusing on product time-to-market.

- Insufficient requirements are provided for verification.
- Minimal documentation.

This article is relevant to the research because it identifies several agile methodology practices shared by agile practitioners as well as several shared challenges. Several of these practices and challenges are also reported in similar related research contained within this annotated bibliography, adding weight to the collective findings derived from independent primary research.

Serrador, P. & Pinto, J. (2014). Does agile work? – A quantitative analysis of agile project success. *International Journal of Project Management, 33*(5), 1040-1051. doi:10.1016/j.ijproman.2015.01.006

Abstract. The Agile project management methodology has been widely used in recent years as a means to counter the dangers of traditional, front-end planning methods that often lead to downstream development pathologies. Although numerous authors have pointed to the advantages of Agile, with its emphasis on individuals and interactions over processes, customer collaboration over contracts and formal negotiations, and responsiveness over rigid planning, there are, to date, very few large-scale, empirical studies to support the contention that Agile methods can improve the likelihood of project success. Developed originally for software development, it is still predominantly an IT phenomenon. But due to its success it has now spread to non-IT projects. Using a data sample of 1002 projects across multiple industries and countries, we tested the effect of Agile use in organizations on two dimensions of project success: efficiency and overall stakeholder satisfaction against organizational goals. We further examined the moderating effects of variables such as perceived quality of the vision/goals of the project, project complexity, and project team experience. Our findings suggest that Agile

methods do have a positive impact on both dimensions of project success. Further, the quality of the vision/goals is a marginally significant moderator of this effect. Implications of these findings and directions for future research are discussed.

Summary. The stated purpose of this article is to investigate through a large-scale quantitative study the evidence that agile methods create higher rates of successful projects than traditional methods. The authors identify several main differences between traditional and agile development methods.:

- Communication is informal in agile but more formal in traditional methods.
- Knowledge management is tacit in agile, with traditional methodologies knowledge management is more explicit.
- Traditional methods have a life-cycle software development model, while agile methodology uses an evolutionary model where product is delivered through a series of iterations.
- Agile teams are an exercise in collaboration whereas traditional development methods use a structured command-and-control management style.

The authors present a detailed examination of existing published literature and note how the literature findings support the effectiveness of agile methods over more traditional methods. Effectiveness is measured as project success across two dimensions: project efficiency (meeting cost, time, and scope goals) and stakeholder success (satisfying expectations of project sponsors and other stakeholders).

As part of a large-scale quantitative study, the authors sent invitations to a survey to members of the Project Management Institute (PMI) and LinkedIn project management groups. The authors received survey responses from 859 individuals, reporting on a total of 1,539

different projects. The authors analyzed the results of the survey to determine if quantitative data existed which related the effectiveness of agile methodology and iterative planning in project management success rates. The results of this analysis show a direct and statistically significant positive correlation between the levels of agile methodologies and iterative planning used and the levels of project success. The findings in this article are relevant to this study because of the empirical validation it offers to the hypothesis that projects managed with agile methods have a greater likelihood of successful outcomes than do projects managed by traditional methods.

Tessem, B. (2014). Individual empowerment of agile and non-agile software developers in small teams. *Information and Software Technology, 56*(8), 873-889. doi:10.1016/j.infsof.2014.02.005

Abstract. *Context*: Empowerment of employees at work has been known to have a positive impact on job motivation and satisfaction. Software development is a field of knowledge work wherein one should also expect to see these effects, and the idea of empowerment has become particularly visible in agile methodologies, in which proponents emphasize team empowerment and individual control of the work activities as a central concern.

Objective: This research aims to get a better understanding of how empowerment is enabled in software development teams, both agile and non-agile, to identify differences in empowering practices and levels of individual empowerment.

Method: Twenty-five interviews with agile and non-agile developers from Norway and Canada on decision making and empowerment are analyzed. The analysis is conducted using a conceptual model with categories for involvement, structural empowerment and psychological empowerment.

Results: Both kinds of development organizations are highly empowered and they are similar in most aspects relating to empowerment. However, there is a distinction in the sense that agile developers have more possibilities to select work tasks and influence the priorities in a development project due to team empowerment. Agile developers seem to put a higher emphasis on the value of information in decision making, and have more prescribed activities to enable low-cost information flow. More power is obtained through the achievement of managing roles for the non-agile developers who show interest and are rich in initiatives.

Conclusion: Agile developers have a higher sense of being able to impact the organization than non-agile developers and have information channels that is significantly differently from non-agile developers. For non-agile teams, higher empowerment can be obtained by systematically applying low-cost participative decision making practices in the manager–developer relation and among peer developers. For agile teams, it is essential to more rigorously follow the empowering practices already established.

Summary. The author conducted a study to examine how empowerment of individuals within IT teams is enabled in both agile and non-agile environments with the aim of determining if implementation of agile methodology has an impact on levels of individual empowerment. The author noted that some positive effects of empowerment include increased productivity, greater product quality, innovation, higher job satisfaction levels, higher motivation and higher overall job performance. The author limited the research to the study of empowerment levels as the effects of empowerment are already well understood. The author notes that agile methods implicitly suggest high team empowerment levels but asserted that how agile methods contribute to feelings of empowerment by individual IT employee had not yet been examined by previous research. The author performed primary research by means of interviews and surveys of

individuals from a variety of industries working in the IT field, from both agile and non-agile organizations. The author concluded that high levels of team empowerment exist in both agile and non-agile IT environments but levels of individual empowerment are greater in agile environments. Agile team members also reported greater levels of decision-making and feelings that they and their work have an impact on their organizations. The findings presented in this article are relevant to this study because they support the hypothesis that the introduction of agile methodology on IT teams not only improves team performance but also improves the performance and work satisfaction levels of individuals within the teams.

Category B: Best Practices for Ensuring a Successful Initial Agile Deployment

Dikert, K., Paasivaara, M. & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *The Journal of Systems and Software, 119,* 87-108. doi:10.1016/j.jss.2016.06.013

Abstract. Agile methods have become an appealing alternative for companies striving to improve their performance, but the methods were originally designed for small and individual teams. This creates unique challenges when introducing agile at scale, when development teams must synchronize their activities, and there might be a need to interface with other organizational units. In this paper we present a systematic literature review on how agile methods and lean software development has been adopted at scale, focusing on reported challenges and success factors in the transformation. We conducted a systematic literature review of industrial large-scale agile transformations. Our keyword search found 1875 papers. We included 52 publications describing 42 industrial cases presenting the process of taking large-scale agile development into use. Almost 90% of the included papers were experience reports, indicating a lack of sound academic research on the topic. We identified 35 reported challenges grouped into nine

categories, and 29 success factors, grouped into eleven categories. The most salient success factor categories were management support, choosing and customizing the agile model, training and coaching, and mindset and alignment.

Summary. The authors performed a literature review of published research on large-scale agile initiatives, beginning with keyword searches from several targeted databases. Their initial iteration resulted in over 2, 500 research paper matches. Subsequent refinements of the search criteria resulted in the final 52 papers the authors used in this study. Forty-six of the papers provided the results of experiential analysis of industries regarding agile deployments, and six provided the results of case studies with clearly defined research methods. These papers were used to answer the following two research questions:

1. What challenges have been reported for large-scale agile transformations?
2. What success factors have been reported for large-scale agile transformations?

A critical analysis of the target literature revealed the following challenge types for agile transformations and how often these challenges were present in organizations pursuing agile transformations:

- Difficulties implementing agile (48%):
    - Teams misunderstand agile requirements.
    - Lack of guidance from agile coaches or inadequate training.
    - Agile is poorly customized to the needs of the organization.
- Requirements (38%):
    - Requirements are not adequately managed.
    - Difficulties creating estimates for user stories.
    - Gaps in planning between long and short term.

- Change Resistance (38%):
    - Skepticism towards new ways of working.
    - Management was unwilling to change or reluctant to relinquish micromanagement tendencies.
    - General adherence to the status quo.
- Organization (33%):
    - Unclear role for middle managers and project managers.
    - Management maintains waterfall method planning expectations.
    - Internal silos are difficult to overcome.
- Lack of buy-in (31%):
    - Lack of coaching and/or training.
    - Old process requirements are maintained.

The authors' analysis of the same literature then revealed the following shared factors which helped to enable success in agile transformations and how often these factors were employed:

- Choosing and customizing agile methods (48%):
    - Keep it simple.
    - Iterative agile deployments.
    - Old methodology mapped to new to ease adaptation.
- Mindset and Alignment (40%):
    - Organized team social functions.
    - Team goals aligned with organization goals.
- Management Support (38%):

    - Management support is visible.
    - Management has received training/education on agile methods.
- Training and Coaching (36%):
    - Provide formal training on agile methodology
    - Have agile coaches available to assist teams.
    - Learn by doing.
- Pilot Programs (33%):
    - Start small with a pilot program to gain acceptance.
    - Gain insights from pilots and leverage the insights for wider rollouts.

In all, the authors identified nine categories and 35 challenges in answer to the first research question and eleven categories and 29 success factors in answer to the second research question. This article is relevant to this research as the success factors demonstrate that agile deployments cannot simply be taken off-the-shelf but need to be planned and customized to the specific needs of the organization. It also shows that challenges to agile may exist throughout an organization and organizations which can unite in their commitment to transitioning to agile methods may have the greatest chance of success.

Gandomani, T. & Nafchi, M. (2015). An empirically-developed framework for agile transition and adoption: A Grounded Theory approach. *The Journal of Systems and Software, 107,* 204-219. doi:10.1016/j.jss.2015.06.006

Abstract. To date, few Agile transition and adoption frameworks have been proposed in the software industry. However, using them is not easy in practice and primarily requires a huge organizational overhead because of their complex and non-flexible structure. These drawbacks make such frameworks difficult to apply in small and medium-sized companies. We have

conducted a large-scale empirical research study using Grounded Theory approach with the participation of 49 Agile experts from 13 different countries. This study inductively developed a substantive Agile transition and adoption framework which appears to be simple and flexible. The main aim of this paper is to present the developed framework. The primary characteristics of this framework, including iterative, gradual, continuous, and value-based are in line with the Agile approach and show promise of being useful in software companies and organizations, regardless of size. This paper also describes how various steps of this framework could help software companies to achieve Agile transformation.

Summary. The essence of this study is to answer the following questions: How do software companies and teams manage the transition to Agile? What are some effective frameworks organizations can use for transition? What characteristics and features should a framework have? What should be its activities? To answer these questions the authors of the article used a Grounded Theory (GT) approach to study 49 agile practitioners from 13 different countries. The authors began with a literature review, examining previously proposed agile frameworks and discussing the relative merits and failings of each. Of particular note were cases where developers of agile frameworks later returned to criticize their own models as unwieldy or overly cumbersome. Based on the analysis of the literature, interviews with practitioners, and prior knowledge of agile practices, the authors developed the unique model for Agile Transformation/Transition Process (ATP) seen below in Figure 1.

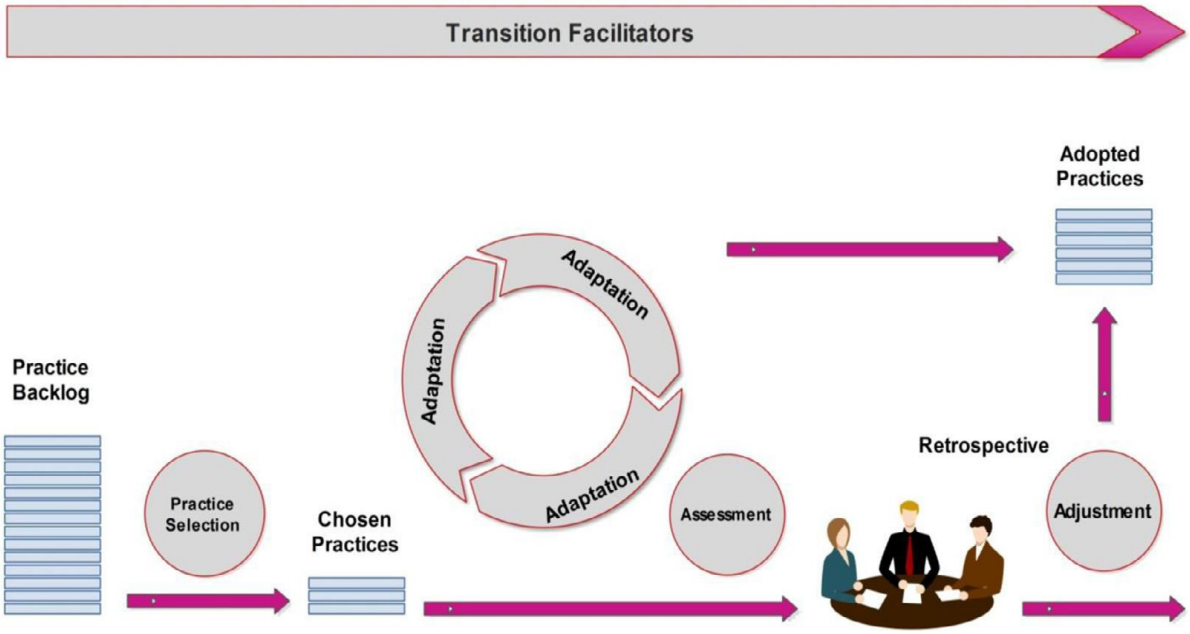

Figure 1. Framework for Agile Transformation/Transition Process. Reprinted from "An empirically-developed framework for Agile transition and adoption: A Grounded Theory approach," by T. Gandomani and M. Nafchi, 2015, *The Journal of Systems and Software, 107,* p. 216

Agile practices are maintained and prioritized in the *Practice Backlog*, teams then *Select* one or more practices from the practice backlog and begin the *Adaptation* process of the *Chosen Practices* to introduce the new practice into their latest development iteration. Once the iteration is completed an *Assessment* is performed to evaluate their successes and challenges in adopting the new agile practice. The team discusses their experience in the *Retrospective* and *Adjustments* are made as necessary. Once complete, the *Adopted Practices* are now a part of the team's agile methodologies. This process is then repeated in iterative steps as often as necessary.

The research the authors performed and their proposed model of implementing agile in a team environment are relevant to this study because they provide a concrete example, based on

real-world case studies, of an agile implementation model which can be proposed to audience members.

Gregory, P., Barroca, L., Sharp, H., Deshpande, A. & Taylor, K. (2015). The challenges that challenge: Engaging with agile practitioners' concerns. *Information and Software Technology, 77*(3), 92-104. doi:10.1016/j.infsof.2016.04.006

Abstract. *Context:* There continues to be concern that research is not addressing the challenges of agile practitioners. For the benefit of academia and industry, researchers need to be aware of practitioners' challenges and their context so that relevant and applicable research is undertaken. *Objective:* This paper investigates two research questions: what challenges do agile practitioners face? How do practitioner challenges manifest themselves in an organizational setting? It aims to map the practitioner challenge landscape, explore challenge characteristics, compare findings with previous literature and identify implications for research that is relevant to practice. *Method:* A combination of methods was used: elicitation of practitioner challenges collected using a Challenge Wall at a series of practitioner events; organizational case study using interviews, document analysis and observation; and online survey. Findings were then compared to previous publications. *Results:* Challenges collected from the Challenge Wall were grouped under 27 subthemes and seven themes: claims and limitations, organization, sustainability, culture, teams, scale, and value. Investigating one challenge in the case study uncovered a set of new challenges, which were inter-related. Over 50% of survey respondents experienced challenges highlighted in the case study. *Conclusion:* The landscape of agile practitioner challenges is complex and intertwined. Some challenges, such as doing agile in a non-agile environment, are multi-dimensional, affect many

aspects of practice, and may be experienced simultaneously as business, organizational, social and adaptation problems. Some challenges, such as understanding cultural change or measuring agile value, persist and are hard to address, while others, such as adoption, change focus over time. Some challenges, such as governance and contracts, are under-researched, while others, such as business and IT transformation, have been researched but findings have not had the expected impact. Researchers wishing to address practitioner challenges need to treat them in context rather than in isolation and improve knowledge transfer.

Summary. The authors seek to answer two research questions:

1. What challenges do agile practitioners face?
2. How are agile practitioner challenges manifested in an organization?

The authors used a Challenge Wall that was deployed five times at agile practitioner conferences throughout a one-year period to answer the first question. During these conferences attendees were invited to fill out "challenge cards" and place them on a board which was set up in a high-traffic area. One challenge was logged per card and people were encouraged to fill out as many cards as they wished about the nature and context of agile challenges. In all, 194 cards were collected, which were later subjected to independent thematic analysis by three separate researchers. This analysis resulted in the organization of challenges into seven themes and 27 subthemes. The themes were:

1. Claims & Limitations.
2. Organization(al).
3. Culture.
4. Teams.
5. Sustainability.

6. Scaling.
7. Value.

While the cards revealed challenges and themes, they did not provide context to enable the analysis of the challenges in the context of a specific organization. This context was necessary to answer the second research question and was addressed with a case study.

After issuing queries for appropriate organizations to use as the subject of a case study, an anonymous organization "BigBank" was selected. BigBank had already deployed an agile framework called Dynamic Systems Development Method (DSDM), but was experiencing a severe disconnect between the managers at the field office attempting to deploy this methodology and managers at the head office to which the field office reported. The head office was skeptical about the agile approach and so lacked sufficient trust in the field office to be able to control their projects. This lack of trust led to miscommunication and attempts by the head office to micromanage the field office. Four challenge themes emerged as a result of the field office attempting to implement agile in a wide non-agile environment:

1. Communication – The head office preferred formal, written communication while the field office sought informal face-to-face communication. This challenge was linked to corporate culture.
2. Expectation – The head office expected certainty of project time, budget and specifics and the field office sought to adapt as needed. This challenge was linked to business value.
3. Control – The head office viewed re-prioritization as a lack of control. This challenge was linked to insufficient management buy-in.

4. Reporting – The head office expected consistent reporting but the field office was still in the process of maturing and deploying their agile methodologies. This challenge was linked to agile adoption.

The authors' analysis of the challenge wall and the case study revealed the following conclusions:

- Challenges with agile implementations are complex and interlinked. Challenges need to be addressed in context (real-world), not in isolation (academic).
- Some agile challenges are difficult to address and will benefit from further research.
- Some agile challenges have persisted for many years and the focus on how to address challenges may have shifted towards organizational concerns.
- Agile practitioners are more concerned with sustainability and less concerned with adoption.
- Some agile challenges (misconceptions and hype) can be best addressed with training.
- Research on some agile challenges has already been performed but findings are not having the expected impact. For example, it is now generally understood that change can be difficult and in order to take root change needs to be embraced by management, but management can be slow to do so. The authors speculate that the findings may need to be re-packaged to find a wider audience.

This article in relevant to this research because it contains an analysis and findings of real-world challenges encountered by agile practitioners and uses a case study to provide context into how these challenges may present themselves in organizations.

Laanti, M., Solo, O., & Abrahamsson, P. (2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology, 53*(3), 276-290. doi:10.1016/j.infsof.2010.11.010

Abstract. *Context*: Many organizations have started to deploy agile methods, but so far there exist only a few studies on organization-wide transformations. Are agile methods here to stay? Some claim that agile software development methods are in the mainstream adoption phase in the software industry, while others hope that those are a passing fad. The assumption here is that if agile would not provide real improvement, adopters would be eager at first but turn pessimistic after putting it into practice.

Objective: Despite the growing amount of anecdotal evidence on the success of agile methods across a wide range of different real-life development settings, scientific studies remain scarce. Even less is known about the perception of the impacts of agile transformation when it is deployed in a very large software development environment, and whether agile methods are here to stay. This study aims to fill that gap by providing evidence from a large-scale agile transformation within Nokia. While we have yet to confirm these findings with solid quantitative data, we believe that the perception of the impacts already pinpoints the direction of the impacts of large-scale agile transformation.

Method: The data were collected using a questionnaire. The population of the study contains more than 1000 respondents in seven different countries in Europe, North America, and Asia. Results: The results reveal that most respondents agree on all accounts with the generally claimed benefits of agile methods. These benefits include higher satisfaction, a feeling of effectiveness, increased quality and transparency, increased autonomy and happiness, and earlier

detection of defects. Finally, 60% of respondents would not like to return to the old way of working.

Conclusion: While the perception of the impact of agile methods is predominantly positive, several challenge areas were discovered. However, based on this study, agile methods are here to stay.

Summary. The purpose of this study was to bridge the gap between anecdotal agile success stories and verifiable quantitative data which can provide empirical evidence that supports the perceived usefulness and sustainability of agile methodologies. The authors examined the backgrounds of two groups of IT workers: those with experience with agile methods and those with little or no experience with agile methods. The authors sought to correlate background with satisfaction levels and perceptions of whether the benefits of agile methods outweighed the challenges which come with introducing this new methodology. The study took the form of more than 1,000 questionnaire survey responses from employees in Asia, Europe and North America regarding different aspects of agile development from the viewpoints of various organizational stakeholders.

One of the more interesting findings was that those with positive views towards agile adoption viewed as benefits of agile what those with a negative view towards agile viewed as challenges of agile adoption. For example, the negative group saw requirements management and iterative planning as one of the primary challenges of agile while the positive group considered these elements to be primary agile benefits. The authors also found that the longer one is exposed to agile methods, the more likely the person is to have positive views towards agile methods, a finding that holds true even for those who initially had negative views towards agile adoption or who were accustomed to working in non-agile teams for as long as ten years.

Of those surveyed, only 9% reported wanting to go back to their previous methods while 60% wanted to stay with agile and the remaining expressed no preference.

This article is relevant to this research as it supports the hypothesis that if agile methods can be successfully introduced and sustained for at least a year then there is less likelihood of those methodologies not being self-sustaining.

Mergel, I. (2016). Agile innovation management in government: A research agenda. *Government Information Quarterly*,*33*(3), 516-523. doi:10.1016/j.giq.2016.07.004

Abstract. Governments are facing an information technology upgrade and legacy problem: outdated systems and acquisition processes are resulting in high-risk technology projects that are either over budget or behind schedule. Recent catastrophic technology failures, such as the failed launch of the politically contested online marketplace Healthcare.gov in the U.S. were attributed to an overreliance on external technology contractors and failures to manage large-scale technology contracts in government. As a response, agile software development and modular acquisition approaches, new independent organizational units equipped with fast reacting teams, in combination with a series of policy changes are developed to address the need to innovate digital service delivery in government. This article uses a process tracing approach, as well as initial qualitative interviews with a subset of executives and agency-level digital services members to provide an overview of the existing policies and implementation approaches toward an agile innovation management approach. The article then provides a research framework including research questions that provide guidance for future research on the managerial implementation considerations necessary to scale up the initial efforts and move toward a collaborative and agile innovation management approach in government.

Summary. The author examines the opportunities and challenges which exist for agile methodologies in government agencies. The author collected data by reviewing existing literature regarding agile methods and contrasting findings to traditional development methods. The author then reviewed the traditional government software development approach which usually relies on a sequential, waterfall methodology. The author notes that government agencies will usually have all project phases predefined with deadlines and tied-to-payment deliverables, leaving little room for the innumerable adjustments and corrections which are commonplace in IT project. The author asserts that this rigidity can often lead to expensive follow-up service contracts and unbudgeted cost overruns.

The author notes that an important aspect of agile is the ability of agile project teams to react to change and that such flexibility can be of particular advantage for institutions know for bureaucratic rigidity. The author uses the disastrous rollout of the federal government's Healthcare.gov website to illustrate the point that traditional acquisition methods used by government agencies are often not up to the task of managing projects in today's digital age. The author also states that when implementing agile in government institutions, a top-down mandate may not be sufficient. Instead, the author asserts that a multi-layered approach may be necessary. Some components of this approach are:

- Implement Blanket Purchase Agreements (BPAs) in which preapproved vendors can demonstrate prototypes rather than be required to only provide fully working deliverables.
- Move away from "grand design" projects to smaller, scalable projects.
- Examine clients' underlying needs and make these needs the focus of the project.

The author then notes that agile methodology, as observed in government organizations, operates on two layers: a policies layer and a management layer. The author further explores these layers and their building blocks in a framework the author calls the "agile innovation management approach" (p. 517). This framework explores the policies and management practices at work in government agencies. Key findings of this framework include a strong top-down management structure for policy implementation and adherence operating within government agencies which makes the buy-in of top managers an essential need when introducing change, as well as the need for top managers to provide 'air cover' for IT teams so they feel free to explore new management methods. The author finishes with a series of agile design principles to be considered when implementing agile in government, including:

- Open By Default – Source code, whether built in-house or vendor supplied, will be open source and shared on social coding sites such as GitHub. This will help prevent other government agencies from duplicating efforts and help remove silos.
- Agile Leadership – Government agencies are highly risk-averse and are managed in a strict top-down fashion. Agile leadership works towards cultural changes of encouraging open collaboration and discourages the belief that only the only safe approach is to deliver a complete plan.
- Alternative Contracting and Outsourcing Approach – Contractors, contract managers, clients, and end-users are consulted when designing contract specifications and there is acknowledgement of and estimates for the project adjustments as they proceed through iterations.

This article is relevant to this research because it may reveal previously unrecognized challenges faced by government teams and offer suggestions for how to address such challenges.

Sureshchandra, K. & Shrinivasavadhani, J. (2008). Moving from waterfall to agile. In *Proceedings from Agile, 2008: AGILE '08 Conference* (pp.99-101). Toronto, ON, Canada: IEEE *Explore.* doi:10.1109/Agile.2008.49

Abstract. In a crisis ridden business environment, customers have become very averse to surprises. Business windows have become smaller; there is a heightened need for shorter development cycles and higher visibility. All this is translating into more and more customers specifically asking for agile. Service organizations such as Wipro Technologies need to adopt lean and agile methodologies to support the transition. As agile coaches, the biggest challenge we face is in transitioning the mindset of the team from that of a waterfall model to an agile thought pattern. Our experience in converting a waterfall team to agile is shared in this report.

Summary. The authors of this article performed a case study and then detailed the real-world experiences of agile coaches who have led teams in their transition when implementing agile methodologies. Some challenges expressed by IT teams when working under traditional methodologies include:

- Overworked teams in the weeks preceding deliverable releases.
- Delayed release dates and missed deadlines.
- Acceptance testing reveals high levels of defects.

The agile coaches performed their study at a large manufacturer of computer peripherals. They first met with the team to conduct agile training. Rather than list the agile practices the team needed to use, the authors let the team choose the agile practices the team believed would solve their problems. The team's choices resulted in a plan which only included time-boxing and daily standup meetings, demonstrating a clear reluctance to adopt agile. The agile coaches gradually addressed the team's concerns and added more involved agile methodologies to the

team processes over the following weeks of their first six-week iteration. Following this first six-week iteration a retrospective was performed where the team reviewed challenges they were experiencing with implementing agile and decided on several agile 'course-corrections' which would help them succeed. The following iteration was shortened to four-weeks and integrated these corrections. Following this second iteration another retrospective was performed. The following is a summary of their lessons learned:

- Teams may not be ready to fully implement agile; a gradual deployment may therefore work best.
- Mentoring and feedback from the agile coach was beneficial.
- Guide and coach rather than relying on a dogmatic approach in adopting agile.
- Some agile methods can be difficult for teams to adopt, such as paired-programming or continuous integration. Start small and gradually increase rigor.
- An agile coach needs a high level of involvement during the initial agile methodology deployment. The coach should conduct group and individual mentoring and focus on developing a self-disciplined team.
- Teams should choose agile practices depending on their need
- Carefully choose measurements as they will influence the way people work. For example, if only successful builds are measured, then developers will check-in code less frequently. Measuring how long a successful build takes encourages teams to build early.
- It may take time to get the Project Manager (PM) on board. Enrolling the PM in a Scrum Master program will help change a reluctant PM's mindset and can put the PM in touch with peer groups who can aid in learning how to facilitate agile.

This article is relevant to this research because it provides real-world examples of some of the challenges faced by IT teams as they attempt to implement agile methodologies as well as proven methods of addressing those challenges.

Vijayasarathy, L. & Turk, D. (2012). Drivers of agile software development use: Dialectic interplay between benefits and hindrances. *Information and Software Technology, 54*(2), 137-148. doi:10.1016/j.infsof.2011.08.003

Abstract. *Context*: Agile software development with its emphasis on producing working code through frequent releases, extensive client interactions and iterative development has emerged as an alternative to traditional plan-based software development methods. While a number of case studies have provided insights into the use and consequences of agile, few empirical studies have examined the factors that drive the adoption and use of agile. *Objective*: We draw on intention-based theories and a dialectic perspective to identify factors driving the use of agile practices among adopters of this software development methodology.

Method: Data for the study was gathered through an anonymous online survey of software development professionals. We requested participation from members of a selected list of online discussion groups and received 98 responses.

Results: Our analyses reveal that subjective norm and training play a significant role in influencing software developers' use of agile processes and methods, while perceived benefits and perceived limitations are not primary drivers of agile use among adopters. Interestingly, perceived benefit emerges as a significant predictor of agile use only if adopters face hindrances to their agile practices.

Conclusion: We conclude that research in the adoption of software development innovations should examine the effects of both enabling and detracting factors and the interactions between

them. Since training, subjective norm, and the interplay between perceived benefits and perceived hindrances appear to be key factors influencing the adoption of agile methods, researchers can focus on how to (a) perform training on agile methods more effectively, (b) facilitate the dialog between developers and managers about perceived benefits and hindrances, and (c) capitalize on subjective norm to publicize the benefits of agile methods within an organization. Further, when managing the transition to new software development methods, we recommend that practitioners adapt their strategies and tactics contingent on the extent of perceived hindrances to the change.

Highlights: (a) We surveyed software development professionals about agile adoption and use. (b) Subjective norm and training are positive drivers of agile use. (c) Perceived benefits and limitations do not directly influence agile use. (d) Perceived benefit is a predictor only if adopters face hindrances to using agile. (e) Both enabling and detracting factors and their interaction are important.

Summary. The aim of this study is to contribute to agile methodology theory and practice by identifying key enabling and detracting factors of agile adoption and use. The authors review prior empirical research and find that management behaviors, influential peers, clients, training and perception of benefits all play a significant role in how impactful agile methodologies will be in an organization. The authors collected data through an anonymous online survey of IT professionals who were determined most likely to be adopters of agile development methods. The authors compared results of this survey against several hypotheses regarding existing positive associations with agile use and negative associations with agile hindrances. The survey results revealed some surprising findings and implications.

Findings:

- Perceived challenges play a larger role in agile uptake than do perceived benefits.
- If there are no perceived hindrances to using agile then:
    - Agile use will be strongly and positively related to experience and training. Team members are more open to agile methodologies than they would be if perceptions of hindrances exist.
    - Organization size is moderately and negatively related to agile use.
    - Perceived benefits and limitations are unrelated to agile use. A total absence of hindrances at the beginning of an agile methodology deployment attempt may be an unreal expectation at most organizations and any benefits derived from agile use may be attributed by team members to other, unrelated, factors.
- If there are perceived hindrances to using agile then:
    - Hindrances are moderated by perceived benefits.
    - The level of experience and training of other team members are still strongly and positively related to agile use, though their effects are weaker.
    - Organization size is unrelated to agile use.

Implications:

- Agile use will depend on the presence of support, the approval of key people, and the availability of relevant training to facilitate the transition.
- The more experienced the IT teams are and the larger the organization, the less likely agile will be successful.
- If it appears that there is no opposition, then identifying key influencers within the organization who use or support the use of agile methods will aid adoption.

- Open communication among IT teams about the perceived benefits and hindrances of agile may be helpful to reach consensus on moving forward with agile adoption.
- Smaller organizations may be better suited to readily apply agile methods.
- When hindrances are perceived to exist, the importance of favorable factors such as smaller organizational size, training, and developer experience become less important than they would be if there were no perceptions of hindrances.

Conclusion

Introduction

Project management using agile methodologies is an iterative and incremental method of managing the design, workload, planning, processes, and activities for engineering and information technology initiatives (Campanelli & Parreiras, 2015). It requires teams and individuals from the relevant businesses groups, working in close collaboration with clients and management in time-compressed iterative stages to complete small portions of the deliverable in each delivery cycle (Campanelli & Parreiras, 2015) and, where possible, deploying deliverables as reviewable prototypes to demonstrate progress and to receive client feedback (Holh, Münch, Schneider & Stupperich, 2016). Agile methods often evolve the entire set of deliverables over time based on changing requirements revealed through iterative testing and review. The end result of an agile project is a product that best meets current customer needs that is delivered with minimal cost, waste, and time.

This research presents findings that will provide project managers, project sponsors, IT managers, and IT teams with information they need to determine: (a) which factors may best influence the successful adoption of agile methodologies within their organizations, (b) how the demonstrable effectiveness of agile methodologies may benefit their organizations, (c) which

agile best practices are commonly shared throughout a wide spectrum of industries, (d) what challenges are commonly faced by organizations when attempting to implement agile methods, and (e) strategies for overcoming those challenges.

Agile Effectiveness within an Organization

Traditional research into why IT projects fail has typically focused on the causes of the failure of IT projects such as insufficient funding, poor leadership, or a culture of change resistance within an organization (Chow & Cao, 2008). The Agile Manifesto (Beck et al., 2001) was first introduced 17 years ago as an attempt to address software development issues which were believed to be the underlying causes of so many of the challenges faced by IT teams as they sought to deliver projects. Since that time, research studies that are focused on understanding which factors are most common amongst organizations that report successful and sustained agile deployments number in the thousands (Inayat, et al., 2015). Early research at first focused on technological factors as keys to success (Orlikowski, 1992) before transitioning to human factors as playing a significant role in project success (Dingsoyr, Nerur, Balijepally, & Moe, 2012). Research performed by Tessem (2014) into successful examples of the principles of self-empowerment advocated by agile methods found that individuals who have a sense of control over their daily work "are structurally empowered by being given the ability to initiate and participate in decision-making processes and have rewarding practices that contribute to their sense of meaningfulness" (p. 884).

A decade ago, Chow and Cao (2008) surveyed scholarly literature to identify several critical success factors of agile methodologies. They then administered a survey to agile professionals and analyzed the results to categorize and gauge the effectiveness of critical success factors for agile software development projects. One finding was nineteen separate

failure factors organized into four categories: organizational, people, process, and technical (Chow & Cao, 2008). Inayat et al. (2015) performed a study in 2015 that utilized a narrower categorization method and sought to identify the five top challenges of agile development, which they identified as communication issues, overscoping, requirements validation, requirements documentation, and limited customer involvement.

Other potential failure factors were managerial in nature and include a lack of vision, poor team communications, insufficient client engagement, and unexpected and unplanned for requirements changes (Serrador & Pinto, 2014). Maruping, Venkatech and Agarwal (2009) researched managerial methods that support agile methodology and found a statistically significant correlation between the use of managerial methods that aided agile methodology at the team level and resulting positive impacts on total project quality. Specific managerial methods cited in their research included fostering greater team and individual autonomy, smaller team sizes, and iterative development timeframes (Maruping, Venkatesh & Agarwal, 2009).

Technology-based challenges with agile projects are usually the result of software or hardware not performing as expected, thus driving cost increases and extending delivery times (Ramesh, Cao & Baskerville, 2010). Failures in delivering agile methodologies may also result from nothing more than poor project management practices (Chow & Cao, 2008).

Best Practices for Ensuring a Successful Initial Agile Deployment

Laanti et al (2011) examined the organization-wide adoption of agile methodologies at Nokia, and their results revealed that respondents to their survey of over 1,000 employees reported an overwhelmingly positive experience and attitude towards agile methods and that "60% wanted to stay in an agile mode of working, while only 9% wanted to go back to traditional working methods" (p. 280). Gregory et al. (2015) also focused on those organizations

that have successfully implemented widespread deployments of agile methodology but still had obstacles and impediments to overcome. Through the use of *Challenge Walls* they identified four categories of major challenges to agile deployments and strategies for their remediation: (a) insufficient communication, (b) expectations which are not supported by agile methodology, (c) unwillingness of management to accept lesser control over teams, and (d) reporting expectations for time and budget estimates that are still linked to the expectations of traditional methodologies (Gregory et al., 2015). Gregory et al. (2015) identified the following strategies to remediate these challenges: (a) establish clear communication guidelines, plans, and terms, (b) ensure all parties undergo some form of agile training so expectations are appropriate and reasonable, and (c) before any agile methodology deployment is initiated, ensure it has the full support and buy-in from management and executive teams and that these stakeholders understand that processes may change as their agile environments mature.

Gandomani and Nafchi (2015) attempt to answer the question of how best to deploy agile methods with their agile deployment framework. This framework presents a multi-step process designed to lead a team through the deployment of agile processes, a few at a time, in an iterative fashion (Gandomani & Nafchi, 2015). Teams select one or more agile processes they would like to implement from a process backlog, then introduce and use the new process in their latest sprint. Once the sprint is complete the teams meet to review their successes and failures in implementing the new process, adjust where necessary, and finally adopt it as a team process once no additional refinements are deemed necessary (Gandomani & Nafchi, 2015).

Dikert, Paasivaara, and Lassenius (2016) identified the practices that may be the most effective in supporting agile deployments:

- Iterative agile deployments.

- Management support and buy-in.
- Provide agile training to team members.
- Hiring or appointing agile coaches.
- Employing pilot programs.
- Keeping team goals and organization goals aligned.

In contrast to private companies, government organizations often present unique challenges in the deployment of agile methods such as rigid top-down management structures and an inability to swiftly react to change (Mergel, 2016). Mergel (2016) suggests the following practices as effective approaches when attempting agile deployments in government organizations:

- Agile Leadership – leadership should encourage transparency in communication and collaboration amongst teams.
- Ask the experts – Knowledgeable, non-government employees may be best suited to create requests for proposal (RFPs), design documents, and system requirements for agile projects.
- Open by Default – To reduce duplication of efforts among government agencies, insist all source code is open and shared.

Vijayasarathy and Turk (2012) note that there are several factors that exist within most organizations attempting to move to an agile development methodology that impact the effectiveness of the transition. They note that the importance of perceived benefits is conditional on the presence or lack of perceived hindrances (Vijayasarathy & Turk, 2012). Vijayasarathy and Turk (2012) also assert that organizations should pay close attention to how their agile initiative is perceived. If an agile initiative is perceived poorly, efforts must be made to understand why.

They assert that one of the most effective methods of changing a negative perception to a positive one is training (Vijayasarathy & Turk, 2012).

Another best practice is to use an agile coach (Dikert, Paasivaara, & Lassenius, 2016; Sureshchandra & Shrinivasavadhani, 2008). Agile coaches can perform a variety of functions such as providing mentoring and feedback to teams on their agile practices, delivering ad hoc training, recommending course-corrections, and serving as respected peers or guides (Sureshchandra & Shrinivasavadhani, 2008).

Final Thoughts

The research, articles, and case studies represented in this research represent a broad range of analyses, experience, styles, and scale. For the organizations represented in the case studies in the selected literature, the driver for selecting agile as a management methodology was almost universal: the recognition that traditional IT project management methodologies were not sufficient for managing IT teams and projects in the modern environment (Gregory, et al., 2015). Common success factors identified in the research were wide-ranging:

- Robust communication amongst all interested parties is critical (Gregory, et al., 2015; Inayat, et al., 2015; Sureshchandra & Shrinivasavadhani, 2008; Vijayasarathy & Turk, 2012).
- Communication is better face-to-face, and teams are better when co-located (Gregory, et al., 2015; Inayat, et al., 2015; Sureshchandra & Shrinivasavadhani, 2008).
- Investment in training and coaching pays dividends (Dikert, Paasivaara, & Lassenius, 2016; Inayat, et al., 2015; Sureshchandra & Shrinivasavadhani, 2008).

- Management must understand agile methods if they are to support these methods (Dikert, Paasivaara & Lassenius, 2016; Inayat, et al., 2015; Sureshchandra & Shrinivasavadhani, 2008).
- There is no best one-size-fits-all approach. Organizations must decide upon an agile strategy which works best *for them* (Dikert, Paasivaara & Lassenius, 2016; Sureshchandra & Shrinivasavadhani, 2008).
- Empowered individuals and teams are engaged; success with agile depends upon those individuals and teams to be engaged (Sureshchandra & Shrinivasavadhani, 2008; Vijayasarathy & Turk, 2012).
- Utilize an agile approach to deploy agile methodologies in an iterative and prioritized fashion, measuring success one piece at a time rather than all at once (Dikert, Paasivaara & Lassenius, 2016; Inayat, et al., 2015; Sureshchandra & Shrinivasavadhani, 2008).

While recognizing critical success factors for agile methods is vital to the successful deployment and maintenance of agile within an organization, it is perhaps equally important for an organization to be aware of hindrances and challenges frequently present when undertaking such an initiative (Dikert, Paasivaara & Lassenius, 2016). Some of the most frequently cited challenges are:

- Organizational inertia is a powerful force, and clear strategies to overcome inertia must be developed and maintained (Dikert, Paasivaara & Lassenius, 2016; Gregory et al., 2015; Vijayasarathy & Turk, 2012).
- Expectations of all involved parties are not aligned. Clients, project managers, executive management, IT teams, and all others involved in an agile initiative must

have clear understandings of their roles in an agile development environment and what to expect, and what not to expect, from others (Dikert, Paasivaara & Lassenius, 2016; Sureshchandra & Shrinivasavadhani, 2008; Vijayasarathy & Turk, 2012).

- Mistaking minimal documentation for insufficient requirements (Dikert, Paasivaara & Lassenius, 2016; Vijayasarathy & Turk, 2012).

This annotated bibliography provides historical information, best practices, and challenges associated with agile implementations and can serve as a guide for those project managers, program managers, PMO leaders, team managers, and other stakeholders embarking upon the implementation of agile methods in their organizations.

References

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M.,…Thomas, D. (2001). Manifesto for agile software development. *The agile manifesto.* Retrieved from: http://agilemanifesto.org

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M.,…Thomas, D. (2001). Principles behind the agile manifesto. *The agile manifesto.* Retrieved from: http://agilemanifesto.org/iso/en/principles.html

Campanelli, A., & Parreiras, F. (2015). Agile methods tailoring – A systematic literature review. *Journal of Systems and Software, 110,* 85-110. doi:10.101/j.jss.2015.08.035.

Centers for Public Issues Education (n.d.). Evaluating information sources. *University of Florida.* Retrieved from: http://www.piecenter.com/

Chow, T. & Cao D. (2008). A survey of critical success factors in agile software projects. *Journal of Systems and Software, 81*(6), 961-971. doi:10.1016/j.jss.2007.08.020

Dikert, K., Paasivaara, M. & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software, 119,* 87-108. doi:10.1016/j.jss.2016.06.013

Dingsoyr, T., Nerur, S., Balijepally, V., & Moe, N. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software, 85*(6), 1213-1221. doi:10.1016/j.jss.2012.02.033

Divr, D., Sadeh, A., & Malach-Pines, A. (2006). Projects and project managers: The relationship between project manager's personality, project types and project success. *Project Management Journal, 37*(5), 17-25. Retrieved from http://www.pmi.org/-/media/pmi/documents/public/pdf/learning/pmj/2006-december.pdf#page=38

Fontana, R., Meyer, V., Reinehr, S., & Malucelli, A. (2014). Progressive outcomes: A framework for maturing in agile software development. *Journal of Systems and Software, 102,* 88-108. doi:10.1016/j.jss.2014.12.032

Fontana, R., Fontana, I., Garbuio, P., Reinehr, S., & Malucelli, A. (2014). Processes versus people: How should agile software development maturity be defined? *Journal of Systems and Software, 97,*140-155. doi:10.1016/j.jss.2014.07.030

Gandomani, T., & Nafchi, M. (2015). An empirically-developed framework for agile transition and adoption: A grounded theory approach. *Journal of Systems and Software, 107,* 204-219. doi:10.1016/j.jss.2015.06.006

Gregory, P., Barroca, L., Sharp, H., Deshpande, A., & Taylor, K. (2015). The challenges that challenge: Engaging with agile practitioners' concerns. *Information and Software Technology, 77*(3), 92-104. doi:10.1016/j.infsof.2016.04.006.

Highsmith, J. (2001). *History: The agile manifesto*. Retrieved from http://agilemanifesto.org/history.html

Holh, P., Münch, J., Schneider, K., Stupperich, M. (2016). Forces that prevent agile adoption in the automotive domain. In Abrahamsson, P., Jedlitschka, A., Ngyuen, D., Felderer, M., Amasaki, S., & Mikkonen, T. (Eds), *Product-focused software process improvement* (pp. 468-177). Springer International Publishing. doi:10.1007/978-3-319-49094-6_32

Inayat, I., Salim, S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior, 51,* 915-929. doi:10.1016/j.chb.2014.10.046

Jalali, S. & Wohlin, C. (2011). Global software engineering and agile practices: A systematic review. *Journal of Software: Evolution and Process, 24*(6), 643-659. doi:10.1002/smr.561

Laanti, M., Solo, O., & Abrahamsson, P. (2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology, 53*(3), 276-290. doi:10.1016/j.infsof.2010.11.010

Maruping, L., Venkatesh, V., & Agarwal, R. (2009). A control theory perspective on agile methodology use and changing user requirements. *Information System Research, 20*(3) 377-399. Retrieved from http://www.jstor.org/stable/23015471

Mergel, I. (2016). Agile innovation management in government: A research agenda. *Government Information Quarterly,33*(3), 516-523. doi:10.1016/j.giq.2016.07.004

Orlikowski, W.J. (1992). The duality of technology: Rethinking the concept of technology in organizations. *Organizational Science, 3*(3), 398-427. doi:10.1287/orsc.3.3.398

Panditi, S. (2018). Survey data shows that many companies still are not agile. *Harvard Business Review*. Retrieved from https://hbr.org/sponsored/2018/03/survey-data-shows-that-many-companies-are-still-not-truly-agile

Qumer, A. & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems and Software, 81,* 1899-1919. doi:10.1016/j.jss.2007.12.806

Serrador, P. & Pinto, J. (2014). Does agile work? – A quantitative analysis of agile project success. *International Journal of Project Management, 33*(5), 1040-1051. doi:10.1016/j.ijproman.2015.01.006

Shao, J. & Muller, R. (2011). The development of constructs of program context and program success: A qualitative study. *International Journal of Project Management, 29*(8), 947-959. doi:10.1016/j.iiprogram.2011.02.003

Sureshchandra, K. & Shrinivasavadhani, J. (2008). Moving from waterfall to agile. Proceedings from Agile, 2008: *AGILE '08 Conference.* Toronto, ON, Canada doi:10.1109/Agile.2008.49

Svensson, M. & Host, M. (2015). Introducing an agile process in a software maintenance and evolution organization. *Proceedings on European Conference of Maintenance and Engineering*. pp. 256-264. doi:10.1109/CSMR.2015.33

Tessem, B. (2014). Individual empowerment of agile and non-agile software developers in small teams. *Information and Software Technology*, *56*(8), 873-889. doi:10.1016.j.infsof.2014.02.005.

Vijayasarathy, L. & Turk, D. (2012). Drivers of agile software development use: Dialectic interplay between benefits and hindrances. *Information and Software Technology, 54*(2), 137-148. doi:10.1016/j.infsof.2011.08.003