

Agile Methods on Large Projects in Large Organizations

Brian Hobbs, *School of Management, University of Quebec at Montreal, Canada*

Yvan Petit, *School of Management, University of Quebec at Montreal, Canada*

ABSTRACT ■

Agile methods have taken software development by storm but have been primarily applied to projects in what is referred to as the “agile sweet spot,” which consists of small collocated teams working on small, non-critical, green field, in-house software projects with stable architectures and simple governance rules. These methods are being used more and more on large projects, but little documentation is available in the academic literature. This article investigates the adoption and adaptation of agile methods for use on large projects in large organizations. The empirical study is based first on case studies, followed by a survey to validate and enrich the case study results. The results are somewhat paradoxical in that some features are common to almost all observations, whereas others show extreme variability. The common features include use of Scrum methodology and agile coaches, as well as the non-respect of the agile principle of emergent architecture.

KEYWORDS: agile; scale; Scrum; traditional; flexibility

INTRODUCTION ■

Agile methods have taken software development by storm since the publication of the *Agile Manifesto*. In recent years, agile methods have become highly prevalent in the software industry (Abrahamsson, Conboy, & Xiaofeng, 2009; Dingsøyr, Nerur, Balijepally, & Moe, 2012) and today, it is one of the hottest topics in project management. Project Management Institute has created the PMI Agile Certified Practitioner (ACP)[®] certification, and *Project Management Journal*[®] will be publishing a special issue on the topic in the future.

Although there have been multiple attempts to apply agile principles to non-software projects (Conforto, Salum, Amaral, da Silva, & de Almeida, 2014; Highsmith, 2003; Petit & Levesque, 2015), current research is limited to the field of software development, the field in which agile methods have become prominent; it focuses on two levels of analysis: the individual project and the organizational context in which projects are carried out. The agile literature has focused almost exclusively on the former while almost completely ignoring the latter (Dingsøyr et al., 2012).

The advantages of using agile methodologies identified in the literature include: a working environment that supports creativity and productivity, rapid adaptation to change, and value for the customer because of the improved identification of needs and priorities and faster multiple deliveries of functionalities (Schwaber, 2004; Thomke & Reinertsen, 1998). These advantages are more readily obtained with certain types of projects in certain contexts (Boehm & Turner, 2005). Kruchten (2013) identified what is referred to as the “agile sweet spot,” which consists of small collocated teams working on small, non-critical, green field, in-house software projects with stable architectures and simple governance rules. Most of the writings on agile methods report on situations that are close to the sweet spot, and projects and contexts with the opposite characteristics are examined much less extensively in the literature (Dingsøyr, Fægri, & Itkonen, 2014; Dingsøyr & Moe, 2014; Dingsøyr et al., 2012). Studies of projects outside the sweet spot have revealed that they are much more problematic (Dikert, Paasivaara, & Lassenius, 2016). Leffingwell (2010) showed that there are a number of impediments to the scaling of these practices in large multi-site, multi-customer, and multi-project organizations. During preliminary interviews prior to undertaking this research, the authors discovered several large organizations that had been experimenting with agile methods over a period of five years or more and that these organizations are struggling to scale from a few agile teams to an organization-wide implementation of agile methods. These observations are consistent with the results of a survey of 3,880 participants, in which VersionOne (2016) found that organizations are continuing to scale agile beyond single teams and single projects and that more energy is being put into scaling agile across the enterprise. The overall objective of this research is

Agile Methods on Large Projects in Large Organizations

to fill this gap by examining projects and contexts that are not in the agile sweet spot, specifically large software projects in large organizations.

The research questions at the project level are: *What challenges are encountered when applying agile methods to large multi-team software projects and what practices have been developed to alleviate these challenges?* The literature that has examined these questions has most often taken an approach that contrasts and/or mixes agile and traditional project management approaches (Boehm & Turner, 2003, 2004, 2005; Conforto et al., 2014; Sommer, Hede-gaard, Dukovska-Popovska, & Steger-Jensen, 2015; Špundak, 2014). This research aims to go beyond this somewhat simplistic approach based on a rich description of practice in specific contexts.

At the organizational level, this research examines the implementation of agile methods in large organizations as it has unfolded over time by tracking implementation strategies and successive adaptations. The adoption of agile methods by a large complex organization requires experimentation and adaptation of the methods to the organization's structure, culture, product/service strategy, human resource management policies, customer interfaces, project roles and governance structures, including program and project portfolio management. At the same time, the organizational context is influenced by the implementation of agile methods. The research question at the organizational level is:

How does the context of large complex organizations affect the adaptation and adoption of agile methods and vice versa?

For a complete research report, including the survey instrument see Hobbs and Petit (in press).

Literature Review

Agile Approaches

Agile methodologies are specific approaches, designed in response to the specific challenges of the software

industry (i.e., high uncertainty, short development cycle, no physical deliverable) and used to implement flexibility in the project management process (Agile Alliance, 2001; Lindvall et al., 2004).

Agile approaches are best defined by Scott W. Ambler (2009) as:

"Agile software development is an evolutionary (iterative and incremental) approach which regularly produces high quality software in a cost effective and timely manner via a value driven life-cycle. It is performed in a highly collaborative, disciplined, and self-organizing manner with active stakeholder participation to ensure that the team understands and addresses the changing needs of its stakeholders. Agile software development teams provide repeatable results by adopting just the right amount of ceremony for the situation they face." (p. 6)

Conboy (2009) conceptualizes the agile approach from two related concepts: flexibility and leanness—two concepts inspired from the flexible mass production systems led by the Toyota production system since the 1950s. Conforto, da Silva, Di Felippo, and Kamikawachi (2016) analyzed 59 definitions covering the concept "agile." They propose a comprehensive definition of the agility construct covering: the entity, the event, the degree, the trigger, the purpose, and the circumstances, as follows: *"Agility is the project team's ability to quickly change the project plan as a response to customer or stakeholders' needs, market or technology demands in order to achieve better project and product performance in an innovative and dynamic project environment."* All agile approaches share common values and principles, as stated in the *Agile Manifesto* (Agile Alliance, 2001) and subsequent publications by authors of the *Manifesto* (Highsmith, 2010; Levin, 2012; Schwaber, 2007). Of these methods, Scrum (Schwaber, 2004; Schwaber & Sutherland, 2013) and Extreme Programming (XP) (Beck, 2000), are by far the best-known and most widely used.

Agile and Traditional Project Management Approaches

Agile approaches are often presented in opposition to the more traditional project management principles and practices (Fernandez & Fernandez, 2008). For example, the *Agile Manifesto* (Agile Alliance, 2001) proposes four core values and twelve principles, which are presented in opposition to the traditional software development project values (for example, "Responding to change over following a plan"). This opposition is particularly relevant when agile is introduced in large organizations where established processes and project management practices are already in place. Boehm and Turner (2005) simply question: *"How do you merge agile, lightweight processes with standard industrial processes without killing agility?" (p. 31)*

Some authors recommend a balanced combination of plan-driven traditional project management and an agile approach to managing projects, stating that the flexibility of an agile approach should be balanced with the advantage of a more traditional approach through a risk-based analysis (Al-Zoabi, 2008; Boehm, 2002; Boehm & Turner, 2003, 2004). Anantatmula and Anantatmula (2008) show empirically that such a combined approach may be very valuable for the management of projects in an IT environment. It has also been shown that agile methods could be combined with stage-gates in large organizations (Karlstrom & Runeson, 2005). This might lead to a different combination and alignment of the software development phases with the project management phases. Another option, according to Vinekar, Slinkman, and Nerur (2006), would be to have both the traditional and the agile methodologies co-exist in separate subunits within an ambidextrous organization. The issue then becomes to develop criteria for selecting which method to use on each project (Barlow et al., 2011); adoption of one method over the other would often be based on the organization's

corporate culture (Gruver & Mouser, 2015; Iivari & Iivari, 2011).

Scaling Agility (or not)

Although agile methodologies have provided flexibility in the software development process, most benefits have been achieved with small projects composed of one or a few dedicated self-managed teams (Kruchten, 2013). For some organizations, the issue might become whether or not to adopt agile approaches, considering the size and legacy of their systems and practices. Kruchten attempts to answer this dilemma by proposing some scaling factors, which might distinguish the contexts in which agile might be easier to introduce than others.

There has been very limited research on the topic of large-scale software development projects using agile approaches (Dyba & Dingsøyr, 2008; Razavi & Ahmad, 2014), although a number of specific cases have been investigated, for example, in the oil industry (Grewal & Maurer, 2007); in large software and product development organizations (Gruver & Mouser, 2015; Lindvall et al., 2004; Mahanti, 2016); in regulated environments (Fitzgerald, Stol, O'Sullivan, & O'Brien, 2013); and at BMC Software Infrastructure (Gat, 2006). In these cases, the level of scaling varies significantly (in terms of project size, number of teams, and number of sites). To help categorize the varying levels of scaling, Dingsøyr et al. (2014) suggest a taxonomy of scale of agile software development projects based on the number of teams, where small-scale corresponds to one team, large-scale to two to nine teams, and very large-scale to more than ten teams. Based on this scale, the main focus was on large-scale projects using agile in this research.

Challenges Facing Large Organizations Trying to Implement Agile Approaches

While agile methodology seems suited for small collocated teams in which the customer can be directly involved, there are a number of impediments to the scaling of these practices in large

multisite, multi-customer, and multi-project organizations. At the 2004 University of Southern California Center for Software Engineering (USC-CSE) Annual Research Review, Boehm and Turner (2005) identified some of the challenges to implementing agile processes in traditional development organizations. The result was a collection of change-related challenges and a list of nearly 40 perceived barriers to agile implementation in large organizations. Many of these challenges were related to scope or scale, but some were related to the clash between the agile and traditional cultures; for example, development process conflicts, variability in subsystems developed that might not integrate easily, different life cycles, and difficulty in using agile on legacy systems.

Leffingwell (2007) groups the challenges of scaling agile into two broad categories: (1) those inherent to the methods themselves, "because of the fixed rule bases and assumptions built into the methods" (p. 87); and (2) challenges imposed by the enterprise that "will prevent the successful application of the new methods" (p. 87). Included in the first category are challenges such as: number and size of teams, difficulty in making the customer an integral part of the team, collocation, emergence of the architecture, lack of requirements analysis, and documented specifications. The impediments related to the enterprise would include: clashes with the existing process and project management organizations, existing formalized policies and procedures, corporate culture, fixed schedule/fixed functionality mandates, and people organized by discipline rather than product line. Mahanti (2006) found similar challenges adopting agile practices in large organizations but identified the prime challenge as the integration of agile projects with the project environment's existing processes.

Scaling Frameworks

The literature on scaling agile has been dominated by consultants proposing a

number of frameworks (Agilealliance.org) developed in recent years to facilitate the use of agile principles in larger projects. According to the survey performed by VersionOne (2016), the most commonly used is Scaled Agile Framework (SAFe®), which has evolved from the Rationale Unified Process (Leffingwell, 2015). Scott W. Ambler (2009) initially proposed a scaling model, called Agile Scaling Model (ASM), which evolved into the framework called Disciplined Agile Delivery (DAD) (Scott W. Ambler & Lines, 2014; DAD—Disciplined Agile Delivery, 2015). Under this model, projects can be scaled based on factors such as team size, geographical distribution, regulatory compliance, domain complexity, organizational distribution, technical complexity, organizational complexity, and enterprise discipline. Other recent frameworks include: Large-Scale Scrum (LeSS) (Larman, 2015; Larman & Vodde, 2014), and Nexus (Schwaber, 2015). All these frameworks attempt to preserve the benefits of agile while improving the links to larger organizations (Gruver & Mouser, 2015). Because of their recent introduction on the market, it is very difficult to assess whether these frameworks have indeed properly addressed some of the issues related to scaling agility and how organizations are actually implementing them.

Investigating the Use of Agile in Large Projects and Large Organizations

Little has been reported in the literature on the interplay between agile methods and the accompanying organizational arrangements (Kettunen, 2007). However, Kettunen (2009) suggests that further improvements in software development could be inspired by organization-oriented business concepts, such as concurrent engineering, multi-project management, and being proactive.

At the XP 2010 conference in Trondheim, Norway, practitioners were polled on what they felt should be the most relevant research topic in their fields. The topic of "agile and large projects"

Agile Methods on Large Projects in Large Organizations

came first on the list (Freudenberg & Sharp, 2010). The exercise was repeated in 2013 and Scaling Agility once again came among the top topics of interest (Dingsoyr & Moe, 2013). This is consistent with the assessment of Ågerfalk, Fitzgerald, and Slaughter (2009), who suggested two of the top research topics in flexible and distributed information system (IS) development should be: (1) organizational selection, adoption, and adaptation of agile methods and (2) agility at the organizational level. It is the ambition of this research to fill this gap in the research on these two topics.

Methodology

Little is known about the subjects investigated in this research and for this reason, the study is exploratory. The research was conducted in two phases. During the first phase, qualitative case studies were conducted. The second phase was a survey to confirm and enrich the results of the case studies. The research can, therefore, be said to employ a mixed method (Tashakkori & Teddlie, 1998). A total of 48 substantially complete responses to the survey were received. The respondents were 44 years of age with 5.6 years of experience with agile methods. Their current roles vary a great deal—in decreasing order of prevalence they are: program managers, IT managers, project managers, ScrumMasters, agile coaches, business or product unit managers, portfolio managers, business analysts, product owners, technical leads, and one software factory director. The respondents come from a diversified group of experienced practitioners. A sample of 48 responses is too small to support most multivariate analyses. The results from the case studies and the survey are very consistent. In addition, the survey results quantify the trends observed in the case studies. The consistency between the results from the two phases is indicative of the generalizability of the results. Comparison with another recent survey by VersionOne (2016) (N = 3,880) with several similar questions reveals that respondent

demographics, the descriptions of the implementation of agile methods, the benefits derived for the use of agile methods, and the obstacles to implementation were very similar. The similarities indicate that the 48 responses to the survey in this research are representative of a larger population. Note that the survey in this research contained several questions not included in the VersionOne survey; however, further research with larger sample sizes would be necessary to validate the generalizability of the results presented here.

The Qualitative Case Studies

Large organizations doing several large software projects were investigated. These types of organizations were chosen because they are ideal sites for exploring the use and impact of agile methods outside of the agile sweet spot at both the project and organizational levels. One of the central features of agile methods is the presence of customer representatives in the role of product owner. This is also one of the problematic features in the application of agile methods in large organizations; for this reason, organizations with different types of relationships with customers were investigated. The customers vary in two important dimensions: number of customers and internal versus external customers.

Three case organizations were used to explore both the project and organizational levels of analysis. In each organization, interviews and analyses of documents were the primary sources of data. Interviews were conducted at both the organizational and project levels. The people targeted for organizational level interviews included members of senior management, program and portfolio managers, directors of project management offices, champions of agile methods, and line managers. The people targeted for project level interviews included ScrumMasters, product owners, project managers, analysts, and system architects. An interview guide was prepared based on previous

work (Petit & Besner, 2013), the literature review, and the research questions. The questions asked during the early part of the interview were open-ended. The interview guide provided a list of information that should be collected during the interview, but not a list of questions to be asked directly except as follow-up questions as the interview progressed. The interview guide was tested during several interviews and revised accordingly. Data relative to each level of analysis (project versus organizational) were provided in many of the interviews. The separation by level of analysis was, therefore, carried out during the analysis of the interviews. All interviews were recorded, transcribed, and analyzed using ATLAS.Ti software.

The original research design called for the analysis of three projects composed of three or more development teams in each of three case study organizations. These were carried out in a financial services company, a company producing large complex systems they sell to outside customers, and a public organization. A total of 41 interviews, with an average duration of one hour, were conducted. The interviewees are presented in Table 1.

The analysis of the nine projects in three case study organizations revealed a very high level of variability in the organizations' contexts and histories, the organizational arrangements and role definitions that accompanied the implementation of agile methods, and the management practices that were put in place on each of the projects investigated. The analysis of the three case studies provided a good understanding of these organizations. However, it was not clear to what extent they were representative; in other words, the authors did not feel that the saturation point had been reached in the interviews (DiCicco-Bloom & Crabtree, 2006). For this reason, three additional 90-minute interviews were conducted with key informants in three additional organizations. The three organizations

	Large System		Financial		Public		Total	
	Number of interviews	Average Duration (minutes)	Number of interviews	Average Duration (minutes)	Number of interviews	Average Duration (minutes)	Number of interviews	Average Duration (minutes)
Project Manager	5	65.6	2	64	3	79	10	69.3
Scrum Master	2	41.5	5	66.6	4	52.3	11	56.8
Product Owner	1	68			3	61.7	4	63.3
Executive					3	44.7	3	44.7
PMO or Portfolio Manager	2	46.5	2	74.5			4	60.5
Analysts			1	56			1	56
Line Manager	1	66					1	66
Architect	1	49			6	52.7	7	52.1
Total / average	12	57.25	10	66.6	19	56.9	41	59.4

Table 1: Summary of interviews from nine projects in three organizations.

were selected to be in industries similar to the initial cases (i.e., financial services, large system development, and the public sector). The complementary interviews investigated the level of the organization and one specific large project, for an overall total of twelve projects in six organizations. The interviews were again recorded, transcribed, and analyzed using ATLAS.Ti. In two of the organizations, one key informant was interviewed. In the third organization, two informants participated in one interview. There were many similarities and several significant differences between the six organizations investigated. After examining the interviews, the understanding of the use of agile methods on large software projects in large organizations was deemed sufficient to adequately support the preparation of the survey for phase two, because that saturation point had been reached.

The Survey Instrument

The survey instrument requested the participation of practitioners who were able to provide a high-level description of a software development project with at least three development teams carried out in an organization with at least 2,000 employees. However, two responses with 1,600 and 1,652 employees, respectively, were retained in the

sample. Three development teams are sufficient to produce coordination challenges not found with a single team. The minimum number of three development teams was based on the results of the analysis of the projects in the six organizations that participated in phase one and contact with the agile community of practice over several years, which showed that projects with significantly more than three development teams are relatively rare. Setting the minimum number of employees at 2,000 was somewhat arbitrary, but the researchers are aware that agile methods are used primarily in small organizations and that setting the limit higher would make it difficult to gather sufficient survey data. An organization with 2,000 employees is large enough to have the effects of specialization and formalization that characterize large organizations (Mintzberg, 1979). The survey instrument, which includes 91 questions, is available in the research monograph (Hobbs & Petit, in press). Due to spatial constraints, the full questionnaire could not be included in this article. A summary description of the survey instrument is provided in Table 2.

Data Collection and Analysis

Invitations to participate in the survey were distributed through several channels. The Project Management

Institute provided logistical support for distribution. Invitations were also distributed through two decentralized international networks of groups dedicated to agile methods (Agile Tours and Agile Alliance). These communities organize annual conferences under the brand "Agile Tour," including the agile community of practice in the authors' home city, which collaborated in this research. Invitations were also distributed using social media and the authors' personal networks. A total of 48 substantially complete responses to the survey were received. All of the case study organizations and the majority of the organizations of the survey respondents use agile methods in contexts in which traditional software project management methods are well established. However, a minority of survey respondents reported on contexts in which all software development projects employ agile methods. Respondents in contexts in which both traditional and agile methods are employed responded to the final section of the survey questionnaire. The objective of this section is to investigate the process by which agile methods have been implemented into contexts dominated by traditional methods and investigate how organizations allocate projects to traditional and agile methods. This section of the survey also investigated the impact of the

Agile Methods on Large Projects in Large Organizations

Section 1. Respondent demographics

Section 2. The relevant organizational context:

- Which in most cases is only a part of the organization, a division or subsidiary, for example
- The years when agile methods were used for the first time on small projects and on large projects. From the case studies and a general knowledge of the context it is known that many organizations, but not all, implement agile methods first on small projects.
- The level of maturity in the use of agile methods
- The presence of an agile community of practice
- The total number of projects and the proportion employing agile methods

Section 3. A specific large project that employed agile methods:

- The level of knowledge of agile methods of people involved in this project
- Types of deliverable, scope, and duration
- Level of integration with other systems
- Number and duration of sprints
- Project organization:
 - Number and composition of development teams
 - Other teams or committees
 - Other individuals
- Project initiation, activities related to architecture and to the planning of sprints before the beginning of software development per se, often referred to as sprint zero
- Agile practices employed
- Benefits and disadvantages of agile methods and project performance

Section 4. The transition from traditional methods to agile methods.

From a general knowledge of the field, it is known that most large organizations had well-established traditional project management methods prior to implementing agile methods and that in most cases the two are present. However, some organizations use agile methods exclusively. For these reasons, section four of the survey was completed only by respondents in contexts where both methods are currently in use.

- Decision rules for determining which projects use agile methods
- Performance objectives that motivated use of agile methods
- Organizational objectives that motivated use of agile methods
- Strategy for implementing agile methods, including change management
- Organizational characteristics that supported implementation
- Organizational characteristics that are obstacles to implementation
- Performance outcomes
- Organizational change outcomes
- New and changed organizational roles, particularly for project managers, ScrumMasters, and product owners

Table 2: Survey questionnaire (overview).

introduction of agile methods in organizations with well-established traditional methods. A total of 35 responses to this final section were received; the remaining 13 respondents, or 27%, are in contexts in which all projects are managed using agile methods.

The Case Study Organizations

All of the case study projects developed and/or made significant changes to large complex systems with multiple interconnections to adjacent systems. All six organizations have introduced agile methods into a context in which

well-established traditional project management methods are in place.

The two financial services companies have formalized structures and procedures and a high degree of specialization. The business processes in this type of organization make extensive use of complex computer systems, which are interconnected internally and with other organizations, including customer organizations and suppliers of information. The software projects are all related to the development and/or updating of systems used by operational employees of the company, by employees in

other organizations that are connected to their systems, and by online customers. In this type of organization, internal business units are the project customers from which product owners are drawn. However, as stated in the literature and shown in the results of this research, the role of product owner is problematic. Financial services is an important sector in which large-scale agile projects are found; 26% of the respondents to the survey are from this sector and it is the second most common industry after software reported in the VersionOne (2016) survey.

The two companies that produce large complex systems that are sold to other organizations for use in their internal business processes have very different relationships with their end users. The product is used by multiple independent customer organizations, which are often in competition with each other. In addition, the contractual process through which the product is sold often specifies what is to be delivered in rather precise detail, which introduces rigidities difficult to reconcile with the flexibility of agile methods and the idea of updating and reprioritizing the product backlog continuously. In these two companies, the product owner role is filled by product managers with the authority and the interest in directing product development.

The systems in the two public sector organizations are numerous, large, interconnected internally and with other organizations, many of which are also in the public sector, but not all. The information technologies in these two organizations are very different. One has many large, poorly performing legacy systems based on outdated technology. The tight integration of the systems, the specialization of both people and organizational units, and the activities used to integrate and test all changes before the infrequent releases into production, all create obstacles to the use of agile methods. The organization has a plan to update its databases and systems with a view to making them less tightly integrated and more amenable to more frequent releases.

The other public sector organization has up-to-date technology. The part of the organization investigated is responsible for a shared communication channel facilitating communication between software modules, and the end users are internal to the organization. The challenge in this organization is to respond more quickly and more adequately to priority system development requests.

Improved collaboration between development teams.	59%
Better communication and understanding between developers and end users.	55%
Better collaboration among organizational units.	50%
Improved satisfaction of personnel.	44%
Empowerment of personnel.	39%
Improved software engineering practices.	37%
Better organizational climate.	35%
Increased motivation and commitment of personnel.	33%
Change from command and control to servant leadership.	26%

Table 3: Objectives of implementing agile.

Results

The results presented here are based on the qualitative case studies, complemented by references to results from the survey questionnaires. The information from the qualitative case studies and the information from the survey are very consistent. The consistency indicates that the case study results are representative of trends that go beyond the six case study organizations. However, an accurate measure of the extent to which they are generalizable would require a much larger sample. The results are presented in four groups: (1) motives and strategies for implementing agile methods, (2) the impact of agile implementation on organizational roles, (3) the impact of agile implementation on project initiation, and (4) agile project organizations.

Motives and Strategies for Implementing Agile Methods

Motives for Implementing Agile Methods

The literature on agile methods claims that these methods provide many performance advantages when compared with traditional methods (Dubey, Jain, & Mantri, 2015; Serrador & Pinto, 2015; Sheffield & Lemétayer, 2013; Stoica, Mircea, & Ghilic-Micu, 2013). According to the respondents, the organizations met the performance objectives, which justified the decision to use agile methods.

In addition to the performance objectives (such as improved speed and quality), organizations are also pursuing

organizational change objectives. Both the interviews and survey report that the following organizational change objectives are both used to justify the use of agile methods and realized during implementation. The objectives are presented in decreasing order of mention in Table 3 and are based on Sections 4a, b, and c of the survey.

All the organizational change objectives reported in both the interviews and the survey, with the exception of improved software engineering practices, are inter-related. Together they represent significant changes in the organizational culture, the working climate, and leadership,

Note that in some of the case study organizations and in the survey data, improved software engineering practices is one of the organizational change objectives. This is the case in which the poor performance of traditional methods is one of the justifications for using agile methods. Software engineering practices are not directly related to agile methods; however, the agile principle that “Continuous attention to technical excellence and good design enhances agility” (Agile Alliance, 2001) can be seen to imply a link between software engineering excellence and agile methods. It is reasonable to assume that improved software engineering practices are at least in part responsible for the performance improvements associated with the change. In these situations, agile methods seem to benefit from performance improvements

Agile Methods on Large Projects in Large Organizations

that are at least in part attributable to improvements in software engineering practices that could have been implemented without the implementation of agile methods. It is plausible that the momentum of the change to agile methods, with significant support from upper management, made improvements in software engineering practices easier to implement. It is also plausible that the frequent integration, tests, and demos created an environment that fosters improvements in the quality of the code.

In several of the case studies, the change to agile methods was motivated in part by the demands of young qualified software developers. In labor markets where the demand for qualified developers exceeds the supply, these workers can be considered volunteer labor in that they can choose the place where they work. In these situations, being able to provide a suitable working environment may provide organizations with a competitive advantage in that they are better able to recruit and retain qualified personnel.

Implementation Strategies

The implementation strategies of all six case study organizations and 74% of the organizations of survey respondents started with pilot projects. All of the development teams in the case studies and all but one in the survey had agile coaches. The role of the agile coaches was primarily to support the implementation of Scrum on development teams; however, the implementation strategies have very little else in common. The extent to which agile methods have become well-established in these six organizations varies a great deal. The six organizations are presented in approximately decreasing order of the extent to which agile methods are well established.

One of the organizations experimented with agile methods on small projects in one department between 2007 and 2009 and deployed these methods on all their software development projects between 2010 and 2013.

Agile methods on both large and small software development projects are now well-established throughout the firm. The other five organizations are in the early stages of implementation, particularly for large projects. A second organization has been using agile methods on small projects for five years and just completed one large and very successful project using agile methods. Agile methods are well on their way to becoming well established for both large and small projects in this organization. A third organization is the IT department of a public organization composed of approximately 250 people, where agile was introduced following years of unsuccessful projects for which traditional methods were used. This organization was one of the case studies in which a scaling framework was used—a mixture of Nexus and LeSS. Their goal is to gradually move away from project development to more continuous deliveries using a central databus (i.e., a shared communication channel facilitating communication between software modules). In a fourth organization, agile methods are currently being applied for the first time on pilot projects. The implementation is being sponsored by the CEO and the board of directors in response to an external audit, which showed that both the organization's software development projects and its legacy systems are performing very poorly. A new vice president and a new senior manager have been brought in to manage the implementation process. Contrary to recommendations in the literature (Kruchten, 2013) and the practice in most other organizations, they have selected large projects as the pilots for implementation. The strategy is to show that agile methods can be applied to large projects and to demonstrate the commitment of upper management to the large-scale conversion to agile methods. At the time of the interviews, the pilot projects had been underway for two years. The pilot projects are well along the learning curve that accompanies

change in the organization and in methods, and several indicators show that the performance of the pilot projects is meeting expectations. It seems likely that upper management will continue to support the large-scale conversion to agile methods in this organization. The fifth organization implemented agile methods on one large project between 2008 and 2012. This organization is a small R&D and product development unit of a large multinational firm. In this organization, the project was very successful and might have led to further use of agile methods in this unit, but due to an unrelated reorganization of the multinational firm, several small R&D units were closed, including this one. It is not possible to know what impact this experiment has had if any. In the final organization, several small projects have been completed using agile methods since 2005. The early pilots were the results of local and personal initiatives. Initially, a set of rules was used to determine whether to use agile or traditional methods; however, these rules were not formalized and, at the time of the interviews, the document containing the rules could not be located. In 2007, the CIO declared that all software development projects should use agile methods, but the resistance from the personnel in the IT department resulted in only a limited number of projects using agile methods. Agile methods were only used when a champion of these methods was on the team. People on other teams said that "agile methods are flexible, so we will be flexible and change agile methods to our liking." Since they preferred traditional methods, there was little change. Many projects did analysis sprints, followed by development sprints, which were followed in turn by integration and test sprints. After a long series of sprints, a product could be demonstrated to customers and potentially released. In 2013, a new CIO put an end to the use of agile methods and returned to traditional methods. The case study projects in this organization were all initiated in the period prior to

2013. In the period following the interviews, a new CIO started to support local initiatives to use agile methods, in part because of the poor performance of traditional methods.

It is difficult to draw conclusions from a small sample, however, implementation in only one of the six organizations relied almost entirely upon initiatives by local champions, and this implementation has been unsuccessful over an extended period of time. The survey results also show that bottom-up implementation strategies are used infrequently (14% of responses). This implementation strategy might not lead to large-scale deployment as shown by the final case study. The survey results also indicate considerable variability: 33% of organizations implemented agile methods on large and small projects the same year, whereas the remaining organizations used agile methods on small projects for up to nine years before implementing them on large projects. The survey results also show a great deal of variation in the proportion of projects using agile methods in the organization: from 9% to 100%, with an average of 55%. It is clear that both the strategies for implementing agile methods and the success of implementation vary a great deal.

The Agile Sweet Spot

The fact that most organizations use agile methods on some projects but not all begs the question: What are the characteristics of projects selected for the use of agile methods? Kruchten (2013) identifies the conditions under which agile methods are easier to implement and produce better results more consistently. In several of the case study organizations, choices are being made as to which projects employ agile methods and which do not. During the interviews in these organizations, respondents were asked to identify the characteristics used to select projects for agile methods; none of these organizations had a clear policy regarding which projects should be done using agile or

traditional methods. The survey results report that only 23% of organizations using both agile and traditional methods have a clear policy. It seems that in many cases the choice to use agile methods is influenced, to a considerable extent, by the personal preferences of managers and project personnel.

The Impact of Agile Implementation on Organizational Roles

The implementation of agile methods introduces many significant changes to the roles and responsibilities within the organization.

The Role of Project Manager

All the case studies and, all but one of the survey responses, show that the implementation of agile methods has a significant impact on the role of the project manager; they also show that the impact varies considerably from one organization to the next. The ScrumMasters and the self-organizing teams of the agile method have taken over the project manager's responsibilities for the detailed planning of activities. The changes to the project manager's role in the coordination of development teams show mixed results. In some cases, the coordination between teams is accomplished largely through the Scrum of Scrums, a regular meeting of the ScrumMasters, typically attended by the project manager. Note, however, that in a small number of cases, project managers are doing more to coordinate teams, a variability that merits further investigation. The project manager role is also reported to be more strategic and involve more effort in stakeholder management.

In one of the case study organizations, they experimented with the removal of the project manager role and concluded that, for small projects with only one development team, the project manager role could be filled by the ScrumMaster, but have elected to maintain the role modified for projects with several development teams. However, in another of the case study

organizations, and in 6% of the survey responses, the project manager role has been eliminated. The implementation of agile methods can have a very significant impact on the role of the project manager, but a better understanding of the circumstances under which the project manager role changes and how it changes is needed.

The Role of ScrumMasters

The Scrum methodology was used in all the case study organizations. This is consistent with the dominant position of this methodology in the agile community worldwide (VersionOne, 2016). The ScrumMaster has a very central role in this method. All of the development teams in the case studies and all but one in the survey responses have ScrumMasters. The introduction of ScrumMasters is not problematic because a clear model for the role is well-established (Cohn, 2009; Schwaber, 2004, 2007; Sutherland, Viktorov, Blount, & Pintikov, 2007), resources for training and coaches are readily available, and a labor market of trained ScrumMasters exists. In many organizations, experienced ScrumMasters from small projects are available at the time large projects using agile methods are undertaken. In the case study organizations, the ScrumMasters had been mostly programmers/analysts or team leaders before becoming ScrumMasters. This is a major change, particularly for those who previously had a technical programmer/analyst role. The agile coaches work very closely with those responsible for planning and coordinating the team's work as well as the team processes.

The Role of Product Owners

The archetype of the product owner as presented in the agile literature (Schwaber, 2004, 2007) is a person with both the knowledge and the authority to make changes to the product backlog and reprioritize at the beginning in each sprint. In this archetype, the product owner is available to development

Agile Methods on Large Projects in Large Organizations

teams, often on a full-time basis. In the reality of large-scale agile projects, the product owner does not always have the knowledge, authority, and availability portrayed in this archetype. The product owner role is problematic. Both the case study and the survey results show that the lack of understanding of this new role is the biggest challenge for product owners. In both the case studies and survey results, the level of knowledge of agile methods and the support for implementation are lower for business unit managers than for all other groups, including upper management.

The product owner needs to be available, knowledgeable of business needs, and have the authority to make decisions. All three of these characteristics pose problems. Making people with these characteristics available to development teams means making them less available for other roles within the business unit, which creates tension around the issue of their availability. In one of the case study organizations, the business units appoint people with very limited availability, business knowledge, and decision-making authority, which in turn causes delays in decisions and many referrals back to those in authority in the business unit, and results in additional delays. This is quite a hierarchical organization in which managers tend not to delegate very much. The role of product owner in agile methods is out of sync with this organization's culture. In this case, the difficulties with the product owner role brought major differences between the agile culture and the host organization's culture to the forefront.

The product owner represents the customer or the end user organization. A key part of this role is the identification and prioritization of the features in the product backlog at the end of each sprint. In traditional development there is no mechanism or incentive to make frequent and precise adjustments to priorities, but with agile methods these take place at the end of each sprint. The product owner role is radically different from the role of the customer

organization in traditional development methods in which the customer specifies requirements and approves the scope at the outset, approves the project at milestones, makes change requests, and accepts the final product at project closure and commissioning. In traditional development, meetings between customer representatives and project representatives are infrequent and structured. The customer representative does not meet the members of the development team. In agile methods, the product owners are members of the project team who meet with the Scrum-Masters and development teams very frequently, at a minimum at the end of each sprint. In many cases, the product owners are full-time members of development teams.

Both the case studies and the survey show that a lack of decision-making authority is also a significant issue for a minority of product owners, and that most product owners need to address questions to the customer organization. Questions are submitted once a week on average and typically receive a reply within a few days; in a minority of organizations, however, the response time is much longer. In one of the case study organizations, a committee of managers from the business unit was set up to respond to such questions.

In cases in which the client is an internal business unit that will use the system in its internal operations, the product owner must know the business and is normally recruited internally. However, the role is different in situations in which the system is sold to organizations or people outside the company. In the two case study organizations in which the customer is external to the organization, the product owner role is filled by a product manager. There are challenges in this role, particularly when a well-specified product has been sold under contract or when a high profile customer makes very specific requests. In two of the case study organizations, the use of agile methods that require the product owner

to update, groom, and reprioritize the product backlog had the effect of making the product manager much more responsive to requests for clear direction than had previously been the case.

In the case of systems sold outside the firm, there are additional problems. The marketing and product management departments have both the contact with and the knowledge of customers and markets. The ways in which these departments function traditionally is not in sync with agile methods, which are very flexible and refine priorities at the end of each sprint and frequently produce potential releases. Changing the functioning of the marketing and product management roles to be in better sync with the agile development methods is a major organizational change, the detailed study of which is outside the scope of this research and, to the best of our knowledge, has not been researched. One of the case study organizations is examining what this might entail; they refer to it as "agile end-to-end," but it has yet to be implemented. A significant part of the problem is the fact that systems are sold under contract and that these contracts specify the scope and features of the systems to be developed and delivered. The question as to how to use the flexibility of agile to deliver systems specified by contract remains largely unanswered. The answer may lie with different types of contracts, but this would mean restructuring the industrial sector, which is neither simple nor without consequences.

In two of the case study organizations, there is a product owner on each development team who was responsible for answering team member questions and for preparing the user stories for the upcoming sprints. Most survey respondents (98%) report having product owners on development teams but, in 32% of the cases, they were part-time employees. In these case studies and in 60% of the survey responses, hierarchical relationships exist between product owners on the development team and a product

manager or “chief product owner.” The latter is the person to whom the product owners on each development team turn for direction and advice. The product owner teams meet regularly to address issues and establish priorities.

In four case studies, there was not a product owner on each team. In three cases, an effective solution was adopted, but in one case no effective solution was found. In two effective cases, a product manager with authority and interest filled the role. In the other effective case, there was a very high level of commitment from the customer organization and very good collaboration between the customer organization and the project organization to the point that they referred to themselves as a team. In the ineffective case, the product owner had insufficient knowledge, availability, and decision-making authority. In this case, the implementation of agile was unsuccessful.

The Role of Analysts

The roles of functional and system analysts are rare in descriptions of agile methods in the literature (Hoda, Noble, & Marshall, 2013; Omar, Syed-Abdullah, & Yasin, 2011). Two of the case study organizations took radically different approaches to the roles of analysts. In one organization the roles have been eliminated. Another organization has an analyst on each development team. The other case study organizations have analysts in the project organization, but not on the development teams. The survey responses indicate that 30% of development teams have functional analysts as team members. In cases in which the ScrumMaster was previously a system analyst, he or she is able to bring this expertise to bear on the project. The question of the roles of analysts remain largely unresolved.

The Role of Architects

All of the case study organizations had architects in the project organization, but not on the development teams. In most cases, an architecture team was

part of the project organization. The survey respondents indicated that 53% of their development teams have architects as members, at least part-time, and that 50% of their project organizations have architecture teams. The projects with both an architecture team and architects on their development teams tend to have architects on the teams on a part-time basis. Architects on the development teams are also members of the architecture teams where these exist. In the case study organizations, functional analysts on development teams are also members of a team led by a system architect. The roles of architects and system analysts begs the question: How is the system architecture developed on large agile projects? This is the subject of the next section.

The Impact of Agile Implementation on Project Initiation

Sprint Zero

The expression “sprint zero” is often used to identify the early activities of an agile project before coding begins. During the interviews, several respondents used the term, but the activities to which they referred varied considerably from one person to another even in the same organization. Seventy-three percent of the survey respondents reported using the term. Table 4 summarizes the list of activities that interviewees included in the expression iteration zero, followed by the percentage of survey respondents who include these activities in the sprint zero. This pattern is very similar to that observed in the case studies.

The planning of sprints	71%
The production of a summary high-level architecture	66%
The writing of user stories for the first sprints	66%
The creation of the product backlog	60%
The creation of epics	51%

Table 4: Activities included in the sprint zero.

All of these activities would need to be done early in the project, but most respondents explicitly exclude some of them from what they consider to be the sprint zero. There is obviously a semantics problem, as the respondents who use the expression “sprint zero” do not agree on its meaning. All of the case study projects and 89% of the survey respondents reported devoting time to these activities prior to the beginning of development per se. An examination of the projects that did not devote time to these activities prior to the beginning of development per se revealed that they were smaller projects and projects considered to be emergencies. Although there is a clear pattern for the majority of projects, there is considerable variation.

System Architecture and Front End Planning

One of the agile principles (Agile Alliance, 2001) is: “The best architectures, requirements, and designs emerge from self-organizing teams.” However, activities related to system architecture before the beginning of software development per se were observed in all the case studies and in all of the survey responses, except for three, which stands in stark contrast to the agile principle of emergent architecture. All interviewees reported that the time and effort invested in architecture and front end planning prior to the beginning of development were significantly less than those with traditional methods. On average, survey respondents reported devoting half the time employed with traditional methods. However, there was considerable variability in responses, with some survey respondents reporting no difference, whereas others reported very significant differences. The fact that some survey respondents reported no difference is intriguing and deserves further investigation.

Interviewees were asked what proportion of the sprints was planned before development began and the responses varied a great deal. In one

Agile Methods on Large Projects in Large Organizations

case study organization all of the sprints were planned in details up front. It should be noted that this case study organization had a strong culture, which required that projects be well defined prior to approval. In other organizations, only a very small number of sprints were planned in advance. The survey respondents report that, on average, 34% of sprints were planned up front, but showed the same variability, with responses ranging from 0% to 100%. This may be an indication that, in some contexts the management of projects has much in common with traditional methods that plan the entire project in detail before starting execution in a plan-driven approach. The survey respondents report that, on average, 57% of the features actually delivered were in the product backlog at the outset. There is again considerable variation among the responses.

The scope is defined in detail in user stories, which are typically not produced far in advance. All the case study organizations and 79% of survey respondents report that user stories are prepared three sprints in advance or less. The most common pattern is therefore to produce a summary architecture, to plan one third of sprints, and to define more than half of the product backlog prior to beginning development per se and to produce user stories one to three sprints in advance during project execution. There is, however, a great deal of variation

Integration with Other Systems

A prominent feature of the case study interviews was the frequent mention of issues related to the integration of the system being developed or modified with many other systems within the organization and with systems external to the organization. Likewise, survey respondents report that their projects are integrated with an average of ten other systems. Interviewees report that the integration with other systems prevented the frequent deliveries of software features to users because of the integration

and testing required and because of organizational policies that limited releases to a few times a year, violating the agile principle of frequent releases (Agile Alliance, 2001). Fifty percent of survey respondents report that organizational or technical constraints delayed the deployment of functionalities ready for use. There was, however, a great deal of variation in the duration of delays; 11% of respondents reported delays in excess of one year, whereas the remaining report an average delay of 3.6 weeks.

The project teams in the case study organizations are organized to deal with the integration issue. The primary strategy is to have human resources available to the team that are experts in the integration with the other systems. This reduces the impact of testing and integration with other systems somewhat but, despite these measures, the delays in delivery remain significant. In several of the interviews, the impact of the organization's technology was brought up as an explanation for the impossibility to remove this barrier to the frequent releases. Two of the case study organizations are changing their database structures to make the systems less interdependent and to allow more frequent releases; this, however, is a very expensive and long-term goal.

Agile Project Organizations

Project Organizations

All the case study projects have rather large project structures. The case study projects have four development teams, on average, whereas the projects described in the survey are larger—with an average of nine development teams. In addition to the development teams, the project organizations had many other individual members and other committees or teams. On average, 94 people worked on the projects in the survey during the most intensive period (full-time equivalent). As stated earlier, most projects have resources within the project structures and outside these structures to deal with the issue of integration with other systems

and most, but not all, projects had project managers. A small proportion of the committees, such as the steering committees and client acceptance committees, have the role of coordinating with the rest of the organization. But the majority of the teams, committees, and individual members play roles in the coordination of the work of the development teams. Specialized personnel on the development teams tend to also be members of the corresponding specialist teams where the coordination within their area of specialization takes place. One of the most common features of the project organizations is the Scrum of Scrums in which the ScrumMasters meet regularly with other participants, including the project manager. Four of the six case study organizations have such teams, and 75% of the projects described in the survey do as well. Although there is great variety in the particular structures set up to manage each project, there is a common pattern of having a large number of teams or committees and individuals in specialist roles, in addition to the development teams.

Development Team Composition

The recommended size of development teams is between five and nine members (Dingsøyr et al., 2014; Schwaber, 2004; Schwaber & Sutherland, 2013). However, one of the case study projects and 16% of the projects described in the survey have larger development teams: between 10 and 15 members according to survey results. All of the case study projects have a full-time ScrumMaster on each team, whereas 98% of the development teams described in the survey have ScrumMasters, some of whom were part-time. All of the case study development teams and 96% of the teams described in the survey have daily stand-up meetings, which is a feature of the Scrum methodology. There is an effort to maintain stable team composition in all the case study projects and in 89% of the organizations described in the survey.

All members of each of the case study development teams are collocated and worked in open space environments, whereas 54% of the teams described in the survey do as well. The teams in five of the six case study organizations are located geographically within one kilometer (approximately one half of a mile) of each other and, in almost all cases, in the same building. The development teams described in the survey are much more dispersed geographically: 60% have members in different time zones, which is consistent with the VersionOne (2016) survey results. The development teams in the case study organizations and in the survey have much in common. The differences can largely be explained by the differences in the data collection methods, between interviews in organizations in the authors' home city, and an online survey.

Discussion

The results from the case studies and from the survey are very consistent. In most cases, the results are very similar, whereas in a few cases the survey adds details and perspectives to the more limited number of projects and organizations in the case studies. The results are somewhat paradoxical in that some features are common to almost all observations, whereas others show extreme variability. The common elements are:

- Scrum is the dominant methodology and almost all projects have ScrumMasters who have daily stand-up meetings. Agile coaches are used to support ScrumMasters and development teams;
- Implementation strategies almost always start with pilot projects, which are almost always small and simple projects;
- The front ends of the projects are remarkably similar. The activities of the front end include the planning of sprints, the production of a summary architecture, the writing of epics and

user stories, and the creation of the product backlog. The time and effort devoted to these activities are significantly less than those with traditional methods;

- Approximately two-thirds of the scope is defined at the outset in the product backlog, but the detail is provided only one or two sprints in advance in user stories; and
- The project organizations are quite large. In addition to the development teams, the project organizations have several other teams, committees, and individuals with specialized expertise who are responsible for the coordination of the work of the development teams and the technical excellence of their respective areas of specialization.

Within this common pattern, there is considerable variability. Some of the most significant areas of variation are:

- Despite a few common features, the implementation strategies vary significantly;
- Despite the similarities found in the front ends of projects, the meaning of the term most commonly used to identify the front end "sprint zero" varies a great deal even within organizations;
- The extent of front-end planning of sprints varies from almost no planning to the planning of all the sprints prior to approval and initiation; and
- There is extreme variability in the details of the project organizations.

Both common patterns and the objects of great variability are important results; from a research perspective both identify opportunities for further research. Because this is an exploratory study of a subject that has received little attention in the research literature, additional studies are needed to confirm both common patterns and the objects of variability. Further research should investigate the reasons for both. From a practitioner perspective, it is important to know the choices other organizations

have made after some experimentation with agile methods on large projects so as to benefit from their experience.

Returning to our research questions:

- At the project level is: What challenges are encountered when applying agile methods to large multi-team software projects and what practices have been developed to alleviate these challenges?
- At the organizational level is: How does the context of large complex organizations affect the adaptation and adoption of agile methods and vice versa?

Three levels of analysis can be distinguished in the results: the development team, the project, and the interaction between the project and the organization.

The Development Team Level

The investigation of development teams is not the focus of this research, because agile team functioning is addressed abundantly in the literature. For this reason, the literature review did not include this topic; however, interviewees spontaneously provided detailed descriptions of their development teams and several questions related to the topic are included in the survey instrument. The examination of this information has led to the observation that the internal functioning of agile development teams is very similar regardless of the number of development teams in the project. Agile methods are based on self-organized teams in which there is limited specialization among the team members. In a few cases, there was tension between the high level of specialization in the large bureaucratic organizations and the flexibility of agile teams, but this was managed locally within the teams and did not present a major challenge. Agile team functioning is not addressed further here.

The Project Level

The differences between small and large projects are more at the project level. Projects with only one development

Agile Methods on Large Projects in Large Organizations

team typically have a limited number of other human resources in the project organization outside the development team; these are typically a product owner; technical specialists such as architects, analysts, testers, and documentarists. Single development team projects often do not have a project manager because this role is filled by the ScrumMaster. Larger projects are faced with three organizational challenges: the coordination of multiple development teams, the organization of a greater number of specialists outside the development teams, and the integration with other systems. The three are interrelated.

Larger projects have multiple teams or committees of specialists, including the meeting of ScrumMasters in Scrums of Scrums. When there are product owners on development teams, they typically meet on a product owner team with a more senior product owner or product manager. When specialists are full-time or part-time members of development teams, they typically meet on specialist teams where they deal with both issues related to their area of specialization and the coordination of the development teams as it relates to their specialization. The administrative coordination takes place in the Scrums of Scrums, usually attended by the project manager. Frequent demos, testing, integration, and retrospectives are at the heart of agile methods because they provide additional opportunities for the coordination among development teams and across areas of specialization.

Large projects also create challenges for the project manager role. Most agree that the impact is significant, but it is difficult to see a generalizable pattern in the adjustments taking place. In some situations, project managers devote more effort to the coordination of development teams, but in others less. In a minority of cases, the project manager role has been abolished. Most agree, however, that cases in which the project manager role has been maintained, it has become more strategic

and is focused more on stakeholder management. It is too early in the evolution of the management of large agile projects to come to a definite conclusion on this subject.

Interaction between the Project and the Organization

Most other challenges are related to the interaction between the projects and the organization. The implementation of agile methods in a large organization with well-established traditional methods is a significant organizational change. As with any significant change, management support is critical (Young & Poon, 2013). The acceptance and support of IT and business unit managers as well as development personnel are also very important. Being knowledgeable of agile methods is a necessary condition for support.

One of the biggest challenges is the relationship between the project and the client organization, which is related to the challenges with the role of product owner. The transition from traditional to agile methods is a radical change. Business unit managers are, on average, less knowledgeable and less supportive of agile methods. Agile methods require them to prioritize their needs much better, be more involved in projects, commit significant resources, and respond more quickly to requests for clarification of business rules and priorities. They also create ambiguity as to the exact nature and scope of the project deliverable. Two elements are keys to the creation of an effective working relationship between the business unit and the project; first, the creation of a partnership between the customer organization and the project based on a clear understanding of each other's context and needs; and second, the role of product owner. One cannot be effective without the other.

There are conflicts between large traditional organizations and agile principles that are related to structures, processes, and culture (Iivari & Iivari, 2011). One of the biggest challenges is the difference in role definitions between the

two. The strategy organizations have adopted to dealing with all of these problems is to use pilot projects to isolate the problems and to develop new processes and new role definitions. The project approval process in traditional organizations that requires project parameters be well-defined in advance is another significant challenge—this is both a process and a cultural issue. For internal projects, the solution has again been negotiating special arrangements on pilot projects. For projects with contracts with external customers, the issue has yet to be resolved, specifically in relation to scope definition. A change in leadership style, from command and control to styles approaching servant leadership, is again a very significant change organizations are addressing on pilot projects (Greenleaf, 2002). It is too early in the evolution of these changes to know how they will develop in the future.

Conclusion

Agile methods are being used more and more on large software projects in large organizations. The use of agile methods in this context requires significant adaptations at both the project and organizational levels. There are fundamental contradictions between large bureaucratic organizations with well-established traditional software development methods and agile principles and practices. The resolution of these contradictions is currently ongoing but is still incomplete. Some aspects of the forms these will take in the future are already quite clear, but others will only become clear as the evolutionary process progresses. This is a subject that will require further research as the adoption and adaptation processes progress.

Future Research

Some of the most important outstanding questions are:

- What will the project manager's role be in the future? Will there be project managers in this type of context?

Will the answers to these questions be contingent upon yet to be unidentified conditions?

- To date, the effect of the use of agile methods has been primarily at the project level, with little or no reconceptualization at the program or portfolio level. However, scaling frameworks that propose solutions at these levels are gaining in popularity. How will agile methods affect program and portfolio management?
- Projects with internal business units as customers are in a situation quite different from those with external customers under contract:
 - For internal projects, how will the relationship between business units and projects evolve and will this fundamentally change the entire organization?
 - For external projects, how will the use of agile methods for systems development effect the entire organization, with redefinitions of the roles of other departments and the relationships with customers, in what might become “agile end-to-end?”

References

- Abrahamsson, P., Conboy, K., & Xiaofeng, W. (2009). ‘Lots done, more to do’: The current state of agile systems development research. *European Journal of Information Systems*, 18, 281–284.
- Ågerfalk, P. J., Fitzgerald, B., & Slaughter, S. A. (2009). Introduction to the special issue: Flexible and distributed information systems development: State of the art and research challenges. *Information Systems Research*, 20(3), 317–328.
- Agile Alliance. (2001). *Manifesto for agile software development*. Retrieved from <http://agilemanifesto.org/>
- Al-Zoabi, Z. (2008). *Introducing discipline to Xp: Applying Prince2 on Xp projects*. Paper presented at the Information and Communication Technologies: From Theory to Applications, 2008, ICTTA 2008, 3rd International Conference.
- Ambler, S. W. (2009). *The agile scaling model (ASM): Adapting agile methods for complex environments*. Retrieved from https://www.researchgate.net/profile/Scott_Ambler/publication/268424579_Adapting_Agile_Methods_for_Complex_The_Agile_Scaling_Model_ASM_Adapting_Agile_Methods_for_Complex_Environments/links/55003e780cf28e4ac347ee34.pdf?origin=publication_detail
- Ambler, S. W., & Lines, M. (2014). *Scaling agile software development disciplined agility at scale*. Retrieved from disciplinedagileconsortium.org
- Anantatmula, V. S., & Anantatmula, M. (2008). *Use of agile methodology for IT consulting projects*. Paper presented at the PMI Research Conference, Warsaw, Poland.
- Barlow, J. B., Keith, M. J., Wilson, D. W., Schuetzler, R. M., Lowry, P. B., Vance, A., & Giboney, J. S. (2011). Overview and guidance on agile development in large organizations. *Communications of the Association for Information Systems*, 29, 25–44.
- Beck, K. (2000). *Extreme programming explained: Embrace change*. Boston, MA: Addison-Wesley.
- Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, 35(1), 64–69.
- Boehm, B., & Turner, R. (2003). *Observations on balancing discipline and agility*. Paper presented at the Agile Development Conference, 2003.
- Boehm, B., & Turner, R. (2004). *Balancing agility and discipline: A guide for the perplexed*. Boston, MA: Addison-Wesley.
- Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, 22(5), 30–39.
- Cohn, M. (2009). *Succeeding with agile: Software development using Scrum*. London, England: Pearson Education.
- Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, 20(3), 329–354.
- Conforto, E. C., Amaral, D. C., da Silva, S. L., Di Felippo, A., & Kamikawachi, D. S. L. (2016). The agility construct on project management theory. *International Journal of Project Management*, 34(4), 660–674.
- Conforto, E. C., Salum, F., Amaral, D. C., da Silva, S. L., & de Almeida, L. F. M. (2014). Can agile project management be adopted by industries other than software development? *Project Management Journal*, 45(3), 21–34.
- DiCicco-Bloom, B., & Crabtree, B. F. (2006). The qualitative research interview. *Medical Education*, 40(4), 314–321.
- Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems & Software*, 119, 87–108.
- Dingsøyr, T., & Moe, N. B. (2013). *Research challenges in large-scale agile software development*. Paper presented at the ACM SIGSOFT Software Engineering Notes.
- Dingsøyr, T., Fægri, T. E., & Itkonen, J. (2014). What is large in large-scale? A taxonomy of scale for agile software development. *Product-focused software process improvement* (Vol. 8892 in the series *Lecture Notes in Computer Science*, pp. 273–276). New York, NY: Springer.
- Dingsøyr, T., & Moe, N. B. (2014). Towards principles of large-scale agile development: A summary of the workshop at Xp2014 and a revised research agenda, *Agile methods: Large-scale development, refactoring, testing, and estimation* (Vol. 199 in the series *Lecture Notes in Business Information Processing*, pp. 1–8). New York, NY: Springer.
- Dingsøyr, T., Nerur, S., Baliyepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213–1221.
- Disciplined Agile Delivery (DAD). (2015). Retrieved from <http://www.disciplinedagiledelivery.com>

Agile Methods on Large Projects in Large Organizations

- Dubey, A., Jain, A., & Mantri, A. (2015).** Comparative study: Waterfall v/s agile model. *International Journal of Engineering Sciences & Research Technology*, 4(3), March.
- Dyba, T., & Dingsøyr, T. (2008).** Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50, 833–859.
- Fernandez, D. J., & Fernandez, J. D. (2008).** Agile project management: Agilism versus traditional approaches. *The Journal of Computer Information Systems*, 49(2), 10–17.
- Fitzgerald, B., Stol, K.-J., O'Sullivan, R., & O'Brien, D. (2013).** *Scaling agile methods to regulated environments: An industry case study*. Paper presented at the Proceedings of the 2013 International Conference on Software Engineering.
- Freudenberg, S., & Sharp, H. (2010).** The top 10 burning research questions from practitioners. *Software, IEEE*, 27(5), 8–9.
- Gat, I. (2006).** *How BMC is scaling agile development*. Paper presented at the IEEE Agile 2006 Conference.
- Greenleaf, R. K. (2002).** *Servant leadership: A journey into the nature of legitimate power and greatness* (25th anniversary ed.). Costa Mesa, CA: Paulist Press.
- Grewal, H., & Maurer, F. (2007).** *Scaling agile methodologies for developing a production accounting system for the oil & gas industry*. Paper presented at the Agile Conference (AGILE), 2007.
- Gruver, G., & Mouser, T. (2015).** *Leading the transformation: Applying agile and devops principles at scale*. Portland, OR: IT Revolution.
- Highsmith, J. A. (2003).** Cutter consortium reports: Agile project management: Principles and tools. *Cutter Consortium*, 4(2), 2–4.
- Highsmith, J. A. (2010).** *Agile project management: Creating innovative products*. Upper Saddle River, NJ: Addison-Wesley.
- Hobbs, B., & Petit, Y. Agile methods on large projects in large organizations.** Newtown Square, PA: Project Management Institute, Newtown Square, PA (in press).
- Hoda, R., Noble, J., & Marshall, S. (2013).** Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering*, 39(3), 422–444.
- Iivari, J., & Iivari, N. (2011).** The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology*, 53(5), 509–520.
- Karlstrom, D., & Runeson, P. (2005).** Combining agile methods with stage-gate project management. *IEEE Software*, 22(3), 43–49.
- Kettunen, P. (2007).** Extending software project agility with new product development enterprise agility. *Software Process: Improvement and Practice*, 12(6), 541–548.
- Kettunen, P. (2009).** Adopting key lessons from agile manufacturing to agile software product development: A comparative study. *Technovation*, 29(6/7), 408.
- Kruchten, P. (2013).** Contextualizing agile software development. *Journal of Software Maintenance and Evolution*, 25(4), 351–361.
- Larman, C. (2015).** Less. Retrieved from <http://less.works/>
- Larman, C., & Vodde, B. (2014).** *Large-scale Scrum: More with less*. Boston, MA: Addison Wesley Professional.
- Leffingwell, D. (2007).** *Scaling software agility: Best practices for large enterprises*. New York, NY: Pearson Education.
- Leffingwell, D. (2010).** *Agile software requirements: Lean requirements practices for teams, programs, and the enterprise*. Upper Saddle River, NJ: Addison-Wesley.
- Leffingwell, D. (2015).** *Safe—Scaled Agile Framework*. Retrieved from <http://www.scaledagileframework.com/>
- Levin, G. (2012).** *Program management: A life cycle approach*. Boca Raton, FL: Auerbach Publications.
- Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., . . . Kahkonen, T. (2004).** Agile software development in large organizations. *Computer*, 37(12), 26–34.
- Mahanti, A. (2006).** Challenges in enterprise adoption of agile methods—A survey. *Journal of Computing and Information Technology*, 14(3), 197–206.
- Mintzberg, H. (1979).** *The structuring of organizations: A synthesis of the research*. Englewood Cliffs, NJ: Prentice-Hall.
- Omar, M., Syed-Abdullah, S.-L., & Yasin, A. (2011).** The impact of agile approach on software engineering teams. *American Journal of Economics and Business Administration*, 3(1), 12–17.
- Petit, Y., & Levesque, M.-M. (2015).** *Assessing the application of agile principles in non-IT projects*. Paper presented at the IRNOP, London, England.
- Razavi, A. M., & Ahmad, R. (2014).** *Agile development in large and distributed environments: A systematic literature review on organizational, managerial and cultural aspects*. Paper presented at the 8th Malaysian Software Engineering Conference (MySEC), Kuala Lumpur, Malaysia.
- Schwaber, K. (2004).** *Agile project management with Scrum*. Redmond, WA: Microsoft Press.
- Schwaber, K. (2007).** *The enterprise and scrum*. Redmond, WA: Microsoft Press.
- Schwaber, K. (2015).** *Nexus*. Retrieved from <https://www.scrum.org/Resources/The-Nexus-Guide>
- Schwaber, K., & Sutherland, J. (2013).** *The Scrum guide™: The definitive guide to Scrum: The rules of the game.*, Retrieved from <http://www.scrumguides.org/>
- Serrador, P., & Pinto, J. K. (2015).** Does agile work? A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040–1051.
- Sheffield, J., & Lemétayer, J. (2013).** Factors associated with the software development agility of successful projects. *International Journal of Project Management*, 31(3), 459–472.

Sommer, A. F., Hedegaard, C., Dukovska-Popovska, I., & Steger-Jensen, K. (2015). Improved product development performance through agile/stage-gate hybrids. *Research Technology Management*, 58(1), 34–44.

Špundak, M. (2014). Mixed agile/traditional project management methodology: Reality or illusion? *Procedia—Social and Behavioral Sciences*, 119(0), 939–948.

Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software development: Agile vs. traditional. *Informatica Economica*, 17(4), 64–76.

Sutherland, J., Viktorov, A., Blount, J., & Pintikov, N. (2007). *Distributed Scrum: Agile project management with outsourced development teams*. Paper presented at the Hawaii International Conference on System Sciences, Hawaii.

Tashakkori, A., & Teddlie, C. (1998). *Mixed methodology: Combining qualitative and quantitative approaches* (Vol. 46). Thousand Oaks, CA: Sage.

Thomke, S., & Reinertsen, D. (1998). Agile product development: Managing

development flexibility in uncertain environments. *California Management Review*, 41(1), 8–30.

VersionOne. (2016). *10th Annual State of Agile Report*. Retrieved from <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>

Vinekar, V., Slinkman, C. W., & Nerur, S. (2006). Can agile and traditional systems development approaches coexist? An ambidextrous view. *Information Systems Management*, 23(3), 31–42.

Young, R., & Poon, S. (2013). Top management support—Almost always necessary and sometimes sufficient for success: Findings from a fuzzy set analysis. *International Journal of Project Management*, 31(7), 943–957.

Brian Hobbs, PhD, PMP, has been a Professor at the School of Management of the University of Quebec at Montreal, Canada, in the Master's Program in Project Management for more than thirty years; this program, of which he is a past director, is accredited by PMI's Global Accreditation Center. He founded the Project Management Research Chair at the University of Quebec in Montreal in 2007 and

held the Chair until 2015. Professor Hobbs holds a degree in Industrial Engineering, an MBA, and a PhD in Management and has served terms on both PMI's Standards and Research Members Advisory Groups and is currently a member of the PMI-Montreal Board of Governors. He received the 2012 PMI Research Achievement Award and, along with his colleague, Monique Aubry, received the 2012 International Project Management Association Research Award for their work on PMOs. In 2013, he received the Research Career Achievement Award from the School of Management and in 2015 became a PMI Fellow. He can be contacted at hobbs.brian@uqam.ca

Yvan Petit, MEng, MBA (Insead), PhD, PMP, PfMP, has been an Associate Professor at the University of Quebec at Montreal (ESG UQAM), Canada, since 2010. His research interests are portfolio management, agile methods, and uncertainty management and he has over 25 years of experience in project management, primarily in software development and R&D in the telecommunications industry. Professor Petit has been a member of the Canadian committee on the ISO TC-258 on Project Portfolio Management and is now a member of the PMI Standards MAG (Member Advisory Group) and is the program director for the post-graduate programs in project management at ESG UQAM. He can be contacted at petit.yvan@uqam.ca