The Journal of Systems and Software 85 (2012) 1255–1268

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Understanding post-adoptive agile usage: An exploratory cross-case analysis

Mali Senapathi [a,*], Ananth Srinivasan [b]

[a] School of Computing and Mathematical Sciences, AUT University, Auckland, New Zealand
[b] Department of Information Systems and Operations Management, The University of Auckland, New Zealand

ARTICLE INFO

Article history:
Received 10 August 2011
Received in revised form 4 February 2012
Accepted 11 February 2012
Available online 20 February 2012

Keywords:
Post-adoptive agile usage
Effectiveness
Kanban
Scrum

ABSTRACT

While past research has contributed to the understanding of how organizations adopt agile methodologies (AM), little is known about their post-adoptive usage in organizations. By integrating theories from systems development methodologies, diffusion of innovations, and agile methodology literature, this paper proposes a new model that identifies a set of critical factors pertinent to post-adoptive usage of agile practices. This model is used to inform analysis of post-adoptive usage of agile practices in two major organizations. The results indicate relative advantage, team attitude and technical competence, championing, and top management support (TMS) are the key factors determining the extent to which agile practices can be assimilated into an organization. Specifically, both findings and this model confirm that the deeper the assimilation of agile practices into the organization, the better understanding of how assimilation leads to specific improvements in its systems development outcomes.

1. Introduction

AM reflects the values articulated in the agile manifesto (Alliance, 2001) which places emphasis on the 'people' factor rather than on processes and tools. Working software over comprehensive documentation, customer collaboration over contract negotiation, and the embracing of change over following a plan (Cockburn, 2002; Highsmith, 2002) are other value factors. Due to a more flexible approach toward software development, many organizations have adopted agile practices. This approach, in turn, has engendered substantial related research; mainly studies reporting the adoption, adaptation, and comparison of specific measures such as productivity and quality software development between traditional and agile methods (Dyba and Dingsoyr, 2009). However, most studies that report high adoption rates and success of AM do not define or measure 'effectiveness' of usage of agile practices or identify effective factors. Therefore, there is a lack of clarity on what 'usage' or 'effectiveness' mean in the context of agile development. This has led to gaps in the literature as noted by (Abrahamsson et al., 2009): "*studies of issues associated with post-adoption use of agile methods are much less in number, even though there is increasing need to have a better understanding of agile methods in use as many organizations have completed adoption stage and agile methods start to become well-established processes of these organizations*".

Accordingly, the first imperative of the current study is to build and test a model that identifies the factors affecting agile usage. The model also proposes agile usage as a mediating factor affecting agile effectiveness. A large body of information systems (IS) research is based on well-established theories such as diffusion of innovations and IS implementation research. While such theories are suitable for a particular range of adoption scenarios and technology types, they cannot be directly applied to complex innovations such as AM, because the outcomes of such application are sensitive to the fit between the underlying assumptions, specific features of the adoption context and the technology in question (Fichman, 1992). Instead, such theories need further evaluation and extension to make them adequate for the study of agile practices (Pikkarainen et al., 2007). Therefore, in this paper, we evaluate the application of these theories to the case of AM through the development of a comprehensive integrated framework for investigating agile usage and the impact of agile usage on agile effectiveness. The study focuses on how extensively the innovation is used after adoption within the organization, rather than its adoption per se. In the context of software process innovations, this is generally referred to as the innovation's degree of assimilation into the organization.

According to Kwon and Zmud (1987), diffusion of innovation is a six-staged process comprising *initiation*, *adoption*, *adaptation*, *acceptance*, *routinization*, and *infusion*. While the initial three phases (*initiation*, *adoption*, *adaptation*) relate to 'adoptive' behavior of an innovation, the last three phases (*acceptance*, *routinization*, *infusion*) relate to an innovation's post-adoptive use. The factors which drive an innovation across the adoptive phases differ from those that affect the post-adoptive phases (Karahanna, 1999). Prior studies on AM have paid more attention to examining factors that drive organizations to initially adopt AM rather than on those that affect their continued usage. Therefore, this study specifically identifies the

* Corresponding author.
E-mail address: mali.senapathi@aut.ac.nz (M. Senapathi).

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2012.02.025

factors affecting the post-adoption usage of agile practices within organizations. The core research question is: *what are the critical factors that affect the post-adoptive usage and effectiveness of agile practices in organizations?* To answer this question, the paper is organized into the following sections. First, an a priori model of AM usage is derived by drawing from established theories such as diffusion of innovation and the IS implementation literature. Next is a description and analysis of the empirical findings that emerged as a result of application of the framework to two organizations that relate to their 'post-adoptive' implementation and use of agile practices into their systems development operation for a period of at least two years.

The case study approach is useful because it enables investigation of a contemporary phenomenon within its natural setting appropriate for contemporary topics such as post-adoptive usage of agile practices where theory and research are still in their formative stages. The use of multiple case design in building or testing an a priori theoretical model is recognized in the literature (Eisenhardt, 1989; Yin, 1994). Therefore, the current study tests the a priori model by using the case study approach to inform analysis of post-adoptive usage of agile practices in two large organizations.

The paper makes three principal contributions. First, it draws from well-established theories to develop an a priori model for understanding post-adoptive agile usage. While such conceptualizations have been criticized for denying relevant theoretical interpretations from the data (Ahrens and Dent, 1998), Eisenhardt (Eisenhardt, 1989) argues that such a process of deriving a theoretical model by constant comparison across cases, evidence, and the literature has the potential to generate less biased theory. Secondly, it identifies a set of critical factors that impact the continued usage of agile practices and examines the relationship between usage and usage effectiveness measures. Thirdly, the application of the framework to two real cases enables conceptualization of the critical factors that affect effective usage of agile practices which should contribute to our research knowledge and inform working practice. One of the cases, BBC Worldwide (BBCW) was an extension of a previous independent research study by Middleton and Joyce (2010a,b), and the results and findings of application of the proposed model to this case have been published (Senapathi et al., 2011). This paper extends and revises this earlier study, adding a second case Statistics NZ (Stats NZ) and performs a cross-comparison and analysis of the two case studies.

The paper is structured as follows. The first section presents the a priori model by integrating insights from relevant theoretical backgrounds and the literature. The second section describes the research methodology. The third section presents the research findings, describing the application of the framework for each organization in turn. Then comes a discussion and analysis section which is followed by notes on the probable implications for future research and practice.

2. Theoretical background and the research model

The usage of systems development methodologies (SDMs) in general is a versatile concept (Iivari and Huisman, 2007) and, given the lack of strong theoretical and conceptual base in AM research, this study draws from the vast base of empirical studies in the diffusion of innovation, IS implementation research, and AM. The diffusion of innovation and IS implementation models have been tested and validated extensively in the IS literature (Cooper and Zmud, 1990; Davis et al., 1989; Fichman and Kemerer, 1993; Gallivan, 2001; Iivari, 1996; Kwon and Zmud, 1987; Raghavan and Chand, 1989) to explain the constructs of the adoption and implementation of new IS innovations (Gallivan, 2001). They were previously recognized as useful frameworks for enhancing our understanding of complex technologies (McChesney and Glass, 1993; Raghavan and Chand, 1989; Rogers, 2003), and recently have generated interest in the agile research community (Pikkarainen et al., 2007).

Information system (IS) process innovation is defined as any new development, implementation, and maintenance of IS in an organizational context (Swanson, 1994). IS process innovations are expected to improve the overall quality and productivity of systems development (Mustonen-Ollila and Lyytinen, 2003). According to the IT diffusion framework (Fichman, 1992), which distinguishes both the locus of adoption (individual vs. organization) and the class of technology (type 1 technology with low knowledge burden and low user interdependence vs. type 2 technology with high knowledge burden and high user interdependence), AM can be classified as an example of type 2 innovations. Organizational assimilation is described as the degree of implementation within the adopting unit as a whole (e.g., the department, division, or team) (Gallivan, 2001). This study focuses mainly on the post-adoption stage and on the factors that influence assimilation in other adopting units such as teams, projects, and divisions.

2.1. Diffusion of Innovations

Though diffusion of innovation (DOI) theory's relevance to the software engineering process innovations is recognized as enhancing overall understanding of the diffusion process (Raghavan and Chand, 1989), it has also been criticized for its inability to explain diffusion of complex technologies due to its simplistic assumptions (Lyytinen and Damsgaard, 2001). It has been suggested that such theories should be suitably adapted for complex process innovations (Fichman, 1992) such as AM. While earlier generalizations of DOI focused on individuals' adoption of personal-use innovations that did not require extensive specialized knowledge prior to adoption, some later research focused on extending diffusion theory to complex technologies including frameworks for studying organizational diffusion (Fichman and Kemerer, 1993; Mangalaraj et al., 2009; Pikkarainen et al., 2007).

Rogers's (2003) organizational DOI is a five-stage sequential process which is divided into two main phases: initiation (adoption) and implementation (post-adoption). DOI theory posits that an innovation is more likely to be rapidly adopted if it (i) offers specific advantages in comparison with its predecessor (relative advantage), (ii) is compatible with existing values, practices, etc. (compatibility), (iii) has ease of use (complexity), (iv) can be trialed (trialability), and (v) is observable (observability). While these attributes may influence an innovation's rate of adoption, not all characteristics necessarily have the same effect on an innovation's post-adoptive usage. For example, while an innovation's trialability (i.e., being able to trial an innovation on a specific project or for a particular period of time before it is adopted) may have some influence on its rate of adoption, it might not have a similar impact on its continued use. In the context of continued usage, attributes such as complexity (Iivari, 1996) and observability (Riemenschneider et al., 2002) have not been found to significantly impact the acceptance of systems development methodologies. Therefore, we propose that only *relative advantage* and *compatibility* be included as innovation factors in our model.

2.2. Information systems implementation model

The concept of an innovation's 'depth of usage' was first introduced by Zmud and colleagues (Cooper and Zmud, 1990; Kwon and Zmud, 1987) who called it 'infusion' in their six-stage implementation model. This phase refers to the '*innovation penetrating deep into the organization*' (Gallivan, 2001), and '*. . .the extent to which an innovation is used completely and effectively* (Wynekoop

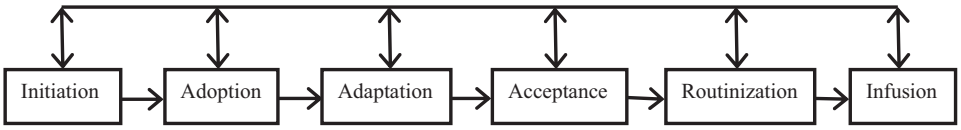

Fig. 1. IS implementation model.
Adapted from Cooper and Zmud (1990) and Kwon and Zmud (1987).

and Russo, 1995)'. Their implementation model is well known in the IS literature as "...an example of good definitions which serve as a model for adequate construct definition (Prescott and Conger, 1995)", which has been used extensively in studying assimilations of a number of complex technologies such as computer aided software engineering (CASE) tools, distributed database management systems, object-oriented methodologies, and recently in agile Extreme Programming (XP) (Gallivan, 2001; Green and Hevner, 1999; Mangalaraj et al., 2009). It consists of six stages as per Fig. 1:

- Initiation: need for change is recognized, a match is identified between an innovation and its application in the organization
- Adoption: a decision is made to adopt an innovation
- Adaptation: an adaptation to suit the contextual needs
- Acceptance: use of the innovation
- Routinization: an increase in the extent and intensity of use
- Infusion: increased usage in a more comprehensive and integrated manner results in increased effectiveness of systems development.

As the focus of our research is on organizational assimilation of AM, the IS implementation model's emphasis on organizational effort and the significance of effective feedback needed to successfully diffuse an innovation within a user community were particularly relevant. The rationale is that knowledge gained by effective feedback and reflective learning is crucial for the successful assimilation of adaptive innovations such as AM. Learning and reflection that occurs in one project or region can influence further assimilation in other projects, regions, etc. (Agarwal et al., 1997; Orlikowski, 1993). For example, a Scrum team that experiences some specific improvements may influence other teams to implement Scrum, or a team that has been using Scrum practices for a long period may come up with a set of practices that better reflect their contextual needs. Therefore, the underlying premise of this study is that the more deeply and widely the agile practices are assimilated into the organization, the greater the likelihood of gaining better understanding of the agile practices that work within a given context, and whether specific improvements in systems development (e.g., quality, productivity) have been achieved.

2.3. AM research

The theoretical foundation from the agile literature is drawn from Williams et al.'s (2004) Extreme programming evaluation framework (XP-EF) which is a framework that can be used to assess the extent to which an organization has adopted XP practices. The framework is specific to XP. However, given the lack of sufficient theoretical foundations in AM, some of the factors in the XP-EF, which were found applicable to the usage of any general agile practices were adapted for use in our framework. The original framework consists of three parts: XP context factors (XP-cf), XP adherence metrics (XP-am), and XP outcome metrics (XP-om). XP context factors record contextual factors which are grouped into six categories: software classification, sociological, project-specific, ergonomic, technological, and international.

Out of these six categories, sociological factors which record the personnel characteristics of the project (such as experience level of team, attitude), and technological factors (agile practices, tool support) were selected as potential factors affecting continued usage. None of the measures from the adherence metrics were included as they were specific to XP. The three metrics (quality, productivity, and customer satisfaction) of XP-om were also included as effectiveness measures. However, there are some important differences between XP-om metrics and the measures specified in this study. In XP-om, productivity is measured using metrics such as lines of code per month (KLOEC/PM) and user stories per month, and quality is measured using metrics such as number of test defects/KLOEC. In this study, however, quality refers to outcomes such as improved quality in defect management, reduction in the number of defects, and improved quality of the delivered product. Similarly productivity refers to outcomes such as lead time, development time, and faster delivery. This is logical because this research is more interested in studying the depth of usage of agile practices including any combination of agile practices (XP + Scrum, XP + Kanban, Scrumban, etc.) rather than those that are specific to a particular AM such as XP (which is the focus of the XP-EF framework). The adapted version of the framework as applicable to this study is shown in Fig. 2.

2.4. Related research

Some recent studies have focused on 'success' factors of agile practices from an adoption perspective. For example, studies conducted by Chan and Thong (2009), Chow and Cao (2007) and Misra et al. (2009) identify a set of potential factors that impact perceived success or 'acceptance' of agile practices where success is measured using outcomes such as quality, time, and cost. Moreover, the stream of the literature relating to agile adoption is not easy for busy practitioners to comprehend and apply in practice (Esfahani et al., 2010). From a sociological perspective, Whitworth and Biddle (2007) highlight the significance of the 'people' factor

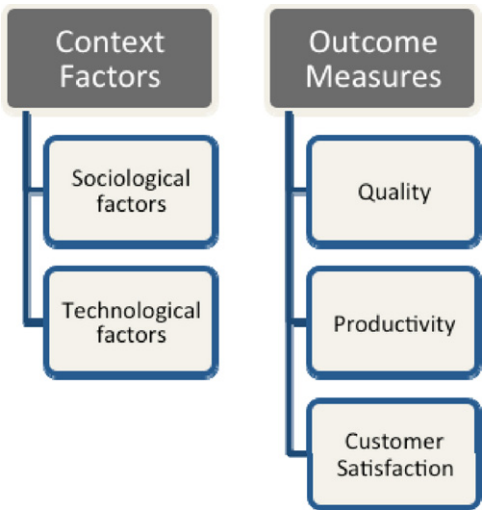

Fig. 2. Agile evaluation framework.
Adapted from Williams et al. (2004).

by investigating the social processes that contribute to the success of AM. More comprehensive frameworks such as the Agile Software Solution Framework (Qumer and Henderson-Sellers, 2008) focus on introducing agile practices in organizations based on an organization's degree of agility. Other studies that have specifically focused on later stages of implementation include studies conducted by (Pikkarainen et al., 2007) and (Mangalaraj et al., 2009). While Pikkarainen et al. focus on the assimilation of XP/Scrum practices using three case studies, Mangalaraj et al. investigate the acceptance of XP practices across different teams within the same organization. While these studies have added value to the extant agile literature, they are limited in terms of focusing only on specific types of AM such as XP or Scrum. Moreover, the concept of effectiveness as an outcome of increased usage is not covered. In software development, terms such as 'agile' or 'AM commonly refer to one of the major agile methods such as XP, feature-driven development and Scrum, each of which prescribes a set of core practices, values and principles. However, in practice, most organizations do not strictly follow or adhere to any one particular agile method, but use a tailored approach by combining a number of agile practices from different agile methods that reflect their contextual requirements as noted by Patton (Patton, 2009) "*. . .if you dumped all these good practices out onto a table, you'd have quite a buffet of very good practices with which to tailor your own process. And that's exactly what most organizations do*". Therefore, in this study, agile usage does not refer to one particular agile method such as Scrum but rather to a mature implementation of a set of agile practices. This is an extended version of our previous study on the investigation of the continued use of agile practices (Senapathi and Srinivasan, 2011). It aims to add value to the current body of knowledge by investigating the assimilation of agile practices at the organizational level and the impact of such usage on achieving effectiveness measures.

3. Research model

The proposed integrated research model (Fig. 3) is derived from the different theoretical backgrounds discussed in the previous section. The model is proposed as an initial formulation of the key factors affecting agile usage and its effectiveness and no claim is made that the factors identified here are exhaustive. Further case studies around agile practices should add to or modify the various factors and measures presented here. The model identifies a set of critical factors that affect agile usage. Specifically, it proposes that nine critical factors (two innovation, three sociological, two technological, and two organizational) influence the degree of agile usage in organizations. The model also proposes the relationship between the degree of usage and agile usage effectiveness. The agile usage model is set out in Fig. 3: (1) Agile innovation factors (*relative advantage*, *compatibility*) adopted from the innovation diffusion literature (Rogers, 2003), (2) Sociological factors (*attitude*, *experience*, *technical competence*) adopted from XP evaluation framework (Williams et al., 2004), (3) Technological factors (*agile practices, tool support*) adopted from the XP evaluation framework (Williams et al., 2004), and (4) Organizational factors (*TMS*, *Methodology Champion* (MC)) adopted from the IS implementation literature (Huisman and Iivari, 2002; Iivari, 1996; Roberts and Hughes, 1996).

3.1. Innovation factors

Relative advantage is defined as by Rogers (Rogers, 2003) as "*. . .the degree to which the innovation is perceived better than its precursor*". An organization can successfully move to post-adoptive phases only after an innovation consistently offers specific improvements in comparison to its predecessor. There is empirical evidence to suggest that relative advantage is a significant factor in predicting the usage of SDMs (Hardgrave et al., 2003; Huisman and Iivari, 2002) as well as evidence at the organizational level (Iivari, 1996). Other studies that have validated the effect of this attribute on the use of an innovation include software process engineering innovations (Fichman and Kemerer, 1993), programming languages (Agarwal and Prasad, 2000), and acceptance of AM (Chan and Thong, 2009).

"Compatibility refers to the degree to which an innovation is perceived as being consistent with the existing practices, values, [and] past experiences. . ." (Rogers, 2003, p23). In terms of post-implementation stages it is described as the fit between an innovation and a particular context (McChesney and Glass, 1993), which implies that the chosen agile practices must match the context in order to be effective and successfully integrated into the organization. Since the adoption of agile methods represents a major shift in the paradigm of systems development (Rajlich, 2006), it entails major alterations to work practices, investment in tools that support and facilitate rapid iterative development, versioning/configuration management, refactoring and other agile techniques (Nerur and Mahapatra, 2005). For example, (Mangalaraj et al., 2009) found that the use of multiple languages such as C,

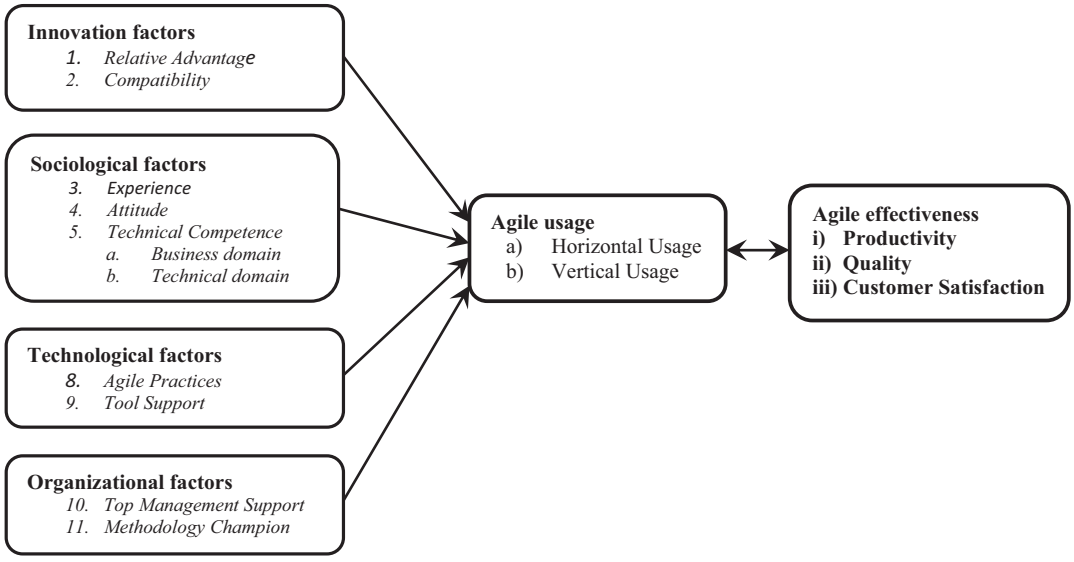

Fig. 3. The research model–agile usage model.

Java, C++, and Motif, in a large legacy project caused serious challenges to the adoption of XP practices such as code ownership, refactoring, and automated testing. There is sufficient evidence for the significance of compatibility in a number of empirical studies on traditional SDMs (Hardgrave et al., 2003; Huisman and Iivari, 2002; Iivari, 1996), and it is also identified as a significant predictor of the acceptance of AMs (Chan and Thong, 2009).

3.2. Sociological factors

The selected sociological factors include *attitude* – the team's positive or negative salient beliefs about the consequences of continuing to use an innovation (Karahanna, 1999) and *experience* – *a* high level of team experience will contribute to increased productivity, *and technical competence* (i.e., *domain expertise*, a team with a high level of expertise may not be subject to the learning curve associated with an unfamiliar domain, and *language expertise*, a team with high level of expertise may not be subject to the learning curve associated with learning an unfamiliar programming language) (Agarwal and Prasad, 2000; Williams et al., 2004).

Past research suggests that high levels of technical knowledge within an organizational unit will strongly influence the effective use of an innovation (Rai and Howard, 1994). Sociological factors have also been found to affect post-adoptive phases of assimilation rather than earlier stages since they influence employees' willingness and ability to adapt to changes in newer practices and adopt the values associated with an innovation. Therefore higher levels of attributes such as personal innovativeness, personal resilience, and tolerance of ambiguity may facilitate diffusion whereas lower levels of these attributes may constrain diffusion (Gallivan, 2001). Throughout the agile literature, technical competency (individual and average team competence) has been repeatedly emphasized as a critical factor to project success (Cockburn and Highsmith, 2001; Highsmith, 2002).

3.3. Technological factors

Technological factors include *agile development practices*, and *tool support* (*use of Project Management tools such as* VersionOne, RallyDev, tools for automated testing). Recording *agile practices* may provide insight into which hybrid practices may lead to more effective outcomes (Williams et al., 2004). This outcome is because there is often a substantial difference between the 'vanilla' textbook version and the actual "method-in-action" used in practice because most prescribed agile practices are tailored to meet the contextual needs of software development (Fitzgerald, 1997; Pikkarainen et al., 2007). For example, Pikkarainen et al. (2007) found that different XP practices reached different levels of assimilation during different periods of usage, and practices that addressed specific needs of the adopting team reached deeper assimilation levels.

While the significance of *tool support* in the adoption of software process innovations in general has been recognized (Green and Hevner, 1999), their importance in facilitating acceptance of agile practices in terms of providing support to specific XP practices like refactoring, continuous integration, and test-driven development is also documented (Mangalaraj et al., 2009). For example, (Mangalaraj et al., 2009) found that the use of software tools such as IntelliJ and CruiseControl greatly helped in speeding up XP practices such as refactoring and test-driven development.

3.4. Organizational factors

Organizational factors include TMS and MC. TMS refers to ongoing, continual support and encouragement of the top management executives in the adoption and implementation of innovations (Sultan and Chan, 2000), which has emerged as a key factor affecting implementation success in IS innovation and the systems development methodology literature (Huisman and Iivari, 2002). Active support and direct involvement of top management is capable of enabling a smoother transition to adaptive practices such as agile ones. For example, Roberts et al. (1998) and Roberts and Hughes (1996) found that lack of management support was one of the biggest obstacles in implementing SDM, while (Iivari, 1996) reported significant relationships between CASE usage and management support both at the individual and the organizational level. In the context of AM (Vijayasarathy and Turk, 2008) found lack of management support to be one of the biggest obstacles in the successful implementation of agile methods.

MC: Successful adoption of AM implies changes to the organizational culture, form and processes, which require significant changes in the systems development methods, values, and the learning of new practices. While traditionally MCs are predominantly viewed as playing a critical role in the introduction of new innovations in organizations (Premkumar and Potter, 1995), they are also characterized as bearers of persuasive and evaluative information about an innovation (Beath, 1991). According to Beath, MCs not only promote the users' vision of using an innovation, but also uses a variety of other influences (social, political, etc.,) to eliminate barriers to successful diffusion and implementation (Beath, 1991). In the context of AM, MCs play a significant role in encouraging and facilitating the on-going usage of agile practices (Pikkarainen et al., 2007), which include responsibilities such as convincing management, pushing through implementation hurdles, mentoring, and ensuring that agile practices are correctly adhered to by team members (Pikkarainen et al., 2007).

Given the increasingly important role of teams in agile development, it is proposed that all the factors identified above be assessed at the team level. This is because members of a team "*. . .work on a common set of tasks, influence each other through interactions, and engage in behaviors acceptable to the group as a whole. . .*" (Sarker et al., 2005), and because they have a deep understanding of the precise situation, they make the majority of decisions pertaining to the use of specific agile practices, tools, technologies, etc., (Mustonen-Ollila and Lyytinen, 2003). Moreover, team management practices that foster 'facilitate-and-co-ordinate' rather than 'command-and-control' style of leadership empowers the team to be more autonomous and is therefore believed to have a significant impact on the continued usage of agile practices (Mangalaraj et al., 2009).

3.5. Agile usage

Usage is a key measure of successful implementation of systems development methodologies (Huisman and Iivari, 2002; Iivari and Maansaari, 1998) and has been referred to using different terminologies such as, acceptance, usage/routinization, infusion (Cooper and Zmud, 1990), and usage, acceptance and incorporation (McChesney and Glass, 1993). The definition adopted for usage in the present study is similar to that proposed by Zmud and colleagues (Cooper and Zmud, 1990; Kwon and Zmud, 1987) where usage is preceded by, and based on the acceptance of the innovation. McChesney and Glass specify two measures for post-adoption usage: (i) *horizontal usage*; concerned with the use of the innovation across the organizations such as percentage of projects, number of teams. It may for example, refer to the spread of the use of Scrum practices from one team or project to multiple teams/projects, or from one region to many regions within an organization. (ii) *vertical usage* is concerned with maximum intensity of their use i.e., depth of use of specific agile practices. For example, Scrum is made up of three roles (Product Owner, ScrumMaster, Team), four ceremonies (Sprint Planning, Daily Scrum, Sprint Reviews, Sprint retrospectives), and three artifacts (Product backlog, Sprint backlog,

Burndown chart), and therefore a team which follows all these practices may ideally seem to have achieved maximum Scrum usage. However, Scrum gives a basic set of constraints to drive a process improvement (Kniberg and Skarin, 2009), and in reality a Scrum team may continue to use some of their predecessor practices such as XP's pair programming that may have been working effectively for their context. Teams gain an understanding of what agile practice works better for them as they start observing how things like quality, and productivity change, which may lead to further adaptations as they start reflecting on and drawing conclusions from the results. Though this is an on-going and continual improvement process, over a period of time teams gain a better understanding of whether a given set of practices works effectively for their context. The intensity of the use of these practices is what is referred to as 'vertical usage'. The important thing is not the agile practices you start with, but the set of practices that you end up with, i.e., the set of agile practices that you derive based on continual learning, improvements, adaptations and change based on a particular context.

3.6. Agile usage effectiveness

As discussed under related research, some studies have identified specific 'success' measures relating to adoption of agile methods such as 'cost', 'quality', and 'time' (Chan and Thong, 2009; Misra et al., 2009). Others have provided some insights on the various factors affecting 'acceptance' or 'use' without identifying any specific outcome measures. Though it might be impossible to specify with certainty what these outcomes are and to what extent they can be achieved, this study is more interested in examining if there are any specific improvements in the overall systems development process as a result of continued usage of agile practices as perceived by the organization. To that extent, we have found three effectiveness measures that have been identified as the core criteria for assessing post-implementation effectiveness of systems development methodologies (McChesney and Glass, 1993): (1) improved productivity in the development process, (2) improved quality of the development process (Huisman and Iivari, 2002; Iivari, 1996), and (3) customer satisfaction. Customer satisfaction has been recognized as a key measure of agile effectiveness in the agile literature where it is argued that an AM can be considered successful only if it adds value to the customer and is perceived as satisfactory by the customer (Cockburn and Highsmith, 2001; Misra et al., 2009). It is believed that in order to know whether there are any improvements in systems development, the innovation practices must be used deeply and extensively. This is because the innovation adoption theories presume that 'incorporation' cannot happen before they become a normal part of organizational life. For example, Iivari (1996) examined the impacts of computer aided software engineering (CASE) usage on perceived impacts. He examined how using the CASE tool impacted developer perceptions of quality and productivity based on the rationale that CASE tools cannot be effective unless they are continually used. However, agile usage as a factor affecting agile effectiveness has been omitted from the agile literature. Analyzing the relationship between agile usage and agile effectiveness, where effectiveness is measured as the impact of agile usage on systems development outcomes is crucial to understanding continued usage of agile practices in organizations. Unless agile practices are used more deeply and widely throughout the organization, we will not be able to ascertain its impact on systems development activities. This is reflected in our framework by the bi-directional arrow in Fig. 3 that links agile usage to agile effectiveness. However, it should be noted that due to the small amount of research on agile usage, the theorized relationships are tentative.

4. Research methodology

The research followed a case study methodology (Yin, 1994), with the aim of validating the proposed model by informing an analysis of the influence of various key factors on the post-adoption use of agile practices in two organizations. This approach was adopted for two main reasons. First, it enables investigation of a contemporary phenomenon within its natural setting (Yin, 1994), and is especially appropriate for newer topic areas such as post-adoptive assimilation of agile practices, where theory and research are still in their formative stages (Eisenhardt, 1989). Since the main objective of this study was to develop a rich, theoretical framework for understanding continued agile usage, an exploratory literal replication approach which enables predicting similar results (Yin, 1994) seemed particularly useful. The following criteria were used to select the cases:

(i) The organization had been conducting systems development for a range of different applications using one of the main AM (Scrum, XP) or a combination of practices from different AM (Scrum + XP, Kanban, Kanban +XP, etc.)
(ii) The organization had been using agile practices for at least 2 years
(iii) Key stakeholders (developers, ScrumMaster, etc.) were available for interviews

Secondly, it facilitated the testing of a well-formulated theoretical model, i.e., the a priori agile usage model. Yin (1994) suggests that whenever there are two cases, a cross-case analysis of the findings is likely to be more robust that results in confirming, challenging, or extending the theory. One of the cases, BBC Worldwide (BBCW) was an extension to a previously conducted independent research study (Middleton and Joyce, 2010a,b), and the findings of application of the proposed model to this case has been published (Senapathi et al., 2011), and the other case study, Statistics NZ (Stats NZ) was conducted as part of this study.

Our primary source of data was semi-structured interviews conducted with key stakeholders in both the cases. A total of seventeen interviews were conducted. An interview protocol was developed to elicit information on the agile practices used in various projects, and to test the factors and measure the outcomes of the usage model. Interviewees were selected based on their primary role in a given agile project. The interviewees' opinions on each of the constructs were sought to validate the model's constructs. They were also given the opportunity to openly discuss and identify any other relevant constructs or measures based on both their current and previous experiences. Different stakeholders such as developers, business analysts, project managers, Scrum masters (SMs), and IT service delivery managers were targeted for the interviews to capture different perspectives relating to post-adoptive usage of agile practices. Ethical guidelines were followed and informed consent was obtained by all participants. The duration of the interviews ranged from 30 to 60 min. All interviews were electronically recorded and transcribed. The interview data was supplemented with secondary data sources, including publicly available information such as websites.

BBCW is a wholly owned subsidiary of BBC which operates under the BBC Charter and Agreement, whereas Stats NZ is a government organization and New Zealand's national statistical office which provides statistical information needed in a wide range of areas. The two organizations were selected by paying attention to theoretical relevance which ensures that the substantive area addressed and the continued usage of agile practices was similar (Eisenhardt, 1989). Thus, both the organizations chosen for this study had within the past few years implemented agile practices in their systems development operations. In addition, the organizations were using

a different methodology. BBCW used Kanban while Stats NZ used Scrum. Some argue that Scrum and Kanban are significantly different, and that Kanban offers specific benefits over Scrum. For example, while Scrum uses time-boxed iterations and focuses on what work people have completed during retrospectives, Kanban uses WIP limits and enumerates the actual work done rather than on people (Middleton and Joyce, 2010b). However, we believe the question should not be on which one is better as it depends on the contextual requirements of an organization. Since both are based on adaptive principles and are radically different from any of the previous prescriptive methods of systems development, they can therefore be categorized under the 'agile' group. In fact, such a difference was sought in order to validate the proposed framework in different agile contexts. Therefore, the pattern matching data analysis technique which compares an empirical based pattern with the predicted one, and data involving both within-case and cross-case thematic analysis organized around the theoretical constructs in the usage model was found to be particularly beneficial (Yin, 1994). These differences allowed useful contrasts to be made during data analysis.

In the following section we present our analysis of the two case studies. Each case is structured around a brief description of the company and the agile projects studied, followed by a discussion of the factors, usage and effectiveness measures identified in the agile usage model outlined in Fig. 3.

5. BBCW

BBCW is the main commercial arm and a wholly owned subsidiary of the British Broadcasting Corporation (BBCWorldwide, 2010). Its mission is to create, acquire, develop and exploit media content and brands around the world in order to maximize the value of the BBC's assets for the benefit of the UK license payer. In a separate study, (Middleton and Joyce, 2010b) reported specific evidence of performance improvements of the Digital Hub (Digi-Hub) development team based on the practical experience gained between April 2008 and October 2009 when the team applied Kanban practices. Kanban is a lean approach to software development. It is based on the principles of lean thinking and has been shown to have the potential to at least double the productivity of both manufacturing and service organizations (Middleton and Joyce, 2010a). The core properties of Kanban are: (a) visualize the workflow: split the work into items and use named columns visually (on a wall for example) to illustrate where each item is in the workflow, (b) limit work in progress (WIP): assign explicit limits to how many items may be in progress at each workflow state, and (c) measure the lead time: optimize the process to make lead time as small and predictable as possible (Kniberg and Skarin, 2009).

The Digi-Hub team was made up of nine staff comprising project manager, business analyst, software architect, tester, lead developer, three developers and a support developer. It was working on a mix of developing new software and software maintenance. The technology used was C#, .NET, MS SQL Server, and legacy Connected Service Framework (CSF) code. Data was collected using semi-structured interviews with the team's key stakeholders, recording the precise operation of the lean system, observing daily 'stand up' meetings, and reviewing statistical analysis of the outputs from the system. The study focused mainly only on the software development team as at that time it was not yet an organization wide implementation. According to (Middleton and Joyce, 2010b), "*. . .team had to work within their existing framework to adopt lean practices where they had discretion in how to manage their own work, and try to influence other parts of the organization where possible*". The evidence of the effectiveness of the use of lean practices over a twelve month period showed that the lead time to deliver software improved by thirty-seven percent, consistency of delivery rose by forty-seven percent and defects reported by customers fell by twenty-four percent. Therefore this case with proven evidence of improvements in systems development over a period of time was found as a suitable candidate to be used as a pilot to test the model proposed in this study. As part of data collection for this study, further interviews were conducted with the members of the Digi-Hub in 2010 using the methodology described in the previous section.

5.1. Innovation factors

Prior to 2009, BBCW had been using Scrum methodology for almost two years. The Digi-Hub team was delivering enterprise messaging systems, which was very complex, poorly designed with very large and complex legacy code base. It was difficult for the team to accurately gauge the time required for completing a specific task. This was because a given piece of work would become bigger and bigger and there was ongoing reprioritization. Therefore, the team found it difficult to adhere to Scrum's time-boxed iterations. Moreover, they were not receiving timely feedback from customers and so there were a number of big projects that were not successful. It was about the same time (2009) that Kanban as a lean approach to software development was gaining popularity. Since the team was already using Scrum and given Kanban's flexibility in allowing other practices in conjunction with its core properties, provided a perfect context for a senior team such as Digi-Hub to implement Kanban practices in combination with their existing Scrum practices such as daily stand up meetings and retrospectives. The specific advantages of using Kanban over its precursor Scrum practices were (i) limiting the work in progress, i.e., stopping context-switching and getting efficiency out of visualizing the process and gaining an understanding of the whole process, (ii) reducing batch size, i.e., by reducing the size of units of work going through the pipeline which resulted in reduction of overheads including changes in requirements and (iii) '*granularity of visualization*'; software coach of the team described this as "*. . .in Scrum you don't get the granularity of visualization, Kanban is very granular, and what really happens is once you start visualizing your work, the whole team gets a collective understanding of how they build software*".

There were no major issues revealed relating to *compatibility*.

5.2. Sociological factors

The core part of the team was not only highly experienced but also technically competent in software development, and had previously used XP and Scrum practices for at least two years. Though the team was not familiar with the business domain, it was very experienced with the technology domain in terms of using Microsoft Stack, Web services, user interfaces, etc.

The team's attributes of positive attitude, willingness to change, high motivation, and personal innovativeness appeared to have an influence on their ability to adapt to changes in agile practices. For example, though Kanban was implemented at a time (2008) when there was limited literature, the team worked together by learning from each other, which ensured that the strengths and expertise of different members were reflected in the choice of their tasks and decisions (for example, some members of the team were very strong on test driven development) and had a common understanding of what the team thought good practices were. Though the members were not mandated to use Kanban practices, they themselves became very interested in and motivated by Kanban and started practicing what they learnt from books, attending Kanban meetings, etc. In addition, the high levels of experience of the team's project manager and software coaches (who taught technical rather than agile practices) also had an influence on the fast and deep assimilation of agile practices.

5.3. Technological factors

The team was previously a Scrum team which used two-weekly sprint time-boxed iterations, retrospectives, planning meetings etc., but when it moved to Kanban in 2008 it incorporated Kanban properties without making major changes to some of the existing XP/Scrum practices that were already working well such as daily meetings, refactoring and test-driven developments. Although the team had a number of engineering practices in place such as test driven development, automated acceptance testing, source control, bug tracking software, and decoupling, their use improved and consolidated during the assimilation stages (Middleton and Joyce, 2010b). In terms of Kanban practices, the team highlighted the need for deeply implementing all the core properties along with the support of necessary tools as one developer stated, "*. . .we visualized our workflow, limited WIP, we used models, collected data to manage and measure flow, i.e., we had all the five properties from David Anderson's book, and we had some of the emergent properties as well*".

A number of technical tools such as Cucumber for automated testing, TeamCity for continuous integration were used, and a lot of data collected from the Kanban board in conjunction with other internal systems for logging hours etc., were fed into project management practices. The team enjoyed autonomy in their choice of tasks, relevant tools, and making decisions relating to breaking a high level problem into smaller modules. Thus, deeper implementation of the core practices with an emphasis on optimizing the existing practices was found to have an impact on usage effectiveness.

5.4. Organizational factors

When the team moved to Kanban in 2008 the governance structure was: Business Board (strategy and budget), Project Board (detail and authorize specific work), Product Owner (reconcile business and customer wants), Users requesting work (sign off work completed), and End users (two–three hundred people). The team received very strong support from the Project Board, the Product Owner, IT management and knowledgeable colleagues, and benefited from hearing world renowned speakers such as Martin Fowler, Craig Larman and David Anderson, who gave talks at BBCW. However, support at the business board level was not as strong due to a lack of understanding of the changes in methodological and technical practices.

The MC played a critical role in liaising between the management and the agile teams. The MC was not only working with the corporate levels of management to educate them but was also involved in coaching other teams and providing one-on-one coaching. The MC coached the teams in ways that enabled them to find self-organizing behaviors. The team felt that the MC played a key role as a skilled facilitator especially in handling difficult issues, and one that enabled the team to challenge their perceptions of what they were really capable of rather than simply answering or solving their issues. One software coach said, "*. . .rather than giving the answer straight away, it's more allowing them to arrive at the solution that should be there*". Everyone in the team felt that the MC was the key driver who was responsible for creating a community interest in BBCW.

5.5. Agile usage

Horizontal usage of agile practices at BBCW increased consistently after the implementation of Kanban. Before Kanban was introduced almost half (fifty percent) of the projects were using Scrum, but when Kanban was implemented usage eventually increased to almost eighty percent. Kanban usage continued to spread widely as other teams (i.e., other than Digi-Hub) also started implementing Kanban. Use of Kanban spread into the BBC from BBCW, the spread being referred to as the 'Kanban flu'. The number of staff working on Kanban projects also increased steadily in almost exactly the same proportion as the number of projects described above.

In particular, vertical usage was found to play a significant role in determining the effectiveness of usage which is best reflected in words of a developer on the team; "*. . .the depth of adoption maps to an exponential curve of success – I think if you just do a couple of practices you are not going to get much benefit. If you do a lot of them and they are deeply engrained you are likely to be more successful*". In addition, the team's focus on reducing technical debt by allowing time for making improvements which might not be directly visible or requested by customers was also found to have an impact on overall effectiveness and increased future productivity (Middleton and Joyce, 2010b).

5.6. Agile usage effectiveness

Usage effectiveness was analyzed using three main factors: improved quality of the development process, improved productivity during the development process, and customer satisfaction. Specific improvements were recorded based on the data collected between October 2008 and October 2009 (for a detailed investigation and specific discussion of relevant results and findings refer to (Middleton and Joyce, 2010b)).

The *quality* of the development process improved. This improvement was measured by the number of live defects (live defects are the bugs reported by customers during a week plus the bugs still open for investigation). The numbers of live defects reported by customers fell by twenty-four percent, i.e., bugs were being fixed more quickly and the mean numbers of bugs open each week also declined slightly.

Three measures were used to gauge improvements in *productivity*. (i) Lead time: the total elapsed time from when a customer requests software to when the finished software is released to the customer. It tracks how quickly software is delivered to customers. Lead time to deliver software improved by thirty-seven percent, and consistency of delivery rose by forty-seven percent. (ii) Development time: sum of the time estimates of stories or tasks completed. It gives insight into the efficiency of development. Development time was recorded in working days. A variation in delivery times reduced by seventy-eight percent from 30.5 to 6.8, and the mean time to develop fewer and smaller software features declined by seventy-three percent from 9.2 to 2.5 working days. (iii) Release frequency (RF) is defined as the number of items released to customers per month. RF increased by a factor of eight from two from November 2007 to sixteen in October 2009.

It was believed that a lot of the team's behavior was driven by focusing on customer needs and satisfaction. One developer stated, "*. . .that's driving a lot of your behavior as well, so, if you are focusing on what the customer receives, then a lot of these practices will start to make more sense – breaking work into smaller units means more frequent feedback from customers which is reflected back into requirements of un started work*". By breaking the work into smaller units, the team was able to get more regular feedback from the customer. This enabled them to deliver new functionalities faster with more predictability, which in turn made their customer happier. The team believed ongoing collaboration with the customer had a positive impact on overall customer satisfaction.

6. Stats NZ

Stats NZ is a government organization which is New Zealand's national statistical agency and the major source of official statistics.

The programme of official social statistics (POSS) aims to provide a coherent set of official social and population statistics to meet the key information needs of both the government and the community. The POSS is made up of a suite of social surveys which comprises a range of statistical data collections such as the five yearly census of population and dwellings, household economic survey, and New Zealand health survey. Stats NZ has about one hundred and forty members in the IS department who are distributed across three locations: Christchurch, Wellington and Auckland. Scrum AM was adopted in a stealth-like manner on one of the projects relating to General Social Survey (GSS) in Christchurch in 2007. Later it was introduced into more low-risk, small to medium size projects in the Wellington and Auckland branches. Scrum is an iterative and incremental development methodology that emphasizes a set of project management values and practices with its distinctive emphasis on self-organizing teams, daily team measurement, and avoidance of following predefined steps (Larman, 2009). Some key practices include a daily stand-up meeting, fixed timeboxed iterations, and a demonstration to external stakeholders at the end of each iteration. Data was collected using ten semi structured interviews with key stakeholders of various teams in Wellington and Christchurch who were involved with at least one project under the POSS.

6.1. Innovation factors

Prior to 2007, Stats NZ was using Waterfall methodology which was supposedly overlaid on top of Rational Unified Process (RUP), but in reality they were not actually following RUP and had an incorrect understanding of the different phases. For example, elaboration was viewed as design, construction as building, and transition as testing. The teams spent the majority of their time on the requirement gathering phase and encountered problems in predicting the time needed for various tasks, which in turn affected their delivery schedules. The morale of the teams was observed to be very low, as the IT delivery manager (ITDM) described "*. . .we decided to trial it, particularly because of the low morale, low measures of success we were having in delivery, and once we trialed it, the developer's level of morale just leaped, and that's what made us continue with it. We could see it really changed their work enthusiasm*. These conditions set the stage for Scrum to be introduced as a viable methodology in some low risk, small projects, and based on the positive feedback from these projects its usage scaled up to other projects over a period of time.

Though there were no major *compatibility* issues in terms of using specific Scrum practices, there were issues in other areas such as resourcing. For example, there were cross-functional teams, and therefore trying to manage cross-functional resources for the duration of the project early on was difficult and incompatible with their existing resourcing model. From a Waterfall perspective, resources would become available in a Waterfall sense, i.e., testers would be available in the testing phase, and their resourcing was done at an individual level rather than at a team level which resulted in some constraints. There was resistance from business analysts as they struggled to understand their identity in the agile process or the agile framework. This was because while there was a lot of information about how developers, testers etc. would sit within the agile framework, there were not many guidelines about product engineers or people who liaise between the project stakeholders and the development team. Business analysts expected recognition for their role because the business rules at Stats NZ were detailed and complex and had many dependencies between them.

6.2. Sociological factors

Most team members were highly experienced (more than ten years) in a variety of roles such as business analysis, testing and development but had no specific experience with Scrum when it was first adopted. Over a period of time, the teams gained experience by learning from each other and applying it on a project by project basis. Most teams had knowledge of the business domain; almost all members in the IS department were internal employees and had been working there for at least five years.

In terms of technical expertise, teams included a mix of different competency levels. One of the project managers said, ". . .when we put them together as a Scrum team, there would be a mixture of these different experience and competency levels, for example, so that we could have a senior person helping a junior". However, most teams had moderate to high levels of expertise in terms of programming languages, tools and technologies etc. Initially, there was only one Scrum Master (SM) who looked after all the Scrum teams, but over a period of time more members were given the opportunity to take the SM role on a rotational basis. According to the MC, this resulted in them gaining a breadth of experience rather than the depth required to mentor other SMs; "But over time we learnt that they've got a breadth of experience, but not really depth. So I'm finding it hard at the moment to find Scrum Masters that can mentor other Scrum Masters. So we've got lots of Scrum Masters or lots of people who have had a go at the Scrum master role, but we don't have anybody who has built up the experience and built up the patterns and that knowledge of what does a particular dynamic in the team look like, and how to put that back on track, or how to encourage that to be able to come back on track". Most Scrum teams perceived themselves as self-organizing, but in reality they were not autonomous, and were finding it hard to move away from formalized roles. This was because there were two streams of management to which the staff members report to, one was the SM of the Scrum team and the other a manager role, for example, a business analyst (BA) who reports to a BA manager, a tester who reports to a test manager, and so on.

6.3. Technological factors

Though the teams were initially required to use all the Scrum practices 'out-of-the box', eventually they adapted the practices to suit the requirements of individual teams. There were variations in Scrum practices between the teams. Some experienced teams used all the practices well, some evolved from the initial stages and struggled to incorporate some of their own practices, while few newer ones still held onto some Waterfall techniques.

In terms of *tool support*, a number of tools/practices such as the Rally project management tool, in-house release management systems, defect management tools, and automated testing were used. In particular, Team Foundation Server (TFS) was extensively used by all the Scrum teams and perceived as highly compatible with their existing practices. One BA said, "*. . .for example, previously we had a repository where we had a lot of templates for different documentation, we were still able to use a lot of that – we just had to adapt. So, instead of having software requirements we changed that document to a user story document, and we also created a new document, called the as-built spec, so instead of worrying about software requirements and continuing to maintain them, we decided that as we went we create this new document about what we were creating as we went – that was quite a new thing, and for all these TFS worked alongside agile quite well*".

6.4. Organizational factors

Top management support: Although the first few Scrum teams had some support from their immediate managers such as their project manager, they were not confident of gaining support from management until they were able to demonstrate the benefits of using agile practices. The MC described this situation: "*. . .we had*

to do it by stealth in that sense which led to it being quite a difficult process bringing in change, especially when the organization wasn't supporting change. But looking back in hindsight, it would have been almost a hundred percent, or a ten-fold easier if we had senior executive team on board who were quite comfortable with what we were trying to do, how we were trying to do it and what the return on investment was going to be. That would have helped us to smooth down all of those issues; isolation, resourcing issues, resistance to change, and would have helped smooth all of that stuff out. And, arguably probably even provided a faster transition as well".

The *MC* who initially started as the only SM at Stats NZ, was involved in a variety of roles such as bringing in the concept of 'what is an agile project', 'how would it work', 'how would it look and feel?'. Providing one-to-one training, coaching other teams/SMs and enabling them to work independently on their respective projects was necessary. The MC was also instrumental in facilitating agile use in other branches, for example, one analyst in Wellington reported, "*. . .though he was based in Christchurch, he was always available and willing to come over to Wellington. . . he did not direct us in any way, just presenting facts and answer any queries. . .*". As the management required evidence of good results before Scrum could be formally adopted, the MC played a critical role in initially encouraging stealth adoption and then using appropriate strategies for the diffusion of agile practices in Stats NZ.

6.5. Agile usage

Horizontal usage of agile practices at Stats NZ increased after the implementation of Scrum on a regional basis. Scrum was first introduced in the Christchurch office in 2007. As it was very different from their previous methodology (waterfall), it was initially introduced in a few low risk, small to medium sized projects, which allowed the teams to evaluate the process. And based on some good positive feedback from the initial projects, Scrum usage scaled up to other projects. The methodology champion stated, "*. . .so with all the positive information that we had compared to previous projects using more of a waterfall approach we had some real data there that we really were encouraged as a team to take it on and it sort of snowballed from there*". And after about a year it spread to the other two regions, i.e., Wellington and Auckland. Overall, the combined horizontal usage from all the three regions continued to increase with almost eighty percent of all small projects and all larger programs of work (those over two years) using Scrum practices quite successfully, and in the words of a BA "*. . .almost every new project that starts now uses Scrum*".

In terms of vertical usage, all the core properties of Scrum were implemented in the initial projects. According to one Business Analyst, "*it was better to do by the book. I don't know whether we did it by the nth degree, at least we followed the process fully, yes, all the practices, definitely*". The MC added, "*they've followed them quite diligently and have been doing that for two and half years, so it'd be quite sustainable to do that*". And, over a period of time, many mature teams evolved beyond Scrum to incorporating their own practices to get the best value of both. So, for STNZ, vertical usage was not just a set of prescribed Scrum practices 'by the book', but a set of practices that were derived over a period of time based on continual adaptation and change.

6.6. Agile usage effectiveness

Two particular themes were found to relate to *quality* which are best reflected in the words of the ITDM and SM respectively: (1) improved quality of the delivered product "*. . .the reason why we are delivering better quality is simply that the user is more involved and so they are prioritizing daily at stand-ups, and definitely every three weeks at the sprint planning meetings by identifying what is more important to them and then the end product is that we are delivering better quality than we were before*", and (2) improved quality in defect management, "*we are closing the defects a lot quicker than we were previously which suggest that we are finding issues early, which is great. In some instances we are actually finding more defects which is a great indicator because it means we are spending time in the right places. And so if our rate of closing those defects is the same as the rate we're opening them for a majority of the projects then we know we've got the right rhythm; we're finding stuff, we're correcting it and that's a great quality measure in its own*".

Productivity: velocity measured in terms of story points per sprint was used to measure improvements in *productivity* and was believed to provide a basis for improving the accuracy and reliability of planning projects. One developer stated, there were "*. . .some instances where teams were reaching a fever-pitch in their productivity. And we were getting to a point where we were finishing projects early and it allowed us time to sit back and look at other aspects such as testing, where we were discovering new areas that we could do to look at improving the project without being under pressure*".

Customer satisfaction: one of the main motivations for adopting agile practices in Stats NZ was to improve their relationships with clients. The agile approach to collaborating with the client throughout the development process showed specific improvements in quality, regular and faster delivery, and played a critical role in improving their customer satisfaction as noted by their ITDM "*. . .their engagement has been phenomenal – they've really just got on board, and really enjoyed working closely with the teams. They feel a lot more, not necessarily in control of the project, but having some real, direct influence on the success of the project. They feel they can contribute with valuable decisions, really understanding what those issues are when technical issues pop-up, and being able to talk freely with the development team. . .the quality of that decision making as well; seeing that visibility of some of the engineering problems we've come up against for example, has really helped them to understand what does it mean when we ask for a particular functionality, because they've seen a lot more of how those decisions can impact the team directly, and how that can have a direct impact on the success of the project, it's helped them really understand what software engineering is all about which is pretty good*".

7. Discussion

While BBCW and Stats NZ both implemented agile practices within their systems development operations, their experiences differed due to variations in the organizational context, usage, and the requirements of key players around the adoption and continued use of agile practices. The structure of Stats NZ can be categorized as a matrix, with hierarchical control of agile practices exercised from the organization's Christchurch branch, while the development work was conducted via project teams operating out of the branches in Wellington and Auckland. BBCW, which owns and administers a number of commercial stations (e.g., BBC World News, BBC Kids) operates in a number of different territories and platforms, and therefore can be categorized as a divisional form with divisions/units having significant autonomy. Therefore, Scrum which prescribes practices such as time-boxed iterations and roles was found to be a perfect fit for Stats NZ's role-orientated, formalized culture. In contrast, for a more autonomous team such as the BBCW's Digi-Hub team which was working on a program with multiples streams where different pieces of work kept coming in, Kanban's properties of splitting the work into pieces, visualizing the workflow, etc., was found to be more suitable in comparison with Scrum's time-boxed iterations which were restrictive and difficult to adhere to.

Table 1 summarizes the findings of our cross-case analysis of these post-adoption usage of agile practices in the two case studies.

Table 1
Summary of findings.

	BBCW	Stats NZ
Methodology	Kanban	Scrum
Predecessor	Scrum	Waterfall/RUP
Innovation factors		
Relative advantage	Limiting work in progress Reducing batch size Granularity of visualization	Better time management in requirements gathering Improved delivery schedules Higher team morale
Compatibility	Not mentioned	Resistance from business analysts Incompatibility with resourcing model
Sociological factors		
Experience	High	Moderate
Attitude	Positive attitude, willingness to learn, and change, highly motivated	Mixed attitudes toward change
Technical competence	Technical – high Business domain – moderate	Technical – moderate Business domain – high
Technological factors		
Agile practices	Deeper use of all practices	Adaptations of Scrum practices
Tool support	Cucumber for automated testing, TeamCity for continuous integration	Rally project management tool, in-house release management systems, and Visual Studio Team Foundation Server
Organizational factors		
Top management support and methodology champion	Played a critical role in influencing agile usage	Played a critical role in influencing agile usage
Agile usage	Significant increase in both horizontal (80% of projects using Kanban – spread referred to as *Kanban flu*) and vertical usage (all core properties of Kanban)	Significant increase in both horizontal (almost all projects using Scrum) and vertical usage (derived adaptations of Scrum properties)
Agile usage effectiveness	Specific improvements recorded in improved quality and productivity	Improvements in both quality and productivity Customer engagement and satisfaction seen as having a direct impact on the success of the project

In both cases, *relative advantage* emerged as a key factor influencing agile usage. For example, Stats NZ faced difficulties in requirement prioritization, delays in delivering to the customer, etc., with their previous approach (Waterfall/RUP) and faced serious consequences such as decreased employee morale, delivery delays that affected their relationship with the customer. They believed that Scrum brought a radical change to Stats NZ. In contrast, BBCW found Scrum's time-boxed iterations very difficult to adhere to due to the dynamic growth of their individual pieces of work and constant reprioritization by the business. For them, Kanban's properties such as limiting the work in progress, reducing batch size and granularity of visualization resulted in a reduction of overheads including changes in requirements. However, it was surprising that no major compatibility issues were reported in either case, given the fact that such challenges have been reported in other recent studies (Mangalaraj et al., 2009). Though this clearly calls for further investigation, the findings might be interpreted as either that the teams (i) were technically competent enough to adapt to newer practices (BBCW) or (ii) the more challenging agile practices such as pair programming, test driven development were not being used effectively (Stats NZ). However, as previously discussed Stats NZ reported incompatibility issues with their resourcing models, resistance from business analysts, and an inability to leverage enough support from the organization, which impacted on the initial diffusion of agile practices.

In terms of sociological factors, BBCW displayed higher levels of technical competence and positive attitude which could be attributed to the fact that some of the key roles such as lead developer, software coach were hired on a contract basis. It is logical to expect such high levels of competence and performance from practitioners who are normally concerned about career mobility and escalating their marketability in the industry. In contrast, most IS employees in Stats NZ were internal with very little external hiring of experienced IS talent. Some of them been with Stats NZ for their entire careers. As a result, most of them had very good knowledge of the business domain but lacked the depth of experience needed to sustain the use of agile practices. Employees displayed mixed attitudes, where some of them were very passionate about agile practices few others were not self-motivated and developed an attitude of "us vs. agilists" This could be partly attributed to the paternalistic culture at STNZ, where the employees expect the organization to cater to their training and learning needs. Such cultural norms and mixed attitudes might reduce the likelihood of long-term sustainability of agile practices. It also highlights the fact that while differences in individual attributes such as attitude, personal resilience, etc., might not be evident during the initial stages, they will become more prominent during the later stages and will have an impact on full organizational assimilation.

In both the cases, the management wanted evidence of good results before the teams could continue using agile practices and therefore they had to do a lot of things very cautiously during the initial stages of agile implementation. Even after that, there was no major support from or direct involvement of the top management due to a lack of understanding of the changes in technical practices. While it is logical that organizations expect to see specific benefits, it is important for management to understand that contemporary software engineering process innovations such as agile practices involve continual learning and improvement based on effective feedback mechanisms over time and not merely the adoption of a new AM. As one project manager in STNZ pointed out, "*. . .there is not much support from the top management, they know only the theory side of it, and it would be good for them to know the practical side of it, they would get a better understanding of why or why not things are working? We need ongoing support. . .*". If organizations are not flexible enough to learn and adapt to the changing needs of contemporary practices, it will challenge the teams' effective use of agile practices and inhibit further diffusion into the organization.

Despite the fact that Stats NZ had substantially progressed in implementing agile systems development practices, they followed an approach that was well-suited to its formalized culture, and continued to use the training resources and strategies that they were familiar with. Some of these were not particularly relevant to the

specific needs of the agile teams as one project manager noted; "*. . .we do attend specialist training on Business Analyst, Project Management, etc., but we need to have specialist training on Scrum – we are keen to learn*". Stats NZ still overly values the specialist skill and knowledge roles and is finding it difficult to move away from the old formalized roles. On the other hand, in BBCW the teams enjoyed autonomy in terms of being able to make their own decisions in choices and adaptation of tools, use of technical and engineering practices etc. They also enjoyed the benefits of regularly hearing from leading experts and thought leaders on key principles of Kanban implementation which motivated the teams to continually learn from their experiences. Both companies believed that the role of the MC was crucial for the successful diffusion of agile practices in their organizations. As argued earlier, while the roles that MCs play at various stages of implementation has been well-recognized in the IS literature in general, the findings from this study in particular highlight the various challenges the role undergoes in agile practice which makes championing crucial to neutralizing the inhibitions, fears, resistance necessary for the effective diffusion of agile practices.

Specific evidence of improvements for all three measures were recorded in the case of BBCW (Middleton and Joyce, 2010b), and Stats NZ believed that there were significant improvements in all three measures (though there was no evidence of any formal measurement) and suggested that factors such as increased morale of team members had an indirect but significant influence on their productivity improvements. As their project manager explained "*. . .we are finding that productivity may be higher because people are happier, they have been used for their specialist skills in the right way*, *. . .and therefore morale has a big impact on productivity*".

Thus, our model, confirms that the deeper the agile practices are assimilated into the organization, the greater will be the likelihood of gaining a better understanding of whether such assimilation leads to specific improvements in its systems development outcomes. Findings from the two cases confirm that all factors, i.e., (1) Agile innovation factors (*relative advantage, compatibility*) (2) Sociological factors (*experience*, *attitude*, *technical competence*) (3) Technological factors (*agile practices*, *tool support*) and (4) Organizational factors (*TMS*, *MC*) had an influence on the effective usage of agile practices. Despite some of the challenges involved in diffusing agile practices as discussed in this paper, both organizations appear to have made significant improvements in achieving effective outcomes as a result of continued horizontal and vertical usage of agile practices. For example, in terms of vertical usage BBCW implemented all the core properties of Kanban in combination with their predecessor Scrum and XP practices such as daily stand-up meetings, test-driven development, and pair programming without making any major changes to their existing workflow, job titles, roles and responsibilities. As one team (Digi-Hub) began experiencing positive outcomes of using Kanban, its use spread to other teams, divisions, etc., to the business divisions and eventually throughout BBC. In the highly data and business-rule oriented environment of Stats NZ where people were very attached to their roles, recognizing the significance of key roles such as business analysts and making people in these roles feel they 'belonged to' the agile framework was critical. Scrum was first introduced in the Christchurch office, where all the practices were adopted 'by the book'. As the ITDM explained "*. . .we were determined at the very start that we should do scrum out of the box, because we knew that we needed to understand how it works first – we said people had to stick with it for at least 6 months. . .before adapting it. and later they adapted things to something that is more beneficial for them*". The complete adoption of all Scrum practices enabled the teams to derive effective adaptations of Scrum practices which in turn had an impact on specific improvements such as faster delivery and customer satisfaction. The use of Scrum eventually spread to all teams, and in the words of the ITDM "*. . .we started out with a handful of projects, and now almost every IT project is done using agile*".

When participants were asked to comment on any other factors (other than those proposed in the model) that might potentially impact agile usage, the common factor that emerged from both the organizations was budgetary constraint. One of the lead developers at BBCW said, "*you need to have enough budget to take on technical debt or practices to change the system, for example, we spent quite a lot of money on automating deployment*". The MC at STNZ reported, "*yes, if we are quite constrained then we cannot afford additional or specialized resourcing and that can have a negative effect*", and added time as another factor, "*in a government funding environment such as ours time constraints are very important. But not in a negative way, it can be actually quite liberating in terms of making conscious decisions. . . because we are looking to manage the project by scope. So, as we get closer to that time constraint we can start making more detailed decisions about what should or should not be in the product at that particular time. So, in the beginning we are quite ambiguous in our decision making in terms of scope. . .. but by the end we are making almost binary decisions on the scope; what is in and what is out of the project*". So time was perceived to have a positive influence on agile effectiveness as it allowed improvements in the quality of decision making. However, the impact of budget and time factors on agile usage needs to be further validated before it can be incorporated in to our proposed model.

In summary, we have seen how two different sets of agile practices each based on a core underlying methodology (Scrum and Kanban) experienced improvements in their systems development operations based on continued and deepened usage of agile practices. The findings highlight the significance of context, and the fact that the question is not which is better; Scrum or Kanban, but rather on finding the set of practices that works for a particular organization and using it deeply and extensively until specific evidence of improvements are evident. Then 'reflect, learn and change' can be based on consistent and continual feedback mechanisms, which needs to be an on-going process.

8. Conclusion

This article has presented an exploratory empirical study into understanding the factors that facilitate effective usage of agile practices in organizations. The evidence from these two case studies suggests that the usage framework presented impacts on research in this area. However, though the findings from this study confirm that the various factors identified in the framework play a significant role in affecting the post-adoptive usage of agile practices, it should be noted that these factors represent general factors conducive to agile usage. Given that the primary objective of our research is to articulate the a priori research model, we have focused our efforts toward this purpose. Usage effectiveness or success in certain organizations may, however, be influenced by specific factors or measures not identified in the model. Therefore, further research would refine or expand the model in several ways. A large scale survey could be used to statistically confirm and validate the usage framework's propositions at a more general level. Studies in different organizational settings and for different types of agile practices would potentially increase the applicability of this framework.

From a research perspective, the study draws upon related streams of literature to synthesize the various factors that relate to the effectiveness of agile usage. It focused on identifying factors that may explain post-adoptive behavior of agile usage. The emphasis was on how extensively and deeply the innovation is used after adoption, rather than its adoption per se. In the context of software process innovations, this is generally to as the innovation's degree

of assimilation into the organization (Fichman and Kemerer, 1997; Gallivan, 2001).

While some recent research (Mangalaraj et al., 2009; Pikkarainen et al., 2007) provides some insights into the post-adoption use of specific agile methods/practices such as XP or Scrum, there is very limited empirical research studying the actual use of agile practices using appropriate theoretical concepts or frameworks (Pikkarainen et al., 2007). Moreover, the relationship between agile usage and agile effectiveness i.e., usage as a factor affecting effectiveness, has been omitted from this important stream of research. Our research can be seen as a first step in addressing this gap in the extant agile literature which has developed a theoretical framework by identifying a set of factors that might affect post-adoptive agile usage where 'usage' refers to the continued use of agile practices ranging from the use of specific methods/practices such as XP or Scrum to the use of combination of properties/practices from different methods such as XP, Scrum and Waterfall.

For the practitioner community, the study has identified important factors that might play a significant role in affecting effectiveness of agile usage in organizations. Since higher levels of technical knowledge and expertise are key factors that affect the effective use of agile practices, managers can cultivate such expertise among their staff through effective coaching, training, and support programs. Adequate support from management is necessary to overcome resistance, compatibility issues and to impart necessary skills to members of the development team. While MCs play a critical role in overcoming the implementation hurdles and facilitating the propagation of agile practices, top management support will be crucial to sustain the efforts of such initiatives in order for the whole organization to evolve into an agile one.

References

Abrahamsson, P., Conboy, K., Wang, X., 2009. 'Lots done, more to do': the current state of agile systems development research. European Journal of Information Systems, 18.

Agarwal, R., Prasad, J., 2000. A field study of the adoption of software process innovations by information systems professionals. IEEE Transactions on Engineering Management 47, 295–308.

Agarwal, R., Tanniru, Wilemon, D., 1997. Assimilating information technology innovations: strategies and moderating influences. IEEE Transactions on Engineering Management 44, 347–358.

Ahrens, T., Dent, J., 1998. Accounting and organizations: realizing the richness of field research. Journal of Management Accounting Research 10, 1–39.

Alliance, A., 2001. Agile Alliance, Agile Manifesto.

BBCWorldwide, 2010. http://www.bbcworldwide.com/about-us.aspx.

Beath, C.M., 1991. Supporting the information technology champion. MIS Quarterly 10, 355–371.

Chan, F.K.Y., Thong, J.Y.L., 2009. Acceptance of agile methodologies: a critical review and conceptual framework. Decision Support Systems 46, 803–814.

Chow, T., Cao, D.-B., 2007. A survey study of critical success factors in agile software projects. The Journal of Systems and Software 81, 961–971.

Cockburn, A., 2002. Agile Software Development. Addison-Wesley, Boston, MA.

Cockburn, A., Highsmith, J., 2001. Agile software development: the people factor. Software Management.

Cooper, R.B., Zmud, R.W., 1990. Information technology implementation research: a technology diffusion approach. Management Science 36, 123–139.

Davis, F.D., Bagozzi, R.P., Warshaw, P.R., 1989. User acceptance of computer technology: comparison of two theoretcial models. Management Science 35, 982–1003.

Dyba, T., Dingsoyr, T., 2009. What do we know about agile software development? IEEE Software (September/October), 6–8.

Eisenhardt, K.M., 1989. Building theories from case study research. The Academy of Management Review 14, 532–550.

Esfahani, H.C., Yu, E., Annosi, M.C., 2010. Capitalizing on empirical evidence during agile adoption. In: AGILE Conference, 2010, pp. 21–24.

Fichman, R.G., 1992. Information technology diffusion: a review of empirical research. In: DeGross, J.I., Becker, J.D., Elam, J.J. (Eds.), Thirteenth International Conference on Information Systems. Dallas, TX, pp. 195–206.

Fichman, R.G., Kemerer, C.F., 1993. Adoption of software engineering process innovations: the case of object orientation. Sloan Management Review 34, 7–22.

Fichman, R.G., Kemerer, C.F., 1997. The assimilation of software process innovations: an organizational learning perspective. Management Science 43, 1345–1363.

Fitzgerald, B., 1997. The use of systems development methodolgies in practice: a field study. Information Systems Journal 7, 201–212.

Gallivan, M.J., 2001. Organizational adoption and assimilation of Comlex technological innovations: development of a new framework. The DATABASE for Advances in Information Systems 32, 51–80.

Green, G.C., Hevner, A.R., 1999. Perceived Control of Software Developers and its Impact on the Successful Diffusion of Information Technology, CMU/SEI-98-SR-013. Carnegie Mellon. Software Engineering Institute, Pittsburg, PA.

Hardgrave, B.C., Davis, F.D., Riemenschneider, C.K., 2003. Investigating determinants of software developers' intentions to follow methodologies. Journal of Management Information Systems 20, 123–151.

Highsmith, J., 2002. Agile software development ecosystems. In: Cockburn, A., Highsmith, J. (Eds.), The Agile Software Development Series. Addison-Wesley, Boston.

Huisman, M., Iivari, J., 2002. The individual deployment of systems development methodologies. In: Pidduck, A.B. (Ed.), CAISE 2002, LNCS 2348. Springer-Verlag, Berlin, pp. 134–150.

Iivari, J., 1996. Why are CASE tools not used. Communications of the ACM 39, 94–103.

Iivari, J., Huisman, M., 2007. The relationship between organizational culture and the deployment of systems development methodologies. MIS Quarterly 31, 35–58.

Iivari, J., Maansaari, J., 1998. The usage of systems development methods: are we stuck to old practices? Information and Software Technology 40, 501–508.

Karahanna, E., 1999. Symbolic adoption of information technology. In: Proceedings of Decision Sciences International Conference, Athens, Greece.

Kniberg, H., Skarin, M., 2009. Kanban and Scrum Making the Most of Both, http://www.infoq.com/minibooks/kanban-scrum-minibook.

Kwon, T.H., Zmud, R.W., 1987. Unifying the fragmented models of information systems implementation. In: Critical Issues in Information Systems Research. John Wiley & Sons, Inc, 227–251.

Larman, C., 2009. Agile and Iterative Development: A Manager's Guide. Pearson Education.

Lyytinen, K.J., Damsgaard, J., 2001. What's wrong with the diffusion of innovation theory? The case of complex and networked technology. In: M.A.A.a.B.L.M.E. (Ed.), Diffusing Software Products Process Innovations. Kluwer Academic Publishers, Norwell MA.

Mangalaraj, G., Mahapatra, R., Nerur, S., 2009. Acceptance of software process innovations – the case of extreme programming. Empirical Software Engineering 18, 344–354.

McChesney, I.R., Glass, D., 1993. Post-implementation management of CASE methodology. European Journal of Information Systems 2, 201–209.

Middleton, P., Joyce, D., 2010a. Lean software development: BBC Worldwide case study. In: Lean Software and Systems Conference, Atlanta, pp. 24–44.

Middleton, P., Joyce, D., 2010b. Lean software management: BBC Worldwide case study. IEEE Transactions on Engineering Management.

Misra, S.C., Kumar, V., Kumar, U., 2009. Identifying some important success factors in adopting agile software development practices. The Journal of Systems and Software 82, 1869–1890.

Mustonen-Ollila, E., Lyytinen, K., 2003. Why organizations adopt information system process innovations: a longitudinal study using diffusion of innovation theory. Information Systems Journal.

Nerur, S., Mahapatra, R., 2005. Challenges of migrating to agile methodologies. Communications of the ACM 48, 73–78.

Orlikowski, W., 1993. CASE tools are organizational change: investigating incremental and radical changes in systems development. MIS Quarterly 17, 309–340.

Patton, J., 2009. Kanband Development Oversimplified, http://www.agileproductdesign.com/blog/2009/kanban_over_simplified.html.

Pikkarainen, M., Wang, X., Conboy, K., 2007. Agile practices in use from an innovation assimilation perspective: a multiple case study. In: Twenty-Eighth International Conference on Information Systems, Montreal.

Premkumar, G., Potter, M., 1995. Adoption of computer aided software engineering technology: an innovation adoption perspective. In: DATA BASE Advances 26.

Prescott, M., Conger, S., 1995. Information technology innovations: a classification by it locus of impact and research approach. In: The DATABASE for Advances in Information Systems 32.

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.

Raghavan, S.A., Chand, D.R., 1989. Diffusing software-engineering methods. IEEE Software 6, 81–90.

Rai, A., Howard, G.S., 1994. Propagating CASE usage for software development: an empirical investigation of key organizational correlates. International Journal of Management Science 22, 133–147.

Rajlich, V., 2006. Changing the paradigm of software engineering. Communications of the ACM 49, 67–70.

Riemenschneider, C.K., Hardgrave, B.C., Davis, F.D., 2002. Explaining software developer acceptance of methodologies: a comparison of five theoretical models. IEEE Transactions on Software Engineering 28, 1135–1145.

Roberts, T.L., Gibson, M.L., Fields, K.T., Rainer, R.K., 1998. Factors that impact implementing a SDM. IEEE Transactions on Software Engineering 24, 640–649.

Roberts, T.L., Hughes, C.T., 1996. Obstacles to implementing a system development methodology. Journal of Systems Management 47, 36–40.

Rogers, E.M., 2003. Diffusion of Innovations, Fifth ed. Free Press., New York.

Sarker, S., Valacich, J., Sarker, S., 2005. Technology adoption by groups: a valence perpective. Journal of the Association of Information Systems 6, 37–68.

Senapathi, M., Middleton, P., Evans, G., 2011. Factors affecting effectiveness of agile usage – insights from the BBCWorldwide case study. In: Sillitti, A., et al. (Eds.), XP2011 L, 1 Ed. Springer-Verlag, Berlin/Madrid, Spain, pp. 132–145.
Senapathi, M., Srinivasan, A., 2011. Understanding post-adoptive agile usage – an exploratory cross-case analysis. In: AGILE Conference (AGILE), 2011, pp. 117–126.
Sultan, F., Chan, L., 2000. The adoption of new technology: the case of object-oriented computing in software companies. IEEE Transactions on Engineering Management 47, 106–206.
Swanson, E.B., 1994. Information systems innovation among organizations. Management Science 40.
Vijayasarathy, L.R., Turk, D., 2008. Agile software development: a survey of early adopters. Journal of Information Technology Management, 19.
Whitworth, E., Biddle, R., 2007. The social nature of agile teams. AGILE 2007, 26–36.
Williams, L., Layman, L., Krebs, W., 2004. Extreme Programming Evaluation Framework for Object-Oriented Languages Version 1.4, NCSU Technical Report.
Wynekoop, J.L., Russo, N.L., 1995. Systems development methodologies: unanswered questions. Journal of Information Technology 10, 65–73.
Yin, R.K., 1994. Case Study Research: Design and Methods. Sage Publications, CA.

Ananth Srinivasan is a Professor of Information Systems and Digital Commerce at the University of Auckland Business School. He is a Co-Director of the Centre of Digital Enterprise which is engaged in research about the deployment of digital technologies in organizations and society. He serves on the editorial boards of major international journals in the area of information systems.

Mali Senapathi is a Senior lecturer in the School of Computing and Mathematical Sciences at AUT University, Auckland, New Zealand. Her special focus is in the areas of systems development methodologies, in particular agile methodologies. She has strong interests on how these emerging methodologies differ from traditional ones with an emphasis on both technical and social aspects of systems development in the context of these new methodologies.