

Understanding Post-Adoptive Agile Usage – an Exploratory Cross-Case analysis

Mali Senapathi

School of Computing and Information Sciences
Auckland University of Technology
Auckland, New Zealand
mali.senapathi@aut.ac.nz

Ananth Srinivasan

Department of Information Systems and Operations
Management, The University of Auckland
Auckland, New Zealand
a.srinivasan@auckland.ac.nz

Abstract - The widespread adoption of agile methodologies raises the question of their continued and effective usage in organizations. An agile usage model consisting of innovation, sociological, technological, team, and organizational factors is used to inform an analysis of post-adoptive usage of agile practices in two major organizations. Analysis of the two case studies found that a *methodology champion* and *top management support* were the most important factors influencing continued usage, while innovation factors such as *compatibility* seemed less influential. Both horizontal and vertical usage was found to have significant impact on the effectiveness of agile usage.

Keywords- Agile usage, effectiveness.

I. INTRODUCTION

Agile methodologies (AM) emerged as a popular alternative to address the problems inherent in established methods to systems development [1]. Due to a strengthened interest in a more agile approach towards software development, many organizations have been motivated to adopt agile practices. This, in turn, has triggered substantial related research, mainly studies reporting on their adoption, systems development performance measures and comparison between traditional and agile methods [2]. The effectiveness of agile usage has not adequately been addressed by the agile community, and very little is known about their post-adoptive use and dissemination in general [1]. This has led to gaps in the literature, especially related to the understanding of the various notions that the measures ‘post-adoption usage’ or ‘effectiveness’ embrace in the context of agile methodologies. Accordingly, the first imperative of the current study is to build and test a model for explaining the post-adoption usage of agile practices.

According to Kwon & Zmud [3], diffusion of innovations is a six-staged process comprising *initiation*, *adoption*, *adaptation*, *acceptance*, *use*, and *incorporation* phases. While the initial three phases (*initiation*, *adoption*, *adaptation*) relate to ‘adoptive’ behavior of an innovation, the last three phases (*acceptance*, *use*, *incorporation*) relate to the ‘post-adoptive’ behavior of an innovation’s continued use and routinization. The factors which drive an innovation across the adoptive phases differ from those that affect the post-adoptive phases [4]. Prior studies on AM have given more attention to examining factors that drive organizations

to initially adopt AM, rather than on factors that affect their continued use after adopting a specific agile methodology. The current study is specifically interested in identifying the factors affecting the later stages of agile use, i.e., the post-adoption use, propagation and routinization of agile methods within organizations.

We use an exploratory case study approach to provide an analysis of two companies’ continued use of agile practices. Given that the usage of agile methodologies is a versatile concept, the current study draws from well-established theories such as diffusion of innovations and IS implementation research to derive an a priori model of AM usage. The next section summarizes this literature and presents the a priori usage model based on innovation, sociological, technological, team, and organizational factors affecting effective usage of AM. We then outline the research design used in the study followed by an analysis of the two case studies. The final part of the article synthesizes some conclusions from the cross-case comparison and discusses further implications for research and practice.

II. THEORETICAL BACKGROUND AND THE RESEARCH FRAMEWORK

The definition adopted for usage in the present study is similar to that proposed by Kwon & Zmud [3] who suggest that acceptance is preceded by, and based on, usage of the innovation beyond the pilot project stage, i.e., *Usage* -> *Acceptance* -> *Incorporation*, where usage is specified using two measures [5]: *horizontal usage* - concerned with the use of the innovation across the organization, and *vertical usage* - concerned with the depth of usage. Applying these definitions’ to the context of agile methodologies, *horizontal usage* is defined as the overall use of agile practices across the organization— for example, percentage of projects and developers/analysts using agile practices, and *vertical usage* is defined as the maximum intensity of their use, i.e., depth of use of specific agile values, practices, and policies.

In software development, terms such as ‘agile’ or ‘agile methodology’ commonly refers to one of the major agile methods such as extreme programming (XP), feature-driven development, crystal method, Scrum, and dynamic systems development method - 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 good agile practices from different agile methods that reflect their contextual requirements, “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” [6]. Therefore, in the current study, agile usage refers does not refer to one particular agile method such as Scrum but rather to the continued usage of agile practices, where practices might include combination of practices from XP and Scrum, or core properties of a kanban system alongside some XP and Scrum practices.

Most studies that report higher adoption rates and success of AM do not define, much less measure ‘effectiveness’ of usage of agile practices or identify factors affecting effectiveness. Post-adoption studies in agile methods have just started and are still in very early stages. For example, few recent studies [7, 8] which have focused on the later stages of agile assimilation have added value to literature by providing insights on the acceptance of specific agile methods such as XP. However, we need to understand the factors that affect the effectiveness of specific agile methods/practices to specific uses and specific contexts. A better understanding of the factors that affect successful incorporation of agile methods is believed to provide valuable insights from at least three perspectives: a) provide new theoretical insights into the factors affecting the effectiveness of agile usage, b) improve our understanding of post adoption use, processes and impact of agile methodologies, and c) contribute to industrial practice by providing insights into how agile methods can be effectively used in organizations.

The agile usage model is depicted in Figure 1. It identifies five groups of factors potentially affecting Agile Usage: (1) Agile innovation factors (*relative advantage, compatibility*) mainly adopted from the innovation diffusion literature [9], (2) Sociological factors (*experience level, Knowledge/expertise* (e.g. domain expertise, language expertise) adopted from eXtreme Programming (XP) evaluation framework [10], (3) Technological factors (*agile practices, tool support* (project management/use of automation tools) adopted from XP evaluation framework [10] (4) Team factors (*team management, team leadership*) adopted from XP literature [7], and (5) Organizational factors (*top management support, methodology champion*) adopted from the IS implementation literature [11-13].

A. Innovation factors:

There is sufficient empirical evidence to suggest that *relative advantage* is a significant factor in predicting the usage of Systems Development Methodologies [11, 14]. In terms of post-implementation stages, *compatibility* is described as the fit between an innovation and a particular context [5], which implies that the chosen agile practices must match its context in order to be effective and successfully infused in the organization. Since the adoption of agile methods represents a major shift in the paradigm of systems development [15], it entails major alterations to work practices, investment in tools that support and facilitate rapid iterative development, versioning/configuration

management, JUnits, refactoring and other agile techniques [16]. For example, [7] found that the use of multiple languages such as 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.

B. Sociological factors:

Some of the specific individual sociological factors that impact usage include attitude, experience, technical knowledge and expertise [17]. People with positive attitude and experience will embrace the newer practices easily and faster, and contribute to increased productivity [10]. Users with high levels of technical expertise in terms of programming languages, knowledge of different application domains, and other related software development technical practices, may not be subject to the learning curve associated with an unfamiliar domain, or with learning an unfamiliar programming language [10].

C. Technological factors:

There is often a substantial difference between the textbook ‘vanilla’ version and the actual “method-in-action” used in practice as most prescribed agile practices are tailored to meet the contextual needs of software development [8, 18]. For example, Pikkarainen et al. [8] found that different XP practices reached different levels of assimilation at 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 [19], their importance in facilitating the acceptance of agile practices in terms of providing support to specific XP practices such as refactoring, continuous integration, and test-driven development is also documented [7].

D. Team factors:

Given the increasingly important role of teams in agile software development, team characteristics and practices that facilitate the use of agile methods are more likely to affect their effective use. Team management practices that foster facilitate-and-co-ordinate rather than command-and-control style of leadership, and empower the team to be more autonomous appear to influence continued usage of agile practices [7].

E. Organizational factors:

Any innovative idea or practice without a champion is not likely to succeed [20]. In the context of agile methodologies, *methodology champions* play the significant role of change agents in encouraging and facilitating the ongoing usage of agile practices throughout the organization, which include responsibilities such as convincing management, pushing through implementation hurdles, mentoring, and ensuring that agile practices are correctly followed and adhered to by team members [8].

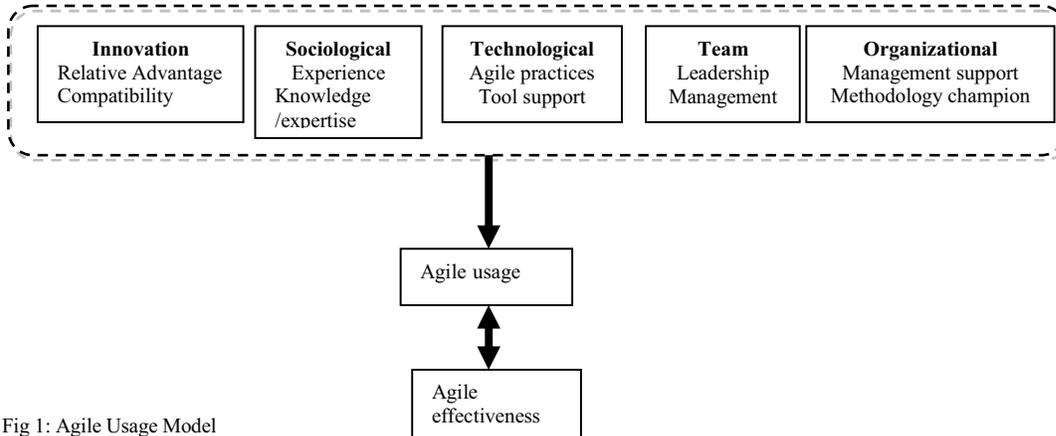


Fig 1: Agile Usage Model

Top Management support refers to ongoing, continual support and encouragement of the top management executives in the adoption and implementation of innovations[21], and is one of the main organizational factors that is consistently reported to facilitate systems development methodology implementation [11]. Lack of management support and interest has been found to be one of the biggest obstacles in implementing systems development methodologies [22] and successful diffusion of agile methods/practices [23].

F. Agile Usage and Effectiveness

As previously discussed, agile usage will be measured using two constructs: *Horizontal Usage*: percentage of projects and developers/analysts using agile practices, and *Vertical Usage*: maximum intensity of their use, i.e., depth of use of specific agile values, practices, and policies. Agile usage effectiveness will be measured using three measures: 1) improved productivity in the development process 2) improved quality of the development process [11, 12], and 3) customer satisfaction, which is recognized as a key measure of agile effectiveness in the agile literature [24, 25]. These measures are associated with the successful usage of systems development methodologies and have been identified as the core criteria for assessing post-implementation effectiveness [5].

III. RESEARCH METHOD

The research objective was to inform an analysis of how the various factors identified in the a priori model affect the post-adoption usage of agile practices in organizations. We used an exploratory case study approach because it enables investigation of a contemporary phenomenon within its natural setting [26], and is especially appropriate for newer topic areas such as post-adoption use or assimilation of agile practices, where theory and research are still in their early formative stages [27]. We applied the agile usage model shown in Figure

1, to two case studies of organizations that had known to use agile practices for more than three years. The cases met the conditions for testing a well-formulated theoretical model, the outcomes of which might result in confirming, challenging, or extending the theory [26].

Our primary source of data was semi-structured interviews conducted during 2010 with key informants in two major companies. An interview protocol was developed to elicit information on the company and its projects, its use of agile practices, and to test the various factors and measures identified in the a priori usage model thus far. Different stakeholders such as developers, business analysts, project managers, Scrum masters, and IT service delivery managers were targeted for the interviews to capture different perspectives relating to post-adoptive usage of agile practices. The interviews were recorded and transcribed for qualitative data analysis. This involved both within-case and cross-case thematic analysis organized around the theoretical constructs identified in the usage model [26]. The interview data were supplemented with secondary data sources, including publicly information available such as websites.

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 five types of factors identified in the usage model outlined in Figure 1: innovation, sociological, technological, team and organizational factors.

IV. BBC WORLDWIDE

BBC Worldwide is the main commercial arm and a wholly owned subsidiary of the British Broadcasting Corporation [28]. Its mission is to create, acquire, develop and exploit media content and brands around the world in order to maximise the value of the BBC's assets for the benefit of the UK licence payer. In a separate study, [29] reported specific evidence of performance improvements of the Digital Hub (Digi-Hub) development team based on continued usage of agile practices after adopting a lean approach. Therefore this case was deemed an appropriate

pilot case for testing the model developed in the current study. The team had an annual operating cost of £1.5m (US\$ 2.2m) and a development budget of £675K (US\$ 965k). It was made up of 9 staff: Project Manager, Business Analyst, Software Architect, Tester, Lead Developer, 3 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.

1) Innovation factors:

At BBC Worldwide, the Digi-Hub team was delivering enterprise messaging systems, which was very complex, poorly architected with very large and complex legacy code base. The specific advantages of using Kanban in comparison with its precursor Scrum method/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 overheads are reduced including changes in requirements and iii) '*granularity of visualisation*', "*.. in Scrum you don't get the granularity of visualisation, Kanban is very granular, and what really happens is once you start visualising your work, the whole team gets a collective understanding of how they build software*" (Developer, Digi-Hub team). There were no major *compatibility* issues as the team had recognized that Kanban was a good fit that matched the contextual requirements of the project requirements and the team.

2) Sociological factors:

People in the team were very experienced in software development and technically very competent. The team was familiar with XP and Scrum practices, though no one had used Kanban before. But most of them had positive attitude and the willingness to learn and change, and were very interested and self-motivated in learning kanban practices. However, the team was not familiar with the business domain, and it was interesting to find that this did not have any significant affect on agile usage. The rationale was that, "*if you have technical expertise you can usually understand new business domains quickly*" (Developer).

3) Technological factors

Though they were a Scrum team which used two-weekly sprint time-boxed iterations, retrospectives, planning meetings etc., when they moved to Kanban they incorporated Kanban properties without making major changes to some of the existing XP/Scrum practices that were working well, for example, daily meetings, refactoring, test-driven development. Therefore deeper and intensive use of agile practices with an emphasis on

optimizing the existing process eventually led to a mature implementation of Kanban "*we visualised our workflow, limited WIP, we used models, collected data to manage and measure of flow,..i.e, we had all the five properties from David Anderson's book, and we had some of the emergent properties as well*" (Developer, Digi-Hub team). 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 the project management practices.

4) Team factors

Kanban was implemented at BBC Worldwide during 2008-2009 which were early days for Kanban with not much literature, but the team worked together by learning from each other and ensured that the strengths and expertise of different members were well 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. The team enjoyed a lot of autonomy in the choice of tasks, relevant tools, and making decisions relating to breaking quite a high level problem into smaller chunks. The coach acted as a mentor who facilitated the effective use of kanban practices, and guided the team in the right direction whenever there were major problems or issues.

5) Organizational factors

The governance structure when the team moved to Kanban in 2008 was: Business Board (Strategy & Budget), Project Board (Detail & authorise specific work), Product Owner (reconcile Business & Customer wants), Users requesting work (Sign off work completed), and End users (200 – 300 people). The team received very strong support from the Project Board, Product Owner, and IT management, and also in terms of knowledgeable colleagues, and benefited from hearing to world renowned speakers such as Martin Fowler, Craig Larman, David Anderson, etc., come and talk at BBC Worldwide. However, support at the business board level was not as strong due to a lack of understanding of the changes in methodologies and technical practices. The *methodology champion* used a variety of influence processes to propagate agile usage throughout BBC Worldwide and played a critical role in actively and vigorously promoting his personal vision for implementing kanban.

B. Agile Usage and Effectiveness

1) Agile Usage:

Horizontal usage of agile practices at BBC Worldwide increased consistently after the implementation of Kanban. Before Kanban was introduced almost half

(50%) of the projects were using Scrum, and when Kanban was implemented the usage eventually increased to almost 80%. 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 BBC Worldwide, the spread being referred to as *Kanban flu*. The number of analysts/developers etc 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*”

2) 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 (refer to [29] for a detailed investigation and more specific discussion of the results and findings).

The *quality* of the development process improved which was measured by the number of live defects (live defects are the bugs reported by customers during a week plus the bugs still open). The numbers of live defects reported by customers fell by 24%, i.e., bugs were being fixed more quickly and the mean numbers of bugs open each week also slightly declined.

Three measures were used to measure 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 and reliably software is delivered to customers. Lead time to deliver software improved by 37%, and consistency of delivery rose by 47% ii) development time: this measure gives insight into the efficiency of development. Development time was recorded in working days - variation in delivery times reduced by 78% from 30.5 to 6.8, and the mean time to develop fewer and smaller software features declined by 73% 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 8 from 2 in November 2007 to 16 in October 2009.

The team was *customer* focused and responsive to customer needs, and was able to respond to the needs of the business by delivering new functionality faster and with more predictability. It was believed that a lot of the team’s behavior was driven by focusing on customer needs and satisfaction, “*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”.

V. AGILE USAGE AT STATISTICS NEW ZEALAND

Statistics New Zealand, is a government organization and is New Zealand's major source of official statistics and leader of New Zealand's Official Statistics System. 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 government and the community. POSS comprises a range of statistical data collections across government including the 5 yearly Census of Population and Dwellings, repeated household surveys and administrative data sources adapted for statistical purposes. Stakeholders are central and local government agencies and service providers, non-government organizations, international bodies, researchers and the public of New Zealand. Scrum agile methodology was adopted in a stealth manner on one of the projects relating to General Social Survey (GSS) in 2007. GSS is one of a number of social surveys and statistical initiatives undertaken through the POSS.

1) Innovation factors:

Prior to 2007, Statistics 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 majority of their time on requirements gathering phase and were encountering problems in predicting the time needed for doing various tasks, which in turn started affecting their delivery schedules. The morale of the teams was observed to be very low. These conditions set the stage for Scrum to be introduced as a viable methodology when it was adopted initially on 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.

There were no major *compatibility* issues within the development teams as they were not forced to use agile practices such as pair programming unless they themselves were keen to learn and use them. However, there were compatibility issues in other areas such as resourcing. For example, there were cross-functional teams, and so trying to leverage cross-functional resources early on for the whole duration of the project was quite difficult and was not compatible with their existing resourcing model. From a waterfall perspective,

normally resources would only become available in a waterfall sense, i.e., testers would only become available in the testing phase, and their resourcing was done at the individual level rather than at the team level which resulted in particular constraints. There was some resistance from the 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 their general 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 Statistics NZ were not only quite detailed and complex but had many dependencies between them.

2) Sociological factors:

Most team members were highly experienced (more than 10 years) in a variety of general roles such as business analysis, testing, development, but had no specific experience with Scrum. Over a period of time, the teams gained experience by learning from each other and using it in more and more projects. As a result, many staff gained a breadth of experience needed to take the role of Scrum master. However, they did not have any depth of experience required to mentor other Scrum masters, “*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*” (ScrumMaster). However, most members had very good knowledge of the business domain and had high levels of technical knowledge expertise in terms of programming languages, tools and technologies etc. For a large organization such as Statistics NZ with a large number of projects, in-depth experience of staff members was found to play a critical role in the continued usage of agile practices.

3) Technological factors:

Though the teams were initially required to use all the Scrum practices ‘out-of-the box’, eventually they were encouraged to adapt to suit the requirements of the individual teams. There was a variation of Scrum practices between the teams, with some experienced teams using all the practices well, and some teams evolving from the initial struggling stages to incorporating some of their own practices. There were also some newer teams still holding onto some of the waterfall techniques. In terms of tool support, a number of

tools such as Rally project management tool, in-house release management systems, and Visual Studio Team Foundation Server appeared to significantly influence the effective usage of agile practices throughout Statistics NZ.

4) Team factors:

Though most Scrum teams were seen as self-organizing, data collected revealed that staff members were not completely autonomous. This is because there were two streams of management to which the staff members report to, one is the Scrum master who facilitates and encourages the team to manage themselves, and the other is related to each staff member by their role in a controlling position, for example, a business analyst (BA) reports to a BA manager, a tester reports to a test manager, and so on. In addition there are number of specialist coaches or champions by role. In order to facilitate the increasing spread of agile usage throughout both the IT and the business departments, a coaching structure has been planned to be implemented at different levels, “*... so we are going set different levels of agile knowledge and each person will attain that level of knowledge and then become a mentor for a person that is at a lower level of knowledge*” (IT Delivery manager).

5) Organizational factors

Top Management support: Though the first few Scrum teams had some support from their immediate managers such as project manager, they were not confident of getting any support from the management until they were able to demonstrate the benefits of using agile practices, “*and so 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*” (Methodology Champion). The methodology Champion was not only instrumental in introducing agile practices by convincing management and pushing over or around approval and implementation hurdles, but played a critical role in facilitating their ongoing use throughout Statistics NZ.

B. Agile Usage and Effectiveness

1) Agile Usage

While there was some variation between teams in the vertical usage (i.e. the depth of use) of agile practices, horizontal usage continued to spread throughout the

organization with almost 80% of all small projects and all larger programs of work (those that are over two years) using Scrum practices quite successfully.

2) Agile Usage Effectiveness

Two particular themes were found relate to *quality*: 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 standups, 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*” (IT Delivery Manager), and 2) improved quality in the 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*” (Scrum master).

Productivity: Velocity measured used in terms of story points per sprint was one of the main measures used to improvements in *productivity*, which provided a basis for improving the accuracy and reliability of planning their projects. And 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*” (Developer).

Customer Satisfaction: The main motivation for Statistics NZ to adopt agile practices was to improve the relationship with their client. 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.. “*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” (Lead Developer)

VI. DISCUSSION

Table 2 summarizes the findings of our cross-case analysis of the post-adoption usage of agile practices in the two case studies.

Perceived *relative advantage* appeared to be influential in both the companies’ continued usage of agile practices. The benefits they achieved related to (1) significant time reduction in the requirements gathering phase (2) improved productivity resulting in faster delivery schedules (3) reducing work in progress, (4) improved team morale, and (5) improved customer satisfaction. There were no major compatibility issues in both the companies’ relating to technical practices. However, at Statistics NZ, issues relating to resourcing, resistance from business analysts affected agile diffusion during the initial stages. Therefore, relative advantage seemed to have a positive influence, and compatibility a negative influence on the degree of agile usage.

Both companies had teams with moderate to high levels of experience, technical knowledge and competence, and worked in collaborative and self-organizing styles. However, Statistics NZ highlighted that doing tasks such as automated testing, deployment, etc., more efficiently and faster will be crucial to seeing significant improvements in productivity. Though these findings are consistent with other recent studies [8], in the context of post-adoptive usage, it highlights that continuous training and coaching will be necessary in order to sustain benefits of agile usage after adoption. In terms of agile practices, BBC worldwide implemented all the core properties of Kanban, whereas Statistics NZ had various patterns of Scrum implementations depending upon the maturity levels of the teams.

However, an interesting fact to note was that both companies achieved success in incorporating agile practices without making major changes to their existing processes and structure. For example, in BBC Worldwide the change was driven by not making major changes to their existing workflow, job titles, roles and responsibilities. And, in a highly data and business-rule oriented environment of Statistics NZ it was important to recognize the significance of some existing key roles such as business analysts, and fitting them in the agile framework.

Table 2. Summary of agile usage findings

	BBC Worldwide	Statistics New Zealand
Methodology	Kanban	Scrum
Predecessor	Scrum	Waterfall
Innovation factors <i>Relative advantage</i> <i>Compatibility</i>	<ul style="list-style-type: none"> Limiting work in progress Reducing batch size Granularity of visualization <ul style="list-style-type: none"> Not mentioned 	<ul style="list-style-type: none"> Better time management in requirements gathering Improved delivery schedules Higher team morale <ul style="list-style-type: none"> Resistance from business analysts as they did not get “expected recognition for their role” Incompatibility with resourcing model
Sociological factors <i>Experience</i> <i>Knowledge/Expertise</i>	<ul style="list-style-type: none"> High levels of experience and technical expertise Positive attitude Not familiar with business domain 	<ul style="list-style-type: none"> Moderate to high levels of experience and technical expertise Generally positive attitude Very familiar with business domain
Technological factors <i>Agile practices</i> <i>Tool support</i>	<ul style="list-style-type: none"> Deeper use of all practices Tool support for a number of tasks, e.g.: Cucumber for automated testing, TeamCity for continuous integration 	<ul style="list-style-type: none"> Variation between teams with 1) some using most prescribed practices well, 2) some adaptations and 3) few newer teams using waterfall techniques Rally project management tool, in-house release management systems, and Visual Studio Team Foundation Server
Team factors <i>Team management</i> <i>Team leadership</i>	<ul style="list-style-type: none"> Collaborative, self-organising, sharing and learning from each other Facilitative style 	<ul style="list-style-type: none"> .. “the teams themselves are self-managing in the scrum environment, but there is still some overhead, because there is another manager that they are reporting to as well, so the testers report to a test manager, the developers report to a developer manager. So in that sense the teams are not so self-managing,..we are trying to relinquish that” (IT Delivery Manager) Coaching structure to facilitate coaching at different levels
Organizational <i>Top management support</i> <i>Methodology champion</i>	<ul style="list-style-type: none"> Strong support from the management in terms of providing training and getting experts in the field to speak at various occasions. Support at the business board level not very strong due to a lack of understanding of the changes in methodologies and technical practices Played a critical role 	<ul style="list-style-type: none"> “it would have been.. 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.” (Methodology Champion) Played a critical role
Agile Usage	<ul style="list-style-type: none"> Significant increase in both horizontal (80% of projects using kanban- spread referred to as <i>Kanban flu</i>) and vertical usage 	<ul style="list-style-type: none"> Significant increase in both horizontal (almost all projects using Scrum) and vertical usage
Agile usage effectiveness	<ul style="list-style-type: none"> Specific improvements recorded in improved quality and productivity 	<ul style="list-style-type: none"> Improvements in both quality and productivity – Customer engagement and satisfaction seen as having a direct impact on the success of the project

The methodology champion's role in actively promoting his/her vision for implementing innovative agile practices and pushing through a variety of constraints and hurdles was deemed critical for the overall success of the change process. Both companies' felt that executive involvement and top management support will be crucial to sustain the efforts of such initiatives, and in the creation of an 'adaptive' organization that can effectively assimilate agile practices.

In summary, data analysis from the interviews confirmed that all the factors, i.e., (1) Agile innovation factors (*relative advantage, compatibility*) (2) Sociological factors (*experience level, Knowledge/expertise*) (3) Technological factors (*agile practices, tool support*) (4) Team factors (*team management, team leadership*) and (5) Organizational factors (*top management support, methodology champion*) influenced effective post-adoption usage of agile practices. Both horizontal and vertical usage increased significantly and improvements in specific effectiveness measures (quality, productivity, and customer satisfaction) were reported in both the cases.

VII. 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 the two case studies suggests that the usage framework presented in the article is likely to be of interest to 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. However, usage effectiveness or success in certain organizations may be influenced by specific factors or measures not identified in the model. Therefore, further research could 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 the framework.

From a research perspective, the current 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 notion is generally referred to as the innovation's degree of assimilation into the organization [30, 31]. While some recent research [7, 8] provide some insights into the post-adoption use of

specific agile methods/practices such as XP or Scrum, there is very limited empirical research that have studied the actual use of agile practices using appropriate theoretical concepts or frameworks [8]. 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. The current study can be seen as a first step in addressing this important gap in the agile extant 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, waterfall, etc.

For the practitioner community, the study has identified the 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 should cultivate such expertise among their staff through effective coaching, training, and support programs. Adequate support is necessary to overcome resistance and compatibility issues and impart necessary skills to members of the development team. While methodology champions 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 to be agile.

REFERENCES

1. Abrahamsson, P., K. Conboy, and X. Wang, *'Lots done, more to do': the current state of agile systems development research*. European Journal of Information Systems, 2009. **18**(281-284).
2. Dyba, T. and T. Dingsoyr, *What do we know about Agile Software Development?* IEEE Software, 2009. **September/October**: p. 6-8.
3. Kwon, T.H. and R.W. Zmud, *Unifying the fragmented models of information systems implementation*, in *Critical issues in information systems research*. 1987, John Wiley & Sons, Inc. p. 227-251.
4. Karahanna, E. *Symbolic Adoption of Information Technology*. in *Proceedings of Decision Sciences International Conference*. 1999. Athens, Greece.
5. McChesney, I.R. and D. Glass, *Post-implementation management of CASE methodology*. European Journal of Information Systems, 1993. **2**(3): p. 201-209.

6. Patton, J. *Kanband Development Oversimplified*, <http://www.agileproductdesign.com/blog/2009/kanban-over-simplified.html>. 2009.
7. Mangalaraj, G., R. Mahapatra, and S. Nerur, *Acceptance of software process innovations - the case of extreme programming*. Empirical Software Engineering, 2009. **18**: p. 344-354.
8. Pikkarainen, M., X. Wang, and K. Conboy. *Agile Practices in Use from an Innovation Assimilation Perspective: A Multiple Case Study*. in *Twenty Eighth International Conference on Information Systems*. 2007. Montreal.
9. Rogers, E.M., *Diffusion of Innovations*. Fifth ed. 2003, New York: Free Press.,.
10. Williams, L., L. Layman, and W. Krebs, *Extreme Programming Evaluation Framework for Object-Oriented Languages Version 1.4*, in *NCSU Technical Report*. 2004.
11. Huisman, M. and J. Iivari, *The Individual Deployment of Systems Development Methodologies*, in *CAISE 2002, LNCS 2348*, A.B. Pidduck, Editor. 2002, Springer-Verlag: Berlin. p. 134-150.
12. Iivari, J., *Why are CASE tools not used?* Communications of the ACM, 1996. **39**(10): p. 94 -103.
13. Roberts, T.L. and C.T. Hughes, *Obstacles to implementing a system development methodology*. Journal of Systems Management, 1996. **47**(2): p. 36-40.
14. Hardgrave, B.C., F.D. Davis, and C.K. Riemenschneider, *Investigating Determinants of Software Developers' Intentions to Follow Methodologies*. Journal of Management Information Systems, 2003. **20**(1): p. 123-151.
15. Rajlich, V., *Changing the paradigm of software engineering*. Communications of the ACM, 2006. **49**(8): p. 67-70.
16. Nerur, S. and R. Mahapatra, *Challenges of Migrating to Agile Methodologies*. Communications of the ACM, 2005. **48**(5): p. 73-78.
17. Agarwal, R. and J. Prasad, *A Field Study of the Adoption of Software Process Innovations by Information Systems Professionals*. IEEE Transactions on Engineering Management, 2000. **47**(3): p. 295-308.
18. Fitzgerald, B., *The use of systems development methodologies in practice: a field study*. Information Systems Journal, 1997. **7**: p. 201-212.
19. Green, G.C. and A.R. Hevner, *Perceived Control of Software Developers and its Impact on the Successful Diffusion of Information Technology*, in *CMU/SEI-98-SR-013*. 1999, Carnegie Mellon, Software Engineering Institute: Pittsburg, PA.
20. VandeVen, A.H., *Central Problems in the Management of Innovations*. Management Science, 1986. **32**(5): p. 590-607.
21. Sultan, F. and L. Chan, *The adoption of new technology: the case of object-oriented computing in software companies*. IEEE Transactions on Engineering Management, 2000. **47**(1): p. 106-206.
22. Roberts, T.L., et al., *Factors that impact implementing a SDM*. IEEE Transactions on Software Engineering, 1998. **24**(8): p. 640-649.
23. Vijayarathy, L.R. and D. Turk, *Agile Software Development: A survey of early adopters* Journal of Information Technology Management, 2008. **19**(2).
24. Cockburn, A. and J. Highsmith, *Agile Software Development: The People Factor*. Software Management, 2001.
25. Misra, S.C., V. Kumar, and U. Kumar, *Identifying some important success factors in adopting agile software development practices*. The Journal of Systems and Software, 2009. **82**: p. 1869-1890.
26. Yin, R.K., *Case Study Research: Design and Methods*. 1994, CA: Sage Publications.
27. Eisenhardt, K.M., *Building Theories from Case Study Research*. The Academy of Management Review, 1989. **14**(4): p. 532-550.
28. BBCWorldwide. 2010; Available from: <http://www.bbcworldwide.com/about-us.aspx>.
29. Middleton, P. and D. Joyce. *Lean Software Management a BBC Case Study*. Proceedings of the *Lean Software and Systems Conference 2010* Atlanta, <http://atlanta2010.leanssc.org/proceedings/>
30. Fichman, R.G. and C.F. Kemerer, *The Assimilation of Software Process Innovations: An Organizational Learning Perspective*. Management Science, 1997. **43**(10): p. 1345 - 1363.
31. Gallivan, M.J., *Organizational Adoption and Assimilation of Complex Technological Innovations: Development of a New Framework*. The DATABASE for Advances in Information Systems, 2001. **32**(3): p. 51-80.