TOWARDS AN UNDERSTANDING OF THE CONTEXTUAL INFLUENCES ON DISTRIBUTED AGILE SOFTWARE DEVELOPMENT: A THEORY OF PRACTICE PERSPECTIVE

Maureen Tanner  
*University of Cape Town*

Chigona Wallace  
*University of Cape Town*

Follow this and additional works at: [http://aisel.aisnet.org/ecis2012](http://aisel.aisnet.org/ecis2012)

Recommended Citation

[http://aisel.aisnet.org/ecis2012/178](http://aisel.aisnet.org/ecis2012/178)
TOWARDS AN UNDERSTANDING OF THE CONTEXTUAL INFLUENCES ON DISTRIBUTED AGILE SOFTWARE DEVELOPMENT: A THEORY OF PRACTICE PERSPECTIVE

Tanner, Maureen, University of Cape Town, South Africa, mc.tanner@uct.ac.za
Chigona, Wallace, University of Cape Town, South Africa, Wallace.Chigona@uct.ac.za

Abstract

As the number of organisations involved in agile software development in the Global Software Development context is rising, there is a need for comprehensive research on distributed agile practices. Numerous papers have been published prescribing how distributed agile should be employed but do not provide enough information on the context of use. However, the successful use of a methodology depends on a broader and more tacit understanding of the usage context and it is inappropriate to assume that practitioners can merely apply prescribed sets of agile best practices and expect them to succeed. As such, this study proposes another perspective on agile software development by exploring the underlying forces shaping agile practices. Instead of merely prescribing sets of best practices, the reasons why distributed agile practices (as embodied by the Scrum methodology) are adapted and followed were investigated. By appreciating the various contextual factors shaping most of their decisions pertaining to how agile practices are adapted and followed, it is hoped that practitioners would make more informed decisions which would meet the needs of most stakeholders.

1. Introduction

Software development practices are constantly evolving and Global Software Development (GSD) is one of the growing trends. Another trend in the software industry is the adoption of agile software paradigms (Agile Alliance, 2001). As the number of organisations employing agile software development in GSD is rising (VersionOne, 2008), so is the need for comprehensive research on globally distributed agile practices. However, the body of knowledge on distributed agile software development appears to be mostly maintained by agile practitioners, who regularly produce white papers and experience reports on how to optimise the distributed agile process (e.g. Cannizzo, Marcionetti, & Moser, 2008). Studies have also been prescriptive, lacked rigour and failed to take the context of use into consideration. This study adopts a different perspective. In line with Introna and Whitley (1997), the study is founded on the basis that the successful use of a methodology as well as the use of sets of best practices depends on a broader and more tacit understanding of the usage context. Hence, the study addresses this research question: Why are particular distributed agile practices (as embodied by the Scrum methodology) adapted and followed? This will be realised by investigating the context of use. Through the lens of Theory of Practice (Bourdieu, 1990), two projects, executed between South Africa and Brazil and South Africa and India, were investigated. A framework of concepts from the Theory of Practice (Bourdieu, 1990) has been proposed to guide the data analysis process.

The study proposes another perspective on agile software development to practitioners by exploring the underlying forces shaping agile practices. By appreciating the various contextual factors shaping most of their decisions pertaining to how agile practices are adapted and followed, it is hoped that practitioners would make more informed decisions which would meet the needs of most stakeholders. Practitioners would also be able to evaluate the applicability of agile practices they currently employ, given the context they engage in. This might improve the chances of project success.
2. Distributed Agile Software Development from the literature’s perspective

As a means to understand the current state of literature on distributed agile software development, a total of 27 papers published between 1999 and 2010 were reviewed of which 12 were journal papers and 15 were conference proceedings. The study seeks to investigate software development as a social phenomenon with some technical implications and to emphasise the project phenomena. Hence, literature adopting a more technical perspective on project management has not been considered.

The publication list selected for review is not exhaustive. However, the list does cover a wide enough timeframe and range of sources, in order to provide a broad coverage of the various perspectives on distributed agile software development best practice. Papers from these specific journals and conferences were chosen because they appeared to have a continuous debate on the topic of distributed agile. A general inductive approach (Thomas, 2003) was employed to facilitate the analysis process. The themes identified specifically corresponded to studies where an understanding of the context was important but was ignored or not probed enough. The results of the analysis can be summarized into five themes; the use of technology, team management, managing the requirements, meetings, and applicability of various parts of agile methods. Each of the themes is described and problematized below.

Based on the literature, it appears that numerous studies and experience reports have focused on the use of technology during distributed agile software development. It was suggested that technologies should be used to mimic the face-to-face component of agile software development (e.g. Phalnikar, Deshpande & Joshi, 2008) and for automation purposes (e.g. Ramesh, Cao, & Baskerville, 2010). Even though the use of technology is clearly important in the context of distributed agile software development, these studies do not shed light on the intricate ways in which these tools are adapted by the participants based on their personal goals and agendas or contextual requirements.

Focus has also been placed on the need for effective team management. Researchers and practitioners have proposed numerous ways of tackling this issue. For instance, ways of enhancing collaboration between dispersed team members and clients have been proposed (e.g. Moore & Barnett, 2004). Various possible team configuration strategies were investigated and put forward as best practices (Sureshchandra & Shrinivasavadhani, 2008), and solutions were proposed on how to manage collaboration (e.g. Layman et al., 2006). The need of creating cultural awareness within the distributed team has also been highlighted (e.g. Fowler, 2003). However, team management is a social endeavour, and calls for a careful understanding of the complex human interactions within a team while they engage in agile software development. Each team member might have different past experiences, goals, interests and power level, which would impact on the manner in which the agile team collaborates. So far, past studies have rarely taken these aspects into consideration while prescribing distributed agile team management practices.

Requirements management is an important aspect of agile software development and software development at large. Past studies and experience reports have thus extensively focused on this topic. Various techniques for the prioritisation of functionalities have been investigated and proposed (e.g. Paasivara et al., 2008) and the use of testing and test-scripts to better understand requirements have been prescribed (e.g. Fowler, 2003). However, as will be discussed later, software projects are complex in nature and are composed of social constructions which render the requirement elicitation issue quite complex. It is thus important to understand what the social forces driving the requirements process are, and studies so far have not focused enough on this aspect.

The need for meetings is a core feature of agile and Scrum in particular (e.g. Beck, 1999). Studies proposed regular short status meetings to introduce transparency in the teams (e.g. Paasivaara & Lassenius, 2004). Various mechanisms to undertake iteration planning meetings tailored for remote sites have also been prescribed (e.g. Paasivaara et al., 2008). However, little is known about the strategies employed by the various stakeholders attending these meetings to achieve their goals and interests within
the agile project. These strategies might be key drivers to agile processes and should be further understood.

Numerous studies and experience reports have attempted to determine which aspects of various agile methods are applicable to the distributed setting (e.g., Holmstrom, et al., 2006). However, there are discrepancies among these studies on which agile practices fail and succeed in GSD. Hence, it is important to understand the underlying factors leading to the successful implementation of practices in some circumstances, and why these practices might fail in others.

3. The Nature of Software Projects

A software project is usually initiated by a project manager and project workers on behalf of an owner, to deliver unique and valued outcomes within a certain period of time, as well as within scope, quality and cost (Sewchurran & Scott, 2009). Up until the 21st century, researchers have been considering project management as a technical and best practice discipline (Sewchurran & Scott, 2009). However, in line with the Information Systems’ discipline perspective, software development is a social phenomenon with some technical implications. Thus, there is a need for a shift in emphasis from project management practices to the project phenomena. This paper argues that the same paradigm shift is necessary for the study of agile software development; agile software development should be understood as a phenomenon firstly and secondly as methods and tools.

Söderholm (2008) defines projects as being contextually dependent and continuously reliant on environmental relations. In practice, people have to cope with various sets of influences (Sewchurran, 2009) which give rise to ad-hoc practices which are circumspectively enacted. Often these ad-hoc practices are later reported as best-practices in numerous industry reports and even in empirical studies. Work practices which emerged as informal end up being prescriptive (Thomas, 2006, p.92). Project work is, however, innovative by nature (Sewchurran, 2009) and often requires improvisation and intuition to cope with unexpected circumstances (Söderholm, 2008). The use of best practices do not allow for the management of these unexpected circumstances and do not allow for innovative actions, applying detachment strategies, setting up intensive meeting schedules, and negotiating project conditions (Söderholm, 2008). Whilst making these prescriptions, projects are perceived as goal-oriented systems of activities which are completed through rigid following of project life cycle steps in order to achieve progress. Since projects are social constructions (Linehan & Kavanagh, 2004), it is important to focus research efforts on understanding the as-lived experiences of projects, based on the assumption that project processes are complex social processes (Cicmil et al., 2006). This study investigates software projects in line with what was mentioned by Cicmil et al. (2006). The focus is on gathering knowledge about the project, people, technologies and the means through which relations between these actors are coordinated and controlled, as well as to what end.

4. Theory of Practice

This study employed the Theory of Practice (Bourdieu, 1990) as a theoretical framework to provide an explanation for why particular agile practices are adapted and followed in the GSD context. This framework was deemed appropriate for this study as it focuses on the relationships between people, social structures and the influences present in individual interactions, through empirically-informed sociological perspective (Rhynas, 2004). To answer the aforementioned research question, the following concepts from the Theory of Practice were deemed relevant: Field, which in turn can have Constraints and Demands; Habitus; Practice, which can pertain to Decision Making and Improvisation; Agents, who have Goals and Interests and Past Experiences, and Strategies. Bourdieu’s Theory of Practice is complex and is composed of several more concepts.
Bourdieu defines a field as a social formation or arena in which agents (i.e. individuals who undertake certain practices) manoeuvre and struggle in pursuit of desirable resources (e.g. cultural goods, housing, intellectual distinction, social class prestige) (Bourdieu, 1992). These social formations can be institutions, rules, rituals, or conventions which allow for certain activities or discourse to occur within the field (Webb, Schirato, & Danaher, 2005).

Habitus relates to the tendencies and pre-dispositions that shape people’s conduct, thoughts, feelings and judgments (Bourdieu & Wacquant, 1992). Habitus is internalized in the minds and bodies of agents and further imposes socially accepted behavior (Hanks, 2005) by functioning as “the generative basis of structured, objectively unified practices” (Bourdieu, 1990, Vi). In doing so, culture is established and common communication practices are shaped, which might include gestures, manner of speech, and other verbal and non-verbal embodied actions (Hanks, 2005).

Field and habitus are reciprocally constructive and interact under given concrete situations, thus resulting in the emergence of practices (Bourdieu & Wacquant, 1992). However, habitus orchestrates practice by both harmonising interests and by ensuring that all agents share and adhere to a common consensus on the meaning of the practices (Acciaioli, 1981). A field provides the frame of analysis for the study of any aspect of social life. Practice is produced because of the encounter between habitus and its disposition, as well as the constraints and demands of the social field in which the actor is moving. Practices can be explained as recognizable patterned actions, in which both individuals and groups engage. Bourdieu (1973) describes this theory as having a dominant circular path. In this causal loop of generation and reproduction, actors internalise the structure of a field as habitus. Habitus, in turn, generates practices, and practices serve to reproduce and reinforce the structure of the field. Practice has a purpose even though it is accomplished without conscious deliberation. Practices are generated by dynamically combining past experience, present situation, and the anticipation of the future consequences of these actions. Practices are not recognised by agents as explicit principles. Instead, they are “embodied corporeally in postures and attitudes and interactionally in the style of strategies whose implementation constitutes practice” (Acciaioli, 1981, pp31).

Actors have goals and interests, and the identification of these goals and interests might help in identifying the source of their practice in their own reality. This is known as strategizing (Jenkins, 2002). The manifestations of these strategies are termed as position taking. Bourdieu (1992) states that “these position-takings are inseparable from the objective positions occupied by the agent as a result of their possession of a determined quantity of specific capital”. Therefore, strategies are employed by agents and institutions to improve or defend their positions in relation to other “occupants” of the fields (Naidoo, 2004). These concepts are represented in Figure 1.

---

**Figure 1**  Proposed Research Framework
5. Methodology

The study was critical in nature. A multiple case study approach was employed to investigate the distributed agile phenomenon within a real-life context (Yin, 1994). The case selection was purposive and site selection was based on geographic coverage and the use of an agile methodology. In addition, a similar agile methodology (i.e. Scrum) was chosen for both sites in order to allow the results from the first case to inform the second, and to seek more richness in the data being gathered. Care was also taken to select organisations having mature agile setups, (i.e. the teams having used Scrum for more than 1 year). However, a heterogeneous GSD setup was sought in order to acquire a wide understanding of the various possible social influences inherent to the various setups.

In the first case, all team members belonged to one organisation but were distributed across two countries, namely South Africa (Cape Town) and Brazil (Sao Paulo). The first case study was conducted in Company 1 (C1) which was part of a group of media companies dispersed all over the world. Three members were located in Brazil and five members in South Africa, under the leadership of the Chief Executive Officer (CEO) and the Chief Operations Officer (COO). The team was involved in the development of a knowledge management tool (MKX). The application was initially developed using a plan-based approach, which eventually proved inadequate for the project nature and setting. Consequently, the Scrum methodology was adopted.

In the second case, a service provider / customer relationship existed between the team members who belonged to different organisations distributed across two countries, namely India (Pune) and South Africa (Durban). The second case study was conducted in C2, a global software and services company spread across 18 countries around the world including India, US, UK, Germany, Sweden, South Africa, and Singapore among others. With a CMMI Level 5 certification, C2 provides expertise that spans Retail and Distribution; Banking; Financial Services and Insurance; Healthcare and Life Sciences; Manufacturing; Energy; Utilities. The team was involved in the development of an online gambling application.

Data was collected through semi-structured interviews, direct observation, documentation, and field notes. Furthermore, triangulation across various techniques of data collection was particularly beneficial as it provided multiple perspectives on an issue, supplied more information on emerging concepts, and allowed for cross-checking (Eisenhardt, 1989). Table 1 details the interviews conducted during the case studies. A total of 16 interviews and one focus group were conducted. The case number, site, job title, duration and mode of interview have been provided. As can be seen in Table 1, the respondents belonged to a wide variety of job titles with different levels of experience within the teams. These respondents were thus selected because they provided different perspectives on the practice and benefits of distributed agile. In particular, the key informants were selected according to the key roles in a Scrum team as it was anticipated that they would have different goals and interests within the Scrum process. In addition, snowballing was employed, whereby whenever one respondent mentioned that one particular team member or stakeholder had key knowledge on a specific topic, that person was also interviewed.

<table>
<thead>
<tr>
<th>Case No</th>
<th>Job Title</th>
<th>Experience in team</th>
<th>Mode of Interview</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>COO</td>
<td>2 years</td>
<td>Face-to-Face</td>
</tr>
<tr>
<td>1</td>
<td>Product Owner</td>
<td>1 year 3 months</td>
<td>Face-to-Face</td>
</tr>
<tr>
<td>1</td>
<td>Scrum Master</td>
<td>1 year 6 months</td>
<td>Face-to-Face</td>
</tr>
<tr>
<td>1</td>
<td>Technical Architect</td>
<td>3 months</td>
<td>Face-to-Face</td>
</tr>
<tr>
<td>1</td>
<td>CEO</td>
<td>3 years</td>
<td>Face-to-Face</td>
</tr>
<tr>
<td>1</td>
<td>Lead Technical Architect</td>
<td>2 years</td>
<td>Conference call</td>
</tr>
<tr>
<td>1</td>
<td>Front End Developer</td>
<td>1 year 8 months</td>
<td>Video Conference</td>
</tr>
<tr>
<td>2</td>
<td>Project Manager + Delivery</td>
<td>N/A</td>
<td>Focus group</td>
</tr>
</tbody>
</table>
Table 1. Case Interview Description

Observations of several meetings were undertaken including Scrum Meetings (18), Sprint Planning 1 and 2 (4), Sprint Reviews (1), Pre-Scrum meetings (4), and Meetings to discuss way forward (1). A total of 28 meetings were observed during the case studies. Examples of documents investigated include user stories, task boards, bug reports, specifications, burn-down charts and emails. The documents were primarily used to augment evidence from other sources. Field notes were recorded into a notebook computer on a daily basis including contextual data, personal comments and immediate interpretation of events (Kirk & Miller, 1986).

As recommended by Patton (1988) and Fereday and Muir-Cochrane (2006), a mixture of inductive and deductive analysis was employed to make sense of the data. Data was analysed in four stages with the theoretical framework to assist in the analysis of the data. In the first stage, a code book was developed based on the concepts from the theoretical framework to assist in the analysis of the data. In the second stage, an inductive approach was followed to summarise and identify themes in the data. The third stage of data analysis was deductive in nature and the code book was used to perform additional coding. Stage four was concerned with connecting the codes and identifying relationships in the data to prove or negate the propositions. Data analysis was also facilitated by the NVivo 8 qualitative analysis tool. In particular, stage four was undertaken in conjunction and iteratively to Stage 3.

6. Findings & Discussion

6.1 Past Experiences of the project team members

In both cases, the findings show that some agents internalised the values of other software development approaches they were previously employing (e.g. Waterfall). Their habitus was thus shaped by the values and beliefs inherent in these methodologies. However, past experiences of distributed agile participants had both a positive and a negative impact on the manner in which they engaged in agile practices. Distributed agile participants were either restrained in applying the prescribed agile practices or were inclined to adopt them when they perceived agile to be superior to other approaches: “having been in other methods like the Waterfall is good because you generally know the downfalls of them […] I didn’t find it that difficult to switch to Scrum”.

Past experiences of the participants impacted the manner in which they improvised while using the agile practices. For instance, Scrum does not recommend the use of any particular form of software engineering standards (Schuh, 2005). It is a project management methodology. When Scrum was used as a standalone methodology (i.e. without XP), participants drew upon experiences from other methodologies and past projects to engage in the software development process and decide on task estimation, analysis, design, development and testing. As mentioned by one interviewee, “we know what works and what doesn’t based on our past projects and we try not to make the same mistakes. We then
have to adapt our practices accordingly”. Smyth and Morris (2007) also put forward the notion of software projects being linked to history and context, and the past experience of agile participants is a particular instance of the historical elements of agile projects.

It was also noted that when agile teams had no prior agile experiences and were required to participate in an agile team / project for the first time, their understanding of agility was based on their past experiences with other projects and other methodologies. As mentioned by Söderholm (2008), the practice of agile in this case was thus based on intuition. They consequently improvised and drew on knowledge of past projects to devise practices which they labelled as agile practices. Often, after undertaking training, they realised that these practices were not in line with what agile prescribed and thus adapted them in further sprints. Past studies also highlighted the need for meetings to improve common project definition, planning, status updates, clearly delineated documentation format, agreement seeking mechanisms and uncertainty minimization (Tanner, 2009). The project teams investigated acknowledged the benefits of such mechanisms mostly from past mistakes. Consequently, they established a common goal of not repeating these errors, and embraced these practices.

6.2 Standards in place within the organisation

In line with Söderholm (2008) and Winter et al. (2006), who claimed that projects are contextually dependent, the study confirmed that agile projects are influenced by contextual factors. For instance in the study, particular agile practices were adapted and followed because the companies had to abide by standards. One interviewee mentioned that he had to “document as much as possible whatever because of the rules within the team about documenting”.

Participants were keen to abide by agile best practices set forward by key players in the field, but also had to abide by other demands in the field pertaining to corporate standards or customer contracts. The need to abide by standards could either create more commitment on the part of the agile participants, especially if the standards are in line with the agile values, or decrease the level of commitment if the agile values and the standards contradicted each other. In either case, the distributed agile practice was adapted accordingly. In situations where agile standards and other corporate standards were in contradiction, agile standards did not take priority. This might be due to the political aspirations, agendas and objectives of the stakeholders (Williams & Samset, 2010) involved in the agile project.

6.3 Improvisation mechanisms followed by project team members

The findings show that distributed agile projects require intuition and some degree of improvisation on the part of the participants to cope with unexpected circumstances as well as the demands and constraints within the development environment. The forms of improvisation emerge from the habitus of the participants. This is supported by Sewchurran (2009) and Söderholm (2008) who also recognized improvisation in software projects in general. Participants improvised while learning about agile, during sprint planning meetings where the planning poker game was avoided for simple tasks, while using the communication technologies, and improvised to mitigate for the daily challenges they faced in the distributed context. For instance, the Scrum tool which was used by C1 was described as not ideal by the team members since it did not provide all the functionalities which should have been available for the practice of distributed agile. However, because of the learning and innovation habitus inherent in the C1 distributed agile team, they “adapted it. [They] were using some columns in ways that [they] were not supposed to but [they] knew what the column was for. So [they] overcame most of the shortcomings”. Also, because of their past experiences with other software development methodologies, some Indian team members in C2 improvised and adapted their practices to match what they thought was the best way to be agile. Improvisation relates to the innovative (Sewchurran, 2009) and emergent (Sabherwal & Grover, 2009) nature of software projects. According to the definition of agility, distributed agile participants were agile in their degree of improvisation. Agility is defined as the “the readiness of an IS
development method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment” (Conboy, 2009).

The work processes which emerged from the sets of improvisation might not perfectly match official agile norms, but have emerged in response to the work environment and were adapted to meet environmental demands and leverage project success. Livari and Livari (2011) posit that a method can be qualified as agile if it exhibits emergent agility sufficiently regularly, and if the method is reasonably faithfully followed by the software developers. Livari and Livari (2011) also rightfully raise the question of how we evaluate whether a method is followed sufficiently faithfully. If the criteria employed for such an evaluation is based on the method’s principles and techniques, this might lead to a “dogmatic” interpretation of “agile” methods. This study deviates from such a form of agile evaluation.

6.4 Goals and Interests of project team members

The behaviour of stakeholders of distributed agile software development was influenced by the goals and interests of these stakeholders, hence revealing complex social constructions in place within these projects (Linehan & Kavanagh, 2004). The goals and interests of the development team members were related to the need to reduce risks of software failure and errors, speed up delivery of completed stories, make accurate estimations pertaining to task completion, provide visibility to customers, ease coordination, meet the customers’ requirements and have a versatile team. The customers’ goals and interests related to the need to retain control over the software development process and to obtain a software product which exactly met their business requirements. Similarly to the software development team, management also wanted to promote versatility in the team and to provide high visibility to the customers. For instance, stories were never allocated randomly to team members. Allocation was instead based on careful strategies employed by key players in the team. One interviewee mentioned that “someone will be involved in a particular area of the system in which they’ve got experience”. Such was the case because the participants did not want to waste time and risk having errors in the code. The goal was thus to have stories completed fast and efficiently. The goals and interests varied at different stages in the project and the resulting strategies also varied. Consequently, the practice of defining sets of best practices for agile software development is questionable since these practices emanate from specific goals and interests which emerge at specific points in the project lifecycle. It is thus inappropriate to posit that these derived practices would be applicable to all situations.

As confirmed by Williams and Samset (2010), the differences in agendas, objectives, perceptions and political aspirations of the stakeholders compromised the attempts to align the IT projects and corporate strategies. Cicmil et al. (2006) also mentioned the political agendas of actors during software projects. This study unpacked instances of these political agendas. For instance, findings show that customers or fund providers were inclined to retain control over the software development process. Because of their vested interest in the project and their financial investment, it was hard for them to delegate power to the development team. The development team did not “have control over the development environment. At times the ability to manipulate the web application development and web servers is limited”. In C2, the servers and data resided in South Africa and this caused numerous challenges for the Indian developers because when the server was down, they were unable to complete their work. Consequently, the developers faced environmental constraints (Söderholm, 2008) because of limited access to resources within the development context.

Another goal and interest of the customer in C2 was to have a clear idea of what the developers in India were doing on a daily basis. Visibility was important to them, irrespective of whether the tasks were being completed on time and of quality. Thus, the practice at C2 was adapted according to the customers’ goals and interests. An onsite coordinator was introduced to enhance the communication process and visibility within the dispersed team. As mentioned by one interviewee: “we were discussing how to send the status
and in which format to send the status to the customer so that there was no misunderstanding. And we found that just the daily scrum was not enough for that. So we introduced a coordinator there”.

The customer also retained control by establishing that matters pertaining to lack of performance should not be discussed openly with the team members in retrospective meetings: “Let’s say productivity of this person is not good. [The customers] think that if they discuss directly with that person, he will feel bad”. However, this could also be perceived as a form of double meaning strategies (Bourdieu, 1990) on the part of the customer to maintain confidentiality on some information and to retain the control over the agile software development process. In doing so, the customers retained control over the software development process by withholding information from the development team. Because of these complex social constructions (Sewchurran, 2009), the practices in place contradict the spirit of scrum pertaining to retrospective meetings (Schuh, 2005) and indicate that in reality, the team has to face numerous constraints which are not fully taken into consideration by the methodology.

6.5 The need to retain competitiveness in the market

It was also uncovered that software providers need to remain competitive in the market. Consequently, they strategized and adapted their agile practices so that they always met their customers’ demands. In C2, the developers “didn’t wait for the scrum meeting to inform the customer of any problems that we are facing while we are working on their tasks”. This was a strategy employed by the developers’ to ensure that they were not blamed for hiding things from the customer, who required visibility at all times.

Because of their need to fulfil the customers’ needs, the developers’ goals and interests lay in the completion of the project within budget and on time. In turn, the customers’ also has vested interests in other projects and were at times not fully committed to the project under investigation. They thus did not provide answers to queries fast enough. In these situations, the project teams applied strategies to obtain answers from the customer. For instance, they contacted team members with more symbolic capital to ensure that their queries were answered: “We communicate with the project manager who can in turn push the product manager to give more timely answers”. Otherwise, they sent emails to all stakeholders informing them that their work was on hold until the query was answered. The latter was used to ensure that the developers were not held responsible in case of delayed delivery. Consequently, they did not lose their position in the field. This implies that various forces are at play during agile projects and all parties have to apply various strategies to meet their ends.

6.6 Decision Making

Based on the habitus shared by the agents in the field, it was commonly accepted that final decisions should be taken by top level management (for C1) and the customers (for C2). As mentioned by one interviewee, “the decision is theirs and if they say do it, we do it”. This did not imply that participants did not voice out their opinion when needed: “but we have space to argue about it and expose our opinions”. In C2, decisions on the manner in which agile practices would be adapted and followed appeared to be driven by the need to meet any standards set forth within the team, the organization and industry. For instance, when the need arose to adopt agile methodologies within the organization, management based their decisions on “industry standards and on what agile recommends. They looked at Scrum and XP, interacted with other companies who were pioneers in agile”. This was not the case in C1 where decisions were mostly taken by the COO and the CEO.

6.7 Demands in the field

In both cases, demands in the fields were mostly customer-related. In both cases, one of the demands related to the need to complete as many stories as possible during one sprint. This demand was exerted by the customer who had enough symbolic capital to do so. Consequently, the practice was adapted such that
stories “were completed in parallel. One feature was given to a maximum of two people. If there was dependency between two features then [they] did it sequentially. Otherwise things would not be completed in four weeks”.

According to Scrum, the Scrum Master should be dedicated to the Scrum Master role and should not be involved in coding. However, in C1 the Scrum master was involved in coding. This is because of the workload which had to be completed: “It is hard. It’s quite a bit of work just being Scrum Master. Even in a small team like this, lots of things come up and need to be dealt with. When I was the Scrum Master, I only gave 50% of my time to coding. And I found it to be too much time to give to coding in any case. So, I dropped it down to 30%”. The workload relates to the demands of the field which influenced practice.

In C2, the impact of workload demands could be perceived in the frequency with which review meeting were held. Review meetings were held more frequently in times of urgency, indicating that the participants adjusted their practice based on the demands in the fields. In C2, demands sometimes related to the upcoming of a sprint filled with complex requirements. In that case, the distributed agile teams adapted their practice and had a shorter sprint of two weeks prior to the long one, including all the simple tasks that they could finish quickly in that sprint.

6.8 Constraints in the field

The study showed that distributed agile software development is undertaken in an environment restrained by various constraints which further influence practice. For instance, infrastructure constraints, limitations of communication tools, GSD issues, lack of resources, external environmental constraints and constraints pertaining to the nature of software projects all impact on agile practice.

In both cases, there were insufficient resources in comparison to the amount of work to be completed. Consequently, some team members “were involved in several projects simultaneously”. In C1, participants formed structures with other team members which, in spite of being external to the MKX project, still influenced the distributed agile practice. During task allocation, the fact that team members had to balance their time between MKX and other projects had to be taken into consideration. Also, by being involved with other teams, participants were often unable to fully involve themselves in the project and planning errors were only discovered too late. One participant explained that they are “sometimes flown to Johannesburg for other projects. These are things that [they] only find out about a day or two before. So it’s not something that [they] could plan properly in the sprint”.

In C1, the lack of resources also resulted in the frequent changes in Scrum Master: “all the changes in Scrum Master were linked to who is available and who is not”. By being in other teams, distributed agile team members’ habitus shifted from being fully committed to the MKX project, to being committed to other projects. Hence, it became common practice (or even normal) for the Scrum Master role to be embraced by different team members at different points in time. The frequent changes in Scrum Master were not perceived as counter-productive. Scrum warns that whenever a change or disturbance occurs within the team, it might take about three weeks for things to stabilise. This warning was not considered as a serious threat for the distributed agile team in C1. Instead, they were adapting Scrum as much as possible to fit their contextual requirements.

7. Conclusion

This study embraced a different perspective on distributed agile software development and projects. It was demonstrated that agile software development is innovative in nature and require a high degree of improvisation on a daily basis to achieve project success. It was suggested that software projects are contextually dependent, whereby demands and constraints emerge from the influencing context and further drive agile practices. Such a finding implies that best-practices should be applied carefully and
should not be considered as silver bullets which can solve any problem. The study also contributed to theory by revealing how agendas, objectives and political aspirations of stakeholders within distributed agile software development impact on practice. No studies on distributed agile software development uncovered such a correlation and the study thus contributes to a better understanding of how people chose to adopt or adapt certain practices. The findings are also particularly important to agile project teams seeking to engage in GSD. The study has revealed that, to some extent, collocated and distributed teams face similar challenges. Some practices will nevertheless have to be adapted to the GSD setting. However, the social and contextual elements highlighted in the study could be used as a basis and a lens by such practitioners who will have to adapt their agile practices to a new context. These practitioners will also be aware of the various possible fields of influence in a GSD context, thus allowing them to take preventive actions to meet the respective demands and constraints. The resulting distributed agile practices could, to some extent, be carefully thought through, instead of being a reaction to mistakes being made.

8. References


