

Ten Heuristics from Applying Agile Practices across Different Distribution Scenarios: A Multiple-Case Study

Raoul Vallon¹ & Thomas Grechenig¹

¹ Research Group for Industrial Software, Vienna University of Technology, Vienna, Austria

Correspondence: Raoul Vallon, Research Group for Industrial Software, Vienna University of Technology, Vienna, Austria. Tel: 43-1587-2197. E-mail: raoul.vallon@inso.tuwien.ac.at

Received: February 6, 2016

Accepted: March 29, 2016

Online Published: April 19, 2016

doi:10.5539/cis.v9n2p68

URL: <http://dx.doi.org/10.5539/cis.v9n2p68>

Abstract

Distributed software development (DSD) has become increasingly popular due to benefits such as cost savings, access to large multi-skilled workforces and a reduced time to market. Agile practices can potentially help increase transparency and mitigate communication and coordination issues in these complex environments. While empirical studies in the field exist, most are single-case studies that miss out on the chance to compare different distribution scenarios, which calls for further investigation. We report on results of a four-year exploratory multiple-case study investigating the agile process implementation in three different distribution scenarios: within-city, within-country and within-continent. We purposefully selected the three different cases and found ten common heuristics emerge which are based on empirical evidence in at least two cases as well as four further candidate heuristics that lack evidence in more than one case. In particular, the understanding of and adaptation to each development site's inherent challenges, travelling ambassadors/proxies between sites, and a balanced distribution of decision makers proved to be important heuristics for a successful process implementation.

Keywords: distributed software development, distributed teams, agile, scrum, multiple-case study

1. Introduction

Agile software development is built around empowered and self-organizing teams with a strong focus on collaboration and communication supported by various agile practices including pairing, customer collaboration, stand-ups, reviews, retrospectives and the planning game (Šmite et al., 2010). Developing software globally is a daily reality for many organizations. The benefits of global software development include cost savings, access to large multi-skilled workforces and reduced time to market, among others (Ó Conchúir et al., 2009). Agile software development may potentially improve collaboration in distributed environments as it relies strongly on frequent communication (Hossain et al., 2011a). However neither the leading agile process scrum (Schwaber & Beedle, 2002) nor eXtreme Programming (XP) (Beck, 2000) was designed for distributed teams working in distributed environments. Hence adaptations to the original process are necessary (Batra, 2009). The goal of these process adaptations is to transfer agile values, which produced excellent results in the last decade for collocated teams (Dingsøyr et al., 2012), to distributed software development (DSD) environments.

This study provides the following contributions to the evolving research field of applying agile process in DSD environments:

1. Empirical evidence from a long-term four-year multiple-case study
2. Cross-case analysis of agile practices among these three different distribution scenarios (within-city, within-country and within-continent distribution)
3. Addressing the need for robust primary empirical studies researching DSD (Marques et al., 2012) and agile practices in DSD in particular (Hanssen et al., 2011)

The remainder of the paper is organized as follows. Section 2 discusses our research objective with regard to previously reported work. The detailed multiple-case research design is explained in Section 3. Section 4 presents each case individually. Section 5 analyzes cross-case results and describes limitations of the study. Section 6 offers conclusions and an outlook to future work.

2. Research Objective in Light of Previous Work

There are various systematic literature reviews covering (globally) distributed software development (Verner et al., 2012; Marques et al., 2012). Most relevant to our line of research are the ones focusing on agile practices in DSD environments (Hossain et al., 2009; Jalali & Wohlin, 2011; Hanssen et al., 2011). We were especially interested in the multiple-case studies conducted in the field and looked at all relevant studies in the systematic review of Jalali and Wohlin (2011) (who cover years 1999-2009) and extended the analysis to years 2010-2014. We found seven multiple-case studies for the years of 2010-2014 as compared to three multiple-case studies for 1999-2009 (Jalali & Wohlin, 2011) which indicates an increasing research focus and interest on agile practices in DSD. Jalali and Wohlin (2011) showed that context is not richly described in empirical studies in the area of agile DSD. We further investigated how context has been described in past studies and built a conceptual framework (cf. Figure 1) to use for our multiple-case study and address the identified shortcoming. Moreover, in previous multiple-case studies we found methodological triangulation to be scarce. The most often used approach was semi-structured/open-ended interviews (Ramesh et al., 2006; Sison & Yang, 2007; Paasivaara et al., 2009; Srinivasan & Lundqvist, 2010; Paasivaara, 2011; Hossain et al., 2011a; Bass, 2012; Paasivaara et al., 2012; Ramesh et al., 2012; Badampudi et al., 2013), followed by document analysis (Sison & Yang, 2007; Hossain et al., 2011b; Ramesh et al., 2012) and observation (Hossain et al., 2011b). This finding supports the claim (Marques et al., 2012) that there is a need for robust primary empirical studies researching DSD and agile practices in DSD in particular (Hanssen et al., 2011).

We identified the following research gap: *As context information in past empirical studies is often not richly provided, it is hard to generalize from past studies in the field. This study aims to investigate three cases implementing agile practices in DSD in significantly different distribution scenarios and presents rich contextual information for each case respectively. The nature of our multiple-case study is exploratory in the way that it has no clear, single set of outcomes (Yin, 2003) and that it is to the best of our knowledge unprecedented in the approach of analyzing the emergence of common heuristics (Heeager & Rose, 2014) in several distribution scenarios (within-city, within-country and within-continent).*

Based on that objective, our main research question (RQ) is: Which heuristics, if any, can we see emerge when applying agile practices in different distribution scenarios? Within this research question, we define the following sub-questions to guide our research:

RQ1. *How have agile practices been applied in the different distribution scenarios?*

RQ2. *Which DSD practices have been used to complement agile practices?*

3. Research Context and Study Design

This multi-case study is a major step in a several-year research project aimed at empirically investigating the application of agile practices in DSD. As such it adds to the empirical basis of the research field and represents an important link between our initial systematic mapping (Vallon et al., forthcoming) and a comprehensive process framework for agile DSD yet to be developed in a future step.

Our research design follows the guidelines of Yin (2003) for general case study design and Verner et al. (2009) for conducting multiple-case studies in software engineering in particular. Furthermore we based our case study protocol on the template by Brereton et al. (2008) and the decisions along the way are embedded in this section.

3.1 Multiple-Case Study Design

We chose a case study design as a natural fit to our research problem as it “investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident” (Yin, 2003, p.13). Moreover, we want to focus on covering contextual background for which the case study is especially well-suited (Yin, 2003). Evidence from multiple cases is generally considered more compelling (Herriott & Firestone, 1983). Our unit of analysis is a project within an organization for which agile processes are applied in a distributed development environment, i.e. the development itself has to take place on at least two sites. Hence, our multiple-case study is an embedded case study focusing on the output of individual projects as compared to a holistic one (Yin, 2003). For selecting information-rich cases we used purposeful maximum variation (heterogeneity) sampling (Patton, 2002), i.e. we chose heterogeneous cases based on the characteristics developed in our conceptual framework (cf. Figure 1). Since scrum is the most widely used agile process also in DSD (Jalali & Wohlin, 2011) we decided to limit our case selection to scrum and choose the distribution scenario of the project as the primary dimension for selecting heterogeneous cases. The final selection involved three cases applying scrum practices in greatly different distribution scenarios: city-wide, country-wide and continent-wide. The assumed problem that individual cases are too different can be

intentionally turned into strength: “Any common patterns that emerge from great variation are of particular interest and value in capturing the core experiences and central, shared dimensions of a setting or phenomenon” (Patton, 2002, p.235). First, each case is treated individually with an individual case report before the cross-case analysis (Yin, 2003).

3.2 Conceptual Framework

Based on our review of previous work in Section 2, we defined a conceptual framework with the key factors that we will focus on for our data collection (Verner et al., 2009). We base our conceptual framework on three main factors relevant to the context of our empirical study: DSD inspired by (Šmite et al., 2012; Jalali & Wohlin, 2011), scrum practices extraction inspired by (Paasivaara et al., 2009; Hossain et al., 2011b) and general information with regard to our unit of analysis (project) inspired by (Jalali & Wohlin, 2011; Hossain et al., 2011b). Figure 1 drafts our conceptual framework with all identified relevant empirical factors.

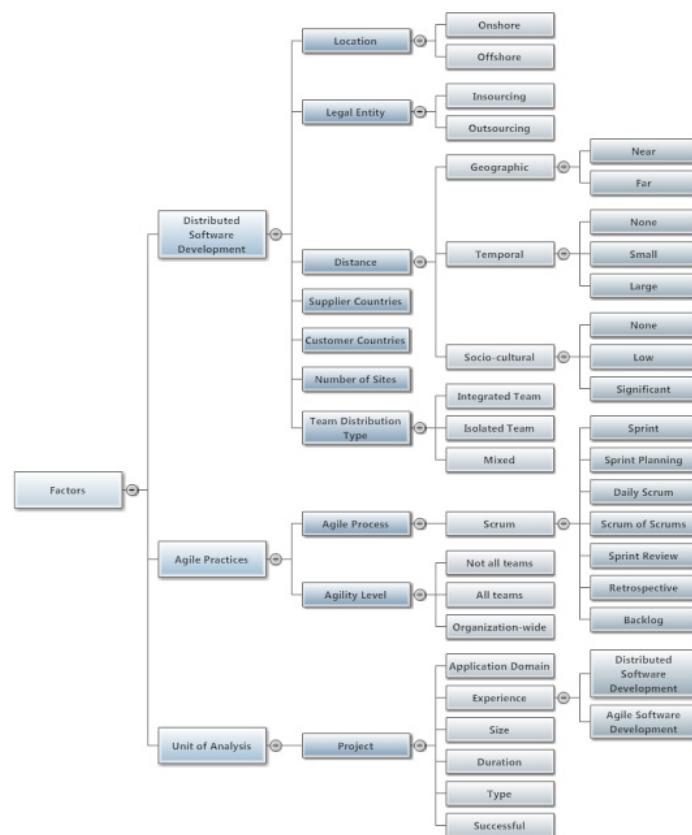


Figure 1. Conceptual framework developed from literature

3.3 Case Organizations

We purposefully selected three heterogeneous cases with regard to their varying distribution scenarios. Identities are withheld to preserve privacy and the three projects receive pseudonyms which will be used throughout the paper: *WithinCity* (sites within the same city), *WithinCountry* (sites within the same country) and *WithinContinent* (sites within the same continent). Table 1 shows an overview of the contextual factors *DSD* and *unit of analysis (project)*. The remaining contextual factor *agile (scrum)* is discussed during case analysis.

3.4 Data Collection

The data collection strategy is aimed at finding out which scrum practices organizations have applied to distributed projects in what way (RQ1), and which DSD practices have been used to complement the agile process (RQ2).

Our multiple-case study is qualitative and triangulation was a major concern because each method reveals different aspects of empirical reality (Denzin, 1978). Denzin (1978) has identified four types of triangulation:

Table 1. Contextual information on the selected cases

Factors	Sub-factors	Case WithinCity	Case WithinCountry	Case WithinContinent
DSD	Location	Onshore	Onshore	Offshore
	Legal Entity	Insourcing	Outsourcing	Insourcing
	Geographic Distance	Close	Near	Close
	Temporal Distance	None	None	-
	Socio-cultural Distance	None	None	Low
	Supplier Country	Austria	Austria	Austria and two other European Countries
	Customer Country	Austria	Germany	Austria
	Number of sites	2	2	3
	Team Distribution Type	Integrated Teams	Integrated Teams	Isolated Teams
Project	Application Domain	Web & Hardware	Enterprise Software	Web
	Experience with Agile	7+ years	3 years	3+ years
	Experience with DSD	10 years	2 years	15+ years
	Project Size	19	30	39
	Team Size	Site A (13), Site B (6)	Site A (20), Site B (10)	Site A (14), Site B (19), Site C (6)
	Project Duration	15 months	6 months	9 months
	Project Type	Industry	Industry	Industry
	Successful	Yes	Yes	Yes

Table 2. Different types of triangulation in the three cases

Cases	Method and Data Triangulation	Investigator and Theory Triangulation
WithinCity	Participant-Observation (1 Action researcher) Documentation (160 documents) Archival Records (3863 tickets in issue tracking system, 274 wiki pages) Physical Artifacts (thousands of sticky notes and dozens of paper boards)	Vallon (Primary Investigator) +1 Senior Researcher +2 Supporting Investigators
WithinCountry	Interviews (N=7) Direct Observation (5 meetings across several sprints) Documents (15 documents) Archival Records (579 tickets, 37 wiki pages) Physical Artifacts (thousands of sticky notes and dozens of paper boards)	Vallon (Primary Investigator) +1 Senior Researcher +3 Supporting Investigators
WithinContinent	Interviews (N=11) Documents (273 documents) Archival Records (only limited view)	Vallon (Primary Investigator) +1 Senior Researcher +4 Supporting Investigators

Data triangulation (variety of data sources), investigator triangulation (use of different researchers), theory triangulation (multiple perspectives to interpret a single set of data achieved by using multiple investigators (Stake, 1995)) and methodological triangulation (multiple methods to study a single problem). Table 2 shows how triangulation was achieved for each of the respective cases. Since triangulation is expensive, we employed it reasonably and practically (Patton, 2002) within the possibilities and limitations of each case. The triangulation sources are used as described by Yin (2003). With one exception, all supporting investigators were only involved in one case of the multiple-case study in an effort to minimize bias. The authors served as investigators in all three cases.

3.5 Data Analysis

Figure 2 illustrates our data analysis process. In case WithinCity we used Action Research (AR) as the primary research approach. AR uses “a spiral of steps, each of which is composed of a circle of planning, action, and fact-finding about the result of the action” (Lewin, 1946, p.38). Researchers and participants collaborate to meet respective goals. We had one researcher participate in the action research assuming the role of a scrum master. The supporting investigators and senior researcher (who were not participating in the action research on site) analyzed new data, discussed findings in regular research meetings each sprint with the action researcher and corroborated findings with documents and archival records. Overall the action researcher engaged in the following activities: explore data in ticket management tool, write notes in project diary, discuss impediments with practitioners and record actions and track their results, discuss and analyze the data collected with the off-site supporting investigators and senior researcher (each sprint).

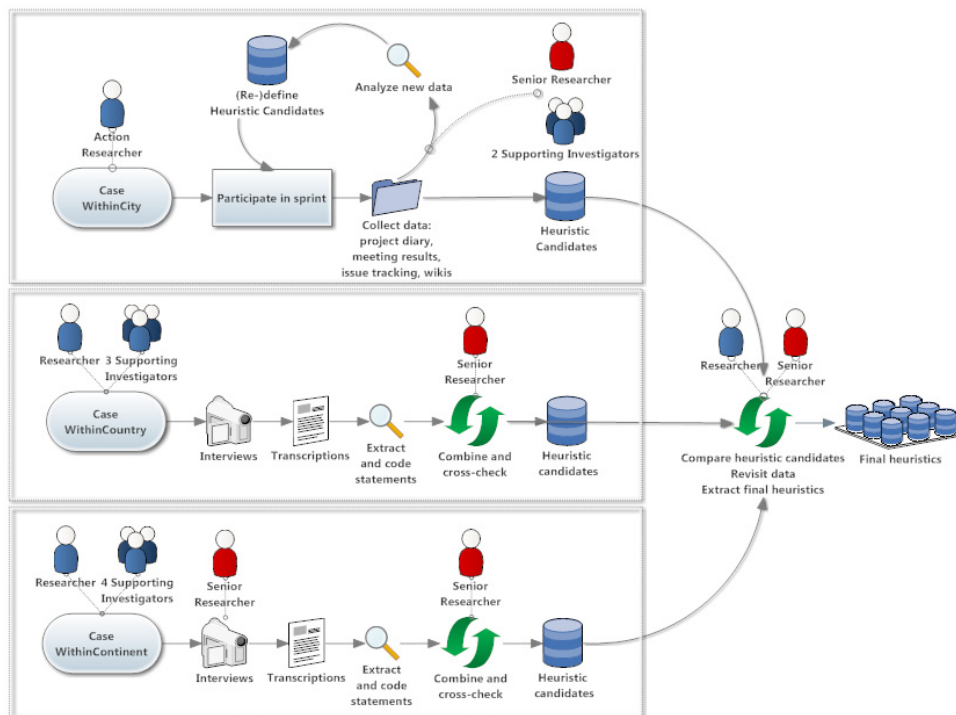


Figure 2. Research methodology of the three individual cases and the cross-case analysis

For cases WithinCountry and WithinContinent we used semi-structured interviews (Patton, 2002) which allow a conversational manner but still follow an interview protocol (Yin, 2003). The interviews were recorded and later transcribed and lasted from 0.5-3 hours (on average 100.5 minutes). We interviewed all different roles at least once for each case and also involved stakeholders. For analysis we applied grounded theory to start with empirical specifics and move towards general statements (Denzin & Lincoln, 2011). To extract findings we applied open coding (Strauss & Corbin, 1998), where the researcher generates categories fitting the data in relation to a general issue of concern (Bryman & Burgess, 1994). The process has been facilitated by using the ATLAS.ti qualitative data analysis software. In case WithinContinent the interview transcripts were additionally formally approved by the case organization before they were used for analysis. In case WithinCountry we also employed direct observation and observed the following meetings: several daily scrums, scrum of scrums, planning, review and retrospective.

For cases WithinCity and WithinCountry we were also able to collect physical artifacts in form of sticky notes and paper boards and extensively collected documents and archival records to triangulate findings. For WithinContinent we also collected documents but were only allowed limited access to archival records. We also held feedback sessions in each case with the participating organization to validate our theory.

4. Individual Case Results

This section presents results of the individual cases by reporting the background (context), challenges and agile practices specific to the case and the identified *heuristic candidates* for the cross-case analysis.

4.1 Case within City

4.1.1 Background

This case covers software development within an organization spread across two sites in Vienna, Austria. Before the beginning of the case study, the development of the product had started as a research and development (R&D) effort on one site for two years aiming at establishing a set of technologies covering both hardware and software. In the herein analyzed 15-month case study time frame, the goal was to turn the R&D prototype into a deliverable product. With the beginning of full-scale product development, the team size doubled and project personnel were distributed across two sites. The team members were split into a development and a test site forming fully distributed cross-functional teams (Sutherland et al., 2007), i.e. teams integrated across both sites. We analyzed 27 two-week sprints over the case study period.

4.1.2 Challenges

The initial process implementation suffered from a strong focus on the former R&D site, the development site, since the testing site was a new addition to the process after the initial two-year R&D phase. The development site used isolated sprints without involving the test site to the end that developed stories in one sprint had to be tested in the next one by the test site. Furthermore, the teams had to deal with technically complex implementations regarding communication between hardware and software and decided to split in closely collaborating on-site micro teams with 3-4 team members. The process implementation exhibited massive problems: Stories were accepted at the end of the sprint without having been tested, which was done in the next sprint. This behavior led to a great number of features with low quality and resulted in a failed shipment to the customer in sprint 7, which was an eye-opener and the test site was then included properly as part of the same sprint. The micro teams were extended to 2-3 developers and one (remote) tester. The amount of contact visits to the other site improved, especially towards the end of each sprint, making sure to deliver high quality stories. Greater effort was put into realistic planning and commitment with the customer shipment always in mind. The retrospective turned out to be a great means to drive continuous improvement and manage frustration with the current process implementation by allowing everybody to voice their opinion.

4.1.3 Agile Practices

Sprint: Two-week iterations were used which were on few occasions prolonged to cope with holidays and customer's deadlines. **Sprint planning:** A joint sprint planning was held in person at the development site with all developers and one or two testers (ambassadors) to represent the testing site. The ambassador(s) then travelled back to the test site to discuss the planning results. **Daily Scrum:** This project environment worked with very closely collaborating micro teams and the practice of daily scrums, although practiced in the beginning, was eventually dropped. The testers contacted the developers directly for information and updates when needed. Vice versa the developers tried to give a heads-up to the testers when possible. Both formal (ticket management system and emails) and informal (instant messaging, chats and phone calls) were extensively used between the two sites to compensate the lack of face-to-face communication. **Scrum of Scrums:** Although there were several micro teams, no scrum of scrums has been used as it was regarded as an overhead and the communication mechanisms in-place sufficed. **Sprint review and retrospective:** These meetings were held in a similar fashion as the sprint planning, at the developer's site with one or more ambassadors from the testing site present. For the retrospective, the ambassador(s) made sure to collect positive and negative comments from all members of the testing site in advance and presented them at the retrospective. **Backlog:** As the customer preferred to work with milestones, there was a rough set of user stories planned for each milestone, a generally two-month time frame. The backlog planning relates to the process described by Hong et al. (2010): The roadmap planning was done for milestones and the detailed planning was done for sprints. Naturally with agile development, the roadmap was subject to change as stories were implemented by priority sprint after sprint.

4.1.4 Heuristic Candidates

HC1.1: Fully distributed cross-functional teams need to synchronize and work towards the same goal in a sprint. Synchronization can be achieved with contact visits, emails, phone calls, chats, instant messaging and an up-to-date ticket management system. HC1.2: Informal face-to-face communication needs substitution in DSD. Tool support can help keep track of changes in requirements. Instant messaging was particularly capable to substitute some of the informal face-to-face coordination unavailable in DSD. HC1.3: Frequent deployments to

the customer set a common goal across all sites and thus enable more accuracy for planning and commitment and better quality of delivered stories. HC1.4: Documentation needs to be updated timely as team members need to rely on it more than in collocated settings. Still it cannot be used to substitute frequent interaction. HC1.5: Results from informal on-site enquiries or meetings need to be accessible to other distant team members. Contact visits can further be used to review test cases for critical stories. HC1.6: Keeping the retrospective as a constant in an ever-changing process environment proved to be a powerful means. The retrospective meeting gathers members from all sites to reflect on the process and discuss improvements. It further functions as an important mood barometer across all sites. HC1.7: Process adaptation in DSD is slower and more difficult to achieve than in regular collocated scrum.

4.2 Case within Country

4.2.1 Background

This case covers two unaffiliated organizations, a main supplier (MainSupp) and an additional supplier (AddSupp), co-developing three product variations of the same code base, resulting in three product owners. Both suppliers have successfully applied regular scrum before and chose to implement an adapted version of scrum to better suit the needs of a DSD environment. The two organizations develop at their own sites in two different cities in Austria, separated by about a four-hour car ride. MainSupp is a large organization whose IT department is involved in the development of the three software products. It acts as point of contact to customers and provides the bigger part of the development staff. AddSupp is a medium-sized core software development company and a subcontractor to MainSupp for the software development. It complements the MainSupp's development with additional staff and know-how. The teams are all integrated (cross-site) distributed ones (Sutherland et al., 2007).

4.2.2 Challenges

Although both companies had previous experience with applying regular scrum successfully, both lacked experience in DSD. Transparency was a big issue between the two suppliers and low quality video conferences and little available documentation for AddSupp handicapped communication and coordination in the first months. There was no high level overview of the progress of all three teams available to everyone since paper scrum boards and burndown charts were used. All three scrum teams were staffed by members of both suppliers, yet all product owners and scrum masters were based on the MainSupp's site. The resulting coordination issues are best described in the words of one of the AddSupp's developers: "I would love to break down tasks to a decent level, but if we do not know what should be developed exactly, that is hard to achieve". The situation eventually improved with heavy use of video conferencing to complement the scrum process meetings.

4.2.3 Agile Practices

Sprint: Two-week sprint iterations were used with reviews every sprint and planning and retrospective meetings only every other sprint. Sprint planning is a two-tiered process and covers two sprints. The first tier involved planning at the MainSupp's site with one ambassador from AddSupp present. The second-tier planning continued at the AddSupp's site. The ambassador returned with pre-estimated user stories which were then broken down into tasks by AddSupp's developers. When a developer accepted a task, he adjusted the original estimation of the MainSupp to his own. An updated planning spreadsheets was then returned to MainSupp. **Daily Scrum:** Each scrum team held a daily video conference meeting, where respective team members of the MainSupp and AddSupp participated. **Scrum of Scrums:** One of the AddSupp's developers travelled to the MainSupp's site once a week for face to face updates and discussions. Both sites also engaged in their own intra-site coordination scrum of scrum right after the daily scrum. The testers of each team also felt the need to coordinate across all teams in their own scrum of scrums. The sprint review was held jointly for all the distributed teams. It was primarily held at the MainSupp's site with one or two "proxies" from AddSupp on site. In contrast to the sprint planning, for the review and retrospective, AddSupp joined directly via video conference. The review consisted of story demonstrations and discussions about different areas of the current product increment. The retrospective followed the review in the same setup, but only every other sprint. **Backlog:** Each product owner maintained a product backlog on the MainSupp' site for his product. AddSupp worked only with the sprint backlog which was a planning spreadsheet created during the two-tiered planning process.

4.2.4 Heuristic Candidates

HC2.1: Establish several means of both formal and informal communication as a substitute for missing face-to-face communication. HC2.2: A travelling ambassador/proxy can help to build trust, exchange information and discuss critical issues or impediments face to face. HC2.3: Beware of a superficial adoption of

agile values, even more so in distributed settings where several sites rely on a working flow of information/communication and transparency. HC2.4: In agile processes all sites should be as equal as possible under the given constraints (such as a subcontractor relationship) as the process does not scale well with hierarchical sites and the withholding of information. HC2.5: Tool choices are secondary as long as everybody is committed to using the same tool. Organizational impediments regarding electronic tool usage may be circumvented with paper boards.

4.3 Case within Continent

4.3.1 Background

This case covers a customer based in Austria and a supplier based in both Austria and two further European countries, which are withheld for privacy reasons. The project had a rushed start because of a tight schedule and deadline. Moreover it was one of the first projects to be done in this setup and there were organizational restrictions on the supplier's side to use the V-model (Boehm, 1979) also in this distributed development. Still, agile practices were eventually used in an effort to improve collaboration among the three development sites. The project spanned 9 months. There were no time zone issues and cultural issues can be regarded as minor between Austria and European countries due to their proximity. Since the overall process was the V-model, there was no scrum master in place, yet several PMO roles such as project manager, test manager, solution architect and change manager.

4.3.2 Challenges

We identified three problem categories in this case, the inflexibility of the V-model, weak feedback loops and collaboration with the customer and further intra-supplier issues. The V-model was implemented because the supplier had many years experience with it, but the project had been under a very tight schedule from the very beginning and the V-model did not allow enough flexibility to cope with unforeseen problems (both technical and organizational in nature). The supplier's internal problems, being distributed across three sites, were in the end successfully mitigated by employing several agile meetings such as daily scrum and daily scrum of scrums that helped bring the project back on track. The customer collaboration could only be improved up to a certain point, because it is not an integral part of the V-model. The customer felt left out of feedback loops and was not fully content with the final product. A mitigation strategy was to make a dashboard available to the customer for live test reports, but it only came late in the project's timeline. This project also served as a ramp-up for future collaborations and as such was a great learning experience for all parties. The customer and the supplier decided to alter the process for future projects in favor of the implementation of more agile practices.

4.3.3 Agile Practices

There was no sprint, sprint planning or backlogs in use as the V-model worked with milestones and formal reviews and a fixed set of requirements with little flexibility other than change requests. Still, a few agile practices were implemented as a crucial improvement to the development collaboration: there was a 15 minutes daily scrum within teams and another daily scrum of scrums including development lead, PMO, solution architect and test manager. Furthermore there were also weekly meetings between development lead of European country #1 and specialists from Austria and European country #2 and two-weekly meetings of all teams and stakeholders to spread knowledge on project's status. In short, communication was very important for the supplier internally but not towards the customer. There was no retrospective meeting, only a one-time lessons-learned workshop after the completion of the project.

4.3.4 Heuristic Candidates

HC3.1: Also in traditionally structured environments agile practices can help establish more frequent communication and thus have a positive output on the quality of software delivery. HC3.2: Understand each other's (customer's and supplier's) internal processes and find a compromise that works for both sides. HC3.3: Define roles clearly, make them known and act on them. This regards both the customer-supplier communication and the supplier's internal communication (among different sites). HC3.4: Shutting the customer out does not build quality in as different understanding of requirements is inevitable that will be spotted very late in the project. HC3.5: Provide environments for both formal and informal means of communication and try to break language barriers by offering language courses and keep documentation in a language acceptable to all parties. HC3.6: Promote contact visits for team building and face-to-face information sharing. Occasional team events to bring together teams on one site may also be beneficial for establishing trust. HC3.7: Try to equally distribute decision makers among different sites to facilitate the flow of information.

Table 3. Cross-case summary of the implemented scrum practices with highlighted cells indicating a modified usage for DSD

Agility Level	WithinCity All teams	WithinCountry All teams	WithinContinent Not all teams
Sprint	Used (two-week)	Used (two-week)	Not Used
Sprint Planning	Used modified (joint sprint planning held at the development site with all developers and one or two ambassadors to represent the other site)	Used modified (two-tiered process, held at main site with all developers and an ambassador from other site, followed by a continued planning effort on the other site with returning ambassador)	Not Used
Daily Scrum	Not used (due to highly collaborative micro teams, successfully substituted with several other formal and informal means of communication)	Used modified (daily scrum in video conference)	Used (daily scrum within teams)
Scrum of Scrum	Not used	Used modified (ambassador travels to main site for scrum of scrums, both sites engage in their own cross-team meetings; also a separate cross-site tester's scrum of scrums was held)	Used modified (daily with development lead, PMO, solution architect and test manager; on top of that also a weekly cross-site specialist meeting and two-weekly global all-site meetings)
Sprint Review	Used modified (same setup as sprint planning)	Used modified (joint sprint review held at the main site with all developers and one or two proxies to represent the other site)	Not Used
Retrospective	Used modified (same setup as sprint planning)	Used modified (same setup as sprint review, but only every other sprints)	Not Used (only lessons learned workshop after project's end)
Backlog	Used (coarse-grained product backlog with "milestones" in accordance with customer and a detailed sprint backlog)	Used modified (the main site groomed the product backlog, the other site only worked with the sprint backlog)	Not Used

5. Results of Cross-Case Analysis

With our multiple-case study research approach we are looking for two kinds of findings (Patton, 2002, p.235):

- High quality, detailed descriptions of each case, useful for documenting uniqueness (Section 4)
- Important shared patterns that cut across cases and derive their significance from having emerged out of heterogeneity (Section 5)

Figure 2 presented our analysis process including the cross-case analysis. Table 3 shows how scrum practices have been applied in three different cases (RQ1) and what DSD practices have been used to enhance the collocated practices (RQ2).

5.1 Agile Practices

5.1.1 Sprint

Cases WithinCity and WithinCountry used two-week sprints, while WithinContinent worked with a V-model and thus did not use iterations. No DSD methods were applied in any of the cases to alter the sprint practice.

5.1.2 Sprint Planning

Cases WithinCity and WithinCountry applied a similar approach using an ambassador and focused the planning physically on one site only. In case WithinCountry the other site also held another (second-level) planning following the return of the ambassador. Case WithinContinent worked with a V-model and up-front heavy-weight requirements and planning. The DSD enhancement of adding a travelling ambassador worked well in both cases and was a substantial improvement for a working scrum process.

5.1.3 Daily Scrum

WithinCity worked in micro teams and dropped the practice in favor of using several other means of formal and informal communication (ticket management system, phone calls, emails, chat, instant messaging and a wiki). WithinCountry and WithinContinent both implemented the practice of a daily scrum, for case WithinCountry with the help of video conferences (integrated distributed teams) and on-site for case WithinContinent (isolated distributed teams).

5.1.4 Scrum of Scrums

WithinCity also did not use scrum of scrums due to the same rationale as not using daily scrums. WithinCountry used scrum of scrums for on-site inter-team coordination. WithinContinent applied several scrum of scrums for cross-team and cross-site coordination by means of video conferencing and screen sharing sessions.

5.1.5 Sprint Review and Retrospective

These two practices have been applied in the same setup within the respective case of WithinCity and WithinCountry with the introduction of a travelling proxy/ambassador similar to the sprint planning acting on behalf of the colleagues not present in case WithinCity and serving as a proxy (with the team joining in video conference) in case WithinCountry. WithinContinent used a V-model with its respective phases and reviews.

5.1.6 Backlog

Case WithinCity used a product backlog with coarse-grained low-priority stories and fine-grained high-priority ones, planned for the next “milestone”, usually a time span of about 4-5 sprints, which would then each have a regular sprint backlog. Case WithinCountry had the product backlog handled by the main site (as consequence of all the product owners residing there) and handed only the sprint backlog to the additional site for co-development by both sites. No DSD practices have been used to facilitate this practice other than ticket management systems. WithinContinent used a V-model with a pre-defined release plan and no backlog practices.

5.1.7 Summary of DSD Enhancements

In our multiple-case study the following DSD practices supported the application of a scrum practices in a distributed environment: Contact visits by a travelling proxy/ambassador (sprint planning, review and retrospective), different types of formal and informal means of communication such as video conferences, phone calls, chat, emails, screen sharing sessions, ticket management systems and wikis (sprint planning, review, retrospective, daily scrum, scrum of scrums, backlog) in order to mitigate the lack of face-to-face communication in DSD environments.

5.2 Final Theory: Ten Heuristics and Four Candidates

Table 4 shows a summary of all agile heuristic candidates as identified during the cases. The table also notes the source (case) for each heuristic and thus describes how the final heuristics have been derived from the heuristic candidates. Heuristic candidates that did not gain empirical support in more than one case retain candidate status but are listed for the sake of completeness.

We can see that ten heuristics H1, H3, H4, H5, H6, H7, H8, H9, H10 and H12 emerged from the heterogeneous cases with empirical evidence in at least two cases. The themes of these ten heuristics can be summarized as frequent formal and informal communication (H1), necessity of up-to-date documentation (H3), informal information sharing across sites (H4), retrospective as driver for continuous improvement (H5), slower process adaptation (H6), travelling ambassador/proxy (H7), risk of a superficial scrum adoption (H8), equal sites with a balanced distribution of decision makers (H9), tools not dictating a sub-par process implementation (H10) and understanding each other's internal processes (H12).

5.3 Limitations

Like any empirical study our study exhibits certain threats to validity (Yin, 2003). The generalizability of the results of our multiple-case study is limited in light of its limitations. We addressed *construct validity* by using multiple sources of evidence, a chain of evidence and had respective informants validate our results for each of

the three cases in separate feedback sessions. To achieve *internal validity* we employed method, data, investigator and theory triangulation. For *external validity* we provided a theoretical framework deduced from previous work as a basis for the analysis of the three cases. We established *reliability* by following a case study protocol, as described in this report, a case study data drive, and the data analysis software ATLAS.ti for consistent handling. The great difference in the cases' context was purposefully introduced via maximum variation (heterogeneity) sampling with the objective of finding that "a theme song emerged from all the scattered noise" (Patton, 2002, p.235). While multiple-case studies provide more value for generalization than single-case studies, empirical evidence from other related studies needs to be systematically analyzed, which we aim to address in the next step of our research. Until then, the heuristics should be regarded as specific to the individual case's context and thus generalized with caution.

Table 4. Agile DSD heuristics emerging from the multiple-case study: the dot (●) indicates that empirical support was found after revisiting data but that there was no initial heuristic candidate and heuristics with an asterisk (*) are candidates

ID	Heuristics from Applying Agile Practices in DSD Environments	Within City	Within Country	Within Continent
H1	Fully distributed cross-functional teams need to synchronize and work towards the same goal in a sprint. Both formal and informal face-to-face communication needs substitution in DSD by means of tool support, contact visits, video conferences, phone calls, instant messaging, emails, chat, wiki, an up-to-date ticket management system and test case or code review. Try to break language barriers by offering language courses and keep documentation in a language acceptable to all parties.	HC1.1 HC1.2	HC2.1	HC3.5
H2*	Continuous customer deployments can help establish focus in DSD and lead to more realistic sprint planning and commitments. Continuous flow means more reliable estimations and higher software quality.	HC1.3		
H3	While up-to-date documentation does not substitute direct interaction, it plays a bigger part in agile DSD than in on-site scrum since direct communication is harder to achieve.	HC1.4	●	●
H4	Results from informal on-site enquiries or meetings need to be made available to other sites, either in terms of updated documentation or updated tickets in the electronic ticket management system.	HC1.5	●	
H5	Keeping the retrospective as a constant in an ever-changing process environment proved to be a powerful means. The retrospective meeting gathers members from all sites to reflect on the process and discuss improvements. It further functions as an important mood barometer across all sites.	HC1.6	●	
H6	Process adaptation in DSD is slower and more difficult to achieve than in regular collocated scrum.	HC1.7	●	●
H7	A travelling Ambassador/Proxy can help to build trust, exchange information and discuss critical issues or impediments face to face. Promote contact visits for team building and face-to-face information sharing. Occasional team events to bring together teams on one site may also be beneficial to establishing trust.	●	HC2.2	HC3.6
H8	Beware of a superficial adoption of agile values, even more so in distributed settings where several sites rely on a working flow of information/communication and transparency.		HC2.3	●
H9	In agile processes all sites should be as equal as possible under the given constraints (such as a subcontractor relationship) as the process does not scale well with hierarchical sites and the withholding of information. Try to equally distribute decision makers among different sites.	●	HC2.4	HC3.7

ID	Heuristics from Applying Agile Practices in DSD Environments	Within City	Within Country	Within Continent
H10	Tool choices are secondary as long as everybody is committed to using the same tool. Organizational impediments regarding electronic tool usage may be circumvented with paper boards.	•	HC2.5	
H11*	Also in traditionally structured environments agile practices can help establish more frequent communication and thus have a positive output on the quality of software delivery.			HC3.1
H12	Understand each other's (customer's and supplier's) internal processes and find a compromise that works for both parties.	•	•	HC3.2
H13*	Define roles clearly, make them known and act on them. This regards both the customer-supplier communication and the supplier's internal communication (among different sites).			HC3.3
H14*	Shutting customer out does not build quality in as different understanding of requirements is inevitable that will be spotted very late in the project.			HC3.4

6. Conclusion

In this paper we presented the results of our multiple-case study spanning an overall timeframe of four years. Through careful analysis we identified ten heuristics with empirical evidence in at least two cases and four heuristic candidates with support in only one case. The heuristics concern the application of agile practices in distributed development environments which evolved from three different distribution scenarios:

- Case WithinCity: sites distributed within one city, spanning two districts
- Case WithinCountry: sites distributed within one country, spanning two cities
- Case WithinContinent: sites distributed within one continent, spanning three countries

Although agile methods are increasingly being adapted to distributed software development, they are no panacea for success and need to be carefully tailored to each individual distributed environment. With our research we strive provide empirical evidence for other researchers to build upon and to help the practitioner find advice on tailoring agile methods to distributed environments. To this end, our future work will include the improvement and enhancement of the heuristics, supported by a full-scale systematic literature review, with the goal of accumulating empirical evidence for designing a process framework to improve the chances of a successful implementation of agile practices in today's complex distributed software development environments.

Acknowledgments

This research was partly funded by the Austrian Marshall Plan Foundation (<http://www.marshallplan.at>) by providing a scholarship to the first author. The research report has been finalized during a research visit at the Center for Design Research at Stanford University. The first author thanks Professor Larry Leifer for hosting the research visit. The authors thank the participating organizations as well as the supporting investigators/researchers throughout the multiple-case study for their assistance in conducting the study and fruitful discussions.

References

- Badampudi, D., Fricker, S. A., & Moreno, A. M. (2013). Perspectives on Productivity and Delays in Large-Scale Agile Projects. *Agile Processes in Software Engineering and Extreme Programming*, 180–194. http://dx.doi.org/10.1007/978-3-642-38314-4_13
- Bass, J. M. (2012). Influences on Agile Practice Tailoring in Enterprise Software Development. *Proceedings of the 2012 Agile India Conference*. <http://dx.doi.org/10.1109/agileindia.2012.15>
- Batra, D. (2009). Modified agile practices for outsourced software projects. *Commun. ACM*, 52(9), 143. <http://dx.doi.org/10.1145/1562164.1562200>
- Beck, K. (2000). *Extreme programming eXplained*. Reading, MA: Addison-Wesley.
- Boehm, B. (1979). *Guidelines for Verifying and Validating Software Requirements and Design Specifications* (Technical Report). University of Southern California, Redondo Beach, California.
- Brereton, P., Kitchenham, B., Budgen, D., & Li, Z. (2008, June). Using a protocol template for case study

- planning. *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. University of Bari, Italy.
- Bryman, A., & Burgess, R. (1994). *Analyzing qualitative data*. London; UK: Routledge.
- Denzin, N. (1978). *The Research Act: A Theoretical Introduction to Sociological Methods* (2nd ed.). New York: McGraw-Hill.
- Denzin, N., & Lincoln, Y. (2011). *The Sage handbook of qualitative research*. Thousand Oaks, Calif.: Sage Publications.
- Dingsøy, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213–1221. <http://dx.doi.org/10.1016/j.jss.2012.02.033>
- Hanssen, G. K., Ššmite, D., & Moe, N. B. (2011). Signs of Agile Trends in Global Software Engineering Research: A Tertiary Study. *Proceedings of the 2011 IEEE Sixth International Conference on Global Software Engineering Workshop*. <http://dx.doi.org/10.1109/icgse-w.2011.12>
- Heeager, L. T., & Rose, J. (2014). Optimising agile development practices for the maintenance operation: nine heuristics. *Empir Software Eng*, 20(6), 1762–1784. <http://dx.doi.org/10.1007/s10664-014-9335-7>
- Herriott, R. E., & Firestone, W. A. (1983). Multisite Qualitative Policy Research: Optimizing Description and Generalizability. *Educational Researcher*, 12(2), 14–19. <http://dx.doi.org/10.3102/0013189x012002014>
- Hong, N., Yoo, J., & Cha, S. (2010). Customization of Scrum Methodology for Outsourced E-Commerce Projects. *Proceedings of the 2010 Asia Pacific Software Engineering Conference*. <http://dx.doi.org/10.1109/apsec.2010.43>
- Hossain, E., Babar, M. A., & Paik, H. (2009). Using Scrum in Global Software Development: A Systematic Literature Review. *Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering*. <http://dx.doi.org/10.1109/icgse.2009.25>
- Hossain, E., Bannerman, P. L., & Jeffery, D. R. (2011b). Scrum Practices in Global Software Development: A Research Framework. *Lecture Notes in Computer Science*, 88–102. http://dx.doi.org/10.1007/978-3-642-21843-9_9
- Hossain, E., Bannerman, P. L., & Jeffery, R. (2011a, May). Towards an understanding of tailoring scrum in global software development: a multi-case study. *Proceedings of the 2011 International Conference on Software and Systems Process*. <http://dx.doi.org/10.1145/1987875.1987894>
- Jalali, S., & Wohlin, C. (2011). Global software engineering and agile practices: a systematic review. *J. Softw. Evol. and Proc.*, 24(6), 643–659. <http://dx.doi.org/10.1002/smr.561>
- Lewin, K. (1946). Action Research and Minority Problems. *Journal of Social Issues*, 2(4), 34–46. <http://dx.doi.org/10.1111/j.1540-4560.1946.tb02295.x>
- Marques, A. B., Rodrigues, R., & Conte, T. (2012). Systematic Literature Reviews in Distributed Software Development: A Tertiary Study. *Proceedings of the 2012 IEEE Seventh International Conference on Global Software Engineering*. <http://dx.doi.org/10.1109/icgse.2012.29>
- Ó Conchúir, E., Holmström Olsson, H., Ågerfalk, P. J., & Fitzgerald, B. (2009). Benefits of global software development: exploring the unexplored. *Software Process: Improvement and Practice*, 14(4), 201–212. <http://dx.doi.org/10.1002/spip.417>
- Paasivaara, M. (2011). Coaching Global Software Development Projects. *Proceedings of the 2011 IEEE Sixth International Conference on Global Software Engineering*. <http://dx.doi.org/10.1109/icgse.2011.33>
- Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2009). Using Scrum in Distributed Agile Development: A Multiple Case Study. *Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering*. <http://dx.doi.org/10.1109/icgse.2009.27>
- Paasivaara, M., Heikkilä, V. T., & Lassenius, C. (2012). Experiences in Scaling the Product Owner Role in Large-Scale Globally Distributed Scrum. *Proceedings of the 2012 IEEE Seventh International Conference on Global Software Engineering*. <http://dx.doi.org/10.1109/icgse.2012.41>
- Patton, M. (2002). *Qualitative research and evaluation methods*. Thousand Oaks, Calif.: Sage Publications.
- Ramesh, B., Cao, L., Mohan, K., & Xu, P. (2006). Can distributed software development be agile? *Commun. ACM*, 49(10), 41. <http://dx.doi.org/10.1145/1164394.1164418>

- Ramesh, B., Mohan, K., & Cao, L. (2012). Ambidexterity in Agile Distributed Development: An Empirical Investigation. *Information Systems Research*, 23(2), 323–339. <http://dx.doi.org/10.1287/isre.1110.0351>
- Schwaber, K. & Beedle, M. (2002). *Agile software development with Scrum*. Upper Saddle River, NJ: Prentice Hall.
- Sison, R., & Yang, T. (2007). Use of Agile Methods and Practices in the Philippines. *Proceedings of the 14th Asia-Pacific Software Engineering Conference*. <http://dx.doi.org/10.1109/aspec.2007.35>
- Šmite, D., Moe, N. B., & Ågerfalk, P. J. (2010). *Fundamentals of Agile Distributed Software Development. Agility Across Time and Space*, 3–7. http://dx.doi.org/10.1007/978-3-642-12442-6_1
- Šmite, D., Wohlin, C., Galviņa, Z., & Prikładnicki, R. (2012). An empirically based terminology and taxonomy for global software engineering. *Empirical Software Engineering*, 19(1), 105–153. <http://dx.doi.org/10.1007/s10664-012-9217-9>
- Srinivasan, J., & Lundqvist, K. (2010). Agile in India. *Proceedings of the 3rd India Software Engineering Conference on India Software Engineering Conference*. <http://dx.doi.org/10.1145/1730874.1730898>
- Stake, R. (1995). *The art of case study research*. Thousand Oaks, Calif.: Sage Publications.
- Strauss, A. & Corbin, J. (1998). *Basics of qualitative research*. Thousand Oaks, Calif.: Sage Publications.
- Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. (2007). *Distributed Scrum: Agile Project Management with Outsourced Development Teams*. Proceedings of the 2007 40th Annual Hawaii International Conference on System Sciences. <http://dx.doi.org/10.1109/hicss.2007.180>
- Vallon, R., Estácio, B., Prikładnicki, R., & Grechenig, T. (forthcoming). *Trends and Directions of Applying Agile Practices in Global Software Development: A Systematic Mapping* (Working paper in revision).
- Verner, J. M., Brereton, O. P., Kitchenham, B. A., Turner, M., & Niazi, M. (2012). *Systematic literature reviews in global software development: a tertiary study*. Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering. <http://dx.doi.org/10.1049/ic.2012.0001>
- Verner, J. M., Sampson, J., Tomic, V., Bakar, N. A. A., & Kitchenham, B. A. (2009). *Guidelines for industrially-based multiple case studies in software engineering*. Proceedings of the 2009 Third International Conference on Research Challenges in Information Science. <http://dx.doi.org/10.1109/rcis.2009.5089295>
- Yin, R. (2003). *Case study research*. Thousand Oaks, Calif.: Sage Publications.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).