THE ROLE OF SHARED UNDERSTANDING IN DISTRIBUTED SCRUM DEVELOPMENT: AN EMPIRICAL ANALYSIS

Markus Hummel
Senacor Technologies AG, markus.hummel@senacor.com

Christoph Rosenkranz
University of Cologne, rosenkranz@wiso.unikoeln.de

Roland Holten
Goethe University, holten@wiwi.unifrankfurt.de

Follow this and additional works at: http://aisel.aisnet.org/ecis2016_rp

Recommended Citation
Hummel, Markus; Rosenkranz, Christoph; and Holten, Roland, "THE ROLE OF SHARED UNDERSTANDING IN DISTRIBUTED SCRUM DEVELOPMENT: AN EMPIRICAL ANALYSIS" (2016). Research Papers. 28. http://aisel.aisnet.org/ecis2016_rp/28
The Role of Shared Understanding in Distributed Scrum Development: An Empirical Analysis

Research Paper

Hummel, Markus, Senacor Technologies AG, Nuremberg, Germany, markus.hummel@senacor.com

Rosenkranz, Christoph, University of Cologne, Cologne, Germany, rosenkranz@wiso.uni-koeln.de

Holten, Roland, Goethe University, Frankfurt am Main, Germany, holten@wiwi.uni-frankfurt.de

Abstract

Originally, agile information systems development (ISD) methods such as Scrum are intended for co-located teams. Recent research efforts provide initial evidence of the applicability of Scrum in distributed environments. What is missing so far, however, are theoretical conceptualizations that explain these findings as well as confirmatory studies that validate those exploratory results. We address this gap by investigating the influence of team distribution on project success, with a focus on shared understanding. Drawing on Shared Mental Models Theory, we investigate the building and maintaining of shared understanding among team members. We conducted a field study in a software product development company and employ a quantitative survey design, which is complemented by seven semi-structured interviews. We find that the level of team distribution does not moderate the highly significant effect of shared understanding on project success. We consult our qualitative data to provide explanations for this surprising finding.

Keywords: Agile Information Systems Development, Team Distribution, Shared Understanding, Project Success.
1 Introduction

Agility in the information systems development (ISD) domain is “the continual readiness of an ISD 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, p. 340). Such agility is considered to be crucial for ISD teams that have to adjust to frequent changes in customer requirements in today’s turbulent business environments (Conboy, 2009). In order to achieve agility in the ISD process, several agile ISD methods have been proposed (Abrahamsson et al., 2002). The most popular agile ISD method today is Scrum (Dingsøyr et al., 2012; VersionOne, 2015). In contrast to traditional, plan-based methods such as the Waterfall model (Royce, 1970), agile ISD methods rely on more iterative and incremental approaches, emphasizing frequent and iterative software development, constant customer involvement, informal and direct face-to-face communication within the team as well as with the customer representative, and self-organizing teams (Beck, 1999; Beck et al., 2001; Schwaber and Beedle, 2002). Although experience reports of practitioners provide evidence for the success of agile ISD methods, research lacks behind in validating this claim (Erickson et al., 2005). This is in parts attributed to the missing “theoretical glue” (Conboy, 2009, p. 330), which has led to calls for a more theory-based approach of empirical studies on agile ISD (Dingsøyr et al., 2012).

Originally, agile ISD methods such as Scrum are intended for co-located ISD teams, with team members sitting in the same office space or close by (Beck, 1999; Schwaber and Beedle, 2002). Similar to teams following traditional ISD methods, distributed agile ISD teams have to face challenges in terms of temporal, geographical, and sociocultural distances (Holmström et al., 2006; Lee et al., 2013). Recent research efforts pinpoint the facets of agility in a distributed context (Sarker et al., 2009; Sarker and Sarker, 2009) and provide initial evidence of the applicability of agile ISD methods in distributed environments (Paasivaara et al., 2008; Paasivaara et al., 2009). What is missing so far, however, are theoretical conceptualizations that explain these findings as well as confirmatory studies that replicate and validate the initial exploratory results. Despite claims that agile ISD works in distributed environments, the literature is surprisingly silent in terms of confirming or refuting these assertions. As a literature review on distributed agile ISD concludes, “[…] there is an immediate need of increasing the quantity and quality of empirical studies to describe, evaluate, explore and explain the use of various Scrum practices in GSD [Global Software Development] projects” (Hossain et al., 2009, p. 183).

We fill this gap by investigating the role of team distribution on success of agile ISD projects, with a special focus on shared understanding and the use of Scrum. The building and maintaining of shared understanding between team members and with the customer representative has repeatedly been identified as one of the critical success factors for ISD projects, both conceptually as well as empirically (e.g., Tan, 1994; Yu and Petter, 2014). As a theoretical lens, we draw on the conceptualization of shared understanding as described in Shared Mental Models Theory (Cannon-Bowers et al., 1993). In sum, we investigate the following research question: “How does team distribution affect the relationship between shared understanding and project success of Scrum projects?”

We conducted a field study in a software product development company and employ a quantitative survey design, which is complemented by seven in-depth, semi-structured interviews. As our main result, we find that the level of team distribution does not moderate the highly significant effect of shared understanding on project success. Since agile ISD allegedly relies heavily on co-located teams, including frequent and direct interactions, we find this result surprising. Consequently, we conducted interviews in order to generate detailed explanations of our findings.

The remainder of this paper is structured as follows. First, we look into related work on the applicability of Scrum in distributed settings and explain Shared Mental Models Theory. Next, we present the hypotheses summarized in our research model. Then, we present our research design, measurement, data collection, and data analysis procedure. Subsequently, we present the results of our study, followed by the discussion of our results, including implications for research and practice.
2 Theoretical Background

2.1 Scrum in Distributed Teams

In this study we focus on the agile ISD method Scrum, which is as of today the most frequently employed method in practice (VersionOne, 2015). Despite having a different focus, the roles and meeting practices of Scrum are also applicable to other agile methods such as Extreme Programming, Crystal, or Lean Software Development (Abrahamsson et al., 2002; Wang et al., 2012). In essence, Scrum is a project management framework that promotes five meeting practices that rely on frequent, direct interactions among team members (Schwaber and Beedle, 2002): “daily scrum”, “sprint planning”, “sprint review”, “sprint retrospective”, and lastly “scrum of scrums” in larger settings.

Initially, methods for agile ISD such as Scrum have been designed for use within small, co-located development teams (Beck, 1999; Schwaber and Beedle, 2002). First conceptual papers on how to apply agile methods in a distributed setting date back to 2001 (Kircher et al., 2001). Experience reports and exploratory empirical studies provide first evidence that agile ISD methods can also be used and applied in distributed environments (Cho, 2007; Lee et al., 2013; Paasivaara et al., 2008; Sutherland et al., 2009; Sutherland et al., 2007). It has been found that Scrum practices need to be adapted to the challenges of the distributed development situation. Failure in addressing these challenges may result in miscommunication, misunderstandings, or confusion among team members (Hossain et al., 2009). For example, in a distributed context, people cannot meet in person so that the use of electronic communication tools is necessary in order to ensure the communication bandwidth for the previously described meetings. Specifically, previous studies report that tools which allow direct synchronous communication among team members are needed, for example, videoconferencing or telephone conferencing with web-cameras and application sharing (Cho, 2007; Paasivaara et al., 2008; Sutherland et al., 2009). For example, demonstrations of working functionality can be held by videoconferencing with desktop sharing (Cho, 2007; Paasivaara and Lassenius, 2006). Moreover, asynchronous communication media such as e-mail and instant messaging have also been found to be useful (Cho, 2007; Paasivaara et al., 2008).

Besides the use of communication tools, other adaptations have been described in the literature. For example, in order to shorten the time of the daily scrum meetings and overcome language barriers, the three questions normally asked during a standup meeting should be answered prior to the meeting by e-mail (Sutherland et al., 2007). Similarly, as many issues as possible should be solved before meetings in order to keep them short (Simons, 2002). Sprint planning or retrospective meetings may usually take four hours or longer, which is a challenge when including team members that experience significant different time zones (Hossain et al., 2009). Due to the differences in time zones, only key team members may participate in the meetings (Layman et al., 2006). Effective retrospectives may be challenging due to the socio-cultural distances between team members because offshore team members may not voice their true opinions (Paasivaara et al., 2008). Other proposed prescriptions for dealing with distributed agile ISD situations include the synchronization of work hours, conducting asynchronous retrospectives, or reducing the number of Scrum meetings (Hossain et al., 2009).

Despite these empirical findings, studies trying to conceptualize, describe, and explain the adaption of practices using theoretical conceptualizations and models on a larger scale are scarce.

2.2 Shared Mental Models Theory

The importance of shared understanding on ISD has been widely acknowledged (e.g., Bittner and Leimeister, 2014; Corvera Charaf et al., 2013; Tan, 1994), but confirmatory research that provides empirical evidence of its role on the distributed Scrum process is lacking. We posit that Shared Mental Models Theory (Cannon-Bowers et al., 1993) is an appropriate theory to frame our investigation of distributed Scrum teams because, first, the theory investigates the role of shared understanding on
team performance and, second, it has been previously used to conceptualize distributed agile ISD in general (Yu and Petter, 2014). Shared understanding is “the degree of cognitive overlap and commonality in beliefs, expectations, and perceptions about a given target” (Cohen and Gibson, 2003, p. 8). It is a state that arises from effective communication processes, which result in the achievement of goals (Churchman and Schainblatt, 1965; Tan, 1994). In contrast to related theories of social cognition such as Transactive Memory Systems Theory (Austin, 2003), Shared Mental Models Theory encourages the development of shared mental models on the basis of common, shared knowledge including low levels of specialization, which is in accordance with principles on agile ISD that encourage cross-functional teams (Schwaber and Beedle, 2002).

The theory posits that teams need to maintain a shared mental model, in other words a shared understanding, in order to work successfully. Such shared mental models are developed in four consecutive stages – knowing, learning, understanding, and executing (Yu and Petter, 2014). In our research, we focus on the ‘understanding’ stage because the goal of this stage is to develop shared understanding about the tasks and the team interaction processes. This is closely related to the understanding of Scrum because the actual development and establishment of shared understanding is supposed to be achieved by team communication and collaboration (Van den Bossche et al., 2011; Warner et al., 2005).

2.3 Shared Understanding in Distributed Scrum Teams

Establishing shared understanding is especially important in a distributed context, when team members are not able to clarify issues face-to-face and language issues hinder the building of a common ground (Damian and Zowghi, 2003). Previous literature suggests several prescriptions for developing shared understanding and shared mental models in a distributed setting.

First, shared understanding is induced by Scrum meeting practices. The sprint planning and review meetings contribute towards the establishment of shared understanding by promoting that the development team involves the customer at least at the beginning and at the end of an iteration (Schwaber and Beedle, 2002). This gives the customer the opportunity to try out the software product and give specific feedback to the development team, which in turn enables that the feedback of the customer is implemented within the next iteration (Schwaber and Beedle, 2002). Those meetings are also a good opportunity to ask business-related questions so that the understanding of requirements is enhanced, especially for the remote team members (Paasivaara et al., 2008). When development team and product owner sit far away from each other, misunderstandings are common and may remain unnoticed until the next sprint review meeting (Paasivaara et al., 2008). Furthermore, daily scrum meetings allow getting a good overview of what is happening in the project, and those meetings help fostering off-shore understanding of customer needs (Sutherland et al., 2009). Questions are solved in those daily meetings by e-mail or instant messaging (Sutherland et al., 2007). Lastly, the Scrum-of-Scrum meetings help to get an overview of what other teams within the same project are doing (Paasivaara et al., 2008).

The development of shared mental models is also encouraged by using communication tools. For example, a wiki can mitigate the issues because it allows to store shared knowledge (Cho, 2007; Sutherland et al., 2009). Furthermore, a global build repository as well as reporting and tracking tools are useful for establishing shared understanding in a distributed Scrum context (Sutherland et al., 2007). Moreover, short asynchronous communication loops and continuous access to product and process information is important (Layman et al., 2006). Less rich communication media such as groupware with “shared desktop” features, coupled with audio support, help to understand other team members (Sarker and Sarker, 2009). Likewise, a continuous awareness and visibility of distributed team members enables shared understanding (Sarker and Sarker, 2009). For example, this may be achieved with always-on Skype connections or other videoconferencing tools that help to share context and priorities (Sutherland et al., 2009).
Another prerequisite for the development of shared understanding is that cultures mesh together (Sarker and Sarker, 2009). Project domain knowledge across different cultural backgrounds is increased by gradually moving from a co-located project to a distributed project. The initial sprints should be developed co-located, before splitting up in different locations (Sureshchandra and Shrinivasavadhan, 2008). Moreover, it is helpful having a key team member physically situated with the offshore team in order to develop shared understanding (Layman et al., 2006; Poole, 2004; Yap, 2005). Frequent visits to other teams also help to develop shared understanding and the necessary level of knowledge (Paasivaara et al., 2008; Sarker and Sarker, 2009; Sutherland et al., 2009). If the customer cannot be co-located with the team, a proxy or remote customer is helpful to transfer business domain knowledge (Nisar and Hameed, 2004). It is essential for the creation of shared understanding to clearly define the customer role who is responsible for the requirements (Layman et al., 2006).

3  Hypothesis Development

In terms of the development of shared understanding in the distributed Scrum process, the ‘understanding’ stage is mostly taking place in meeting practices. In accordance with Shared Mental Models Theory, some agile practices are useful in developing a shared understanding about the tasks to be completed, while other agile practices create shared mental models about team processes and team interactions (Yu and Petter, 2014).

The impact of shared mental models on project success within ISD projects has already been theoretically conceptualized (e.g., Yu and Petter, 2014), but so far empirical evidence is scarce. Existing confirmatory studies on shared mental models are mostly based on team performance literature (e.g., Ko et al., 2005; Schmidt et al., 2014) or literature dealing with traditional, plan-based ISD (e.g., Hsu et al., 2011). Results on traditional ISD methods show that shared mental models including shared understanding and shared knowledge are important for team performance because they improve the collaboration between developers as well as with the customer, and there are positive effects on the quality of the delivered software product (Holmström et al., 2006; Hsu et al., 2011; Ko et al., 2005; Nelson and Cooprider, 1996; Tan, 1994). Shared understanding also influences the way team members coordinate and perform (Espinosa et al., 2001; Hinds and Weisband, 2003; Zakaria et al., 2004) and it ensures that requirements of the customer are transferred successfully into the software or IT system (Aranda et al., 2010; Corvera Charaf et al., 2013; Sharp and Robinson, 2004). A common understanding about how teams at different sites work, think, communicate, and deal with problems is also important because it enables a seamless transition of work sequentially across time zones in order to achieve temporal bridge-based agility (Paasivaara et al., 2008; Sarker and Sarker, 2009).

We transfer those related, mostly explorative findings to the distributed Scrum domain and presume that if there is no common ground between the participants of the agile ISD process, there is a high probability that the ISD project will fail. Consequently, we expect that shared understanding between both the team members and with the customer enables the project team to deliver a successful software product.

Hypothesis 1: Shared understanding is positively correlated with project success.

Related work highlights the problem of creating shared understanding in a distributed ISD context (e.g., Simons, 2002; Yap, 2005; Zakaria et al., 2004). “[…] information and knowledge sharing issues were the most important issues in the company due to its geographically distributed working environment” (Cho, 2007, p. 4). A critical challenge is the varying cultural background that may lead to misunderstandings and confusion as results of language and interpretation problems. “This has implications for communication, coordination, and control and makes it a real challenge to create mutual understanding within and between teams” (Holmström et al., 2006, p. 12). Other findings also emphasize that the language barrier is a significant problem for achieving shared understanding (Damian and Zowghi, 2003). Similarly, literature also proposes several solutions to the challenge of creating shared
understanding in a distributed context, for example, appropriate communication tools (Paasivaara et al., 2008), frequent visits (Sarker and Sarker, 2009), and clearly defined roles (Layman et al., 2006).

In terms of the impact of team distribution on project success, previous findings suggest that the productivity of co-located teams is doubled in comparison to non co-located teams (Teasley et al., 2000). Also, distributed virtual teams are observed having a lower team performance than co-located teams (Gallupe et al., 1994; Herbsleb and Mockus, 2003; Potter and Balthazard, 2002).

In sum, shared understanding has been identified as critical challenge of distributed ISD. Despite several possibilities to improve the development of shared understanding in those contexts, we expect that the challenges outweigh the solutions so that higher degrees of team distribution are negatively correlated with the effect of shared understanding on project success. The rationale is that even though appropriate tools etc. help to build up shared understanding, it is not possible to create a similar working environment as when sitting co-located.

Hypothesis 2: Higher levels of team distribution influence the effect of shared understanding on project success in a negative way.

Figure 1 summarizes the previous discussion and presents our research model.

Figure 1. Research Model

4 Method

4.1 Research Design

We conducted a field study in a product and service development department of a multinational telecommunications company. The department also develops software and software-based products. For reasons of confidentiality, we call the company Alpha. Alpha employs more than 3000 employees worldwide. Alpha develops software products, communication services, media, entertainment, cloud services, advertisement and payment services for business and private customers. Specifically, our study is situated in the department of Alpha that is responsible for creating new products, which is considered to be a suitable setting for agile ISD methods such as Scrum because Scrum is supposed to enable high levels of speed-to-market. Moreover, the department managers assume the delivery of higher quality products by having adopted Scrum.

We employed a multi-method design including both a quantitative as well as a qualitative research component (Kaplan and Duchon, 1988; Lee, 1991; Venkatesh et al., 2013). Complementing quantitative with qualitative research allows combining the strengths of both approaches. On the one hand, quantitative research seeks to ensure statistical conclusion validity and generalizability of the results. On the other hand, qualitative research allows taking the context into account and providing in-depth insights into processes and mechanisms of the case site.

The unit of analysis of our study is the individual level. The data collection is carried out on the individual level, but the individual data is also used to derive conclusions about the team level, which is an accepted practice as demonstrated by previous studies (e.g., Guinan et al., 1998; Lee and Xia, 2010). In terms of the data collection, we first distributed a quantitative online questionnaire within the organization. 387 persons of the product development department were invited by e-mail to participate in
the survey. The quantitative data was complemented by seven semi-structured interviews that lasted between 30 and 45 minutes. When choosing interview partners, we ensured that the three project roles Scrum master, product owner, and developer are well represented in the sample. This allows us to investigate the ISD process from different angles. After conducting seven interviews, we felt that we captured the most important insights and theoretical saturation was achieved. Unfortunately, we were not allowed to record the interviews due to security and privacy concerns, but we captured the essential insights by taking handwritten notes in a structured template. After the interviews, we edited, processed, and complemented the notes with additional information.

4.2 Measurement & Data Analysis Procedure

For the questionnaire, the three constructs shared understanding, team distribution, and project success were measured using seven point Likert scales ranging from “Strongly Disagree” to “Strongly Agree”. Survey participants were encouraged to refer to the last completed development project that included distributed team members. Shared understanding was measured by adapting four reflective items (Biocca et al., 2001) and the scale for measuring team distribution is based on the reflective scale “co-located office space” (So and Scholl, 2009), which was reversed in order to capture the dispersion of the team members. Several attempts have been made to unearth and measure the dimensions of success in ISD projects, of which one of the most comprehensive ones is the Unified Model of ISD success (Siau et al., 2010), but the definition of success should focus on the research question at hand (DeLone and McLean, 1992). For our study, we selected facets of success that are crucial in the agile ISD domain, namely customer satisfaction and on-time delivery of software (Beck et al., 2001). Consequently, we define project success as (1) satisfying customer needs by delivering the appropriate software functionality and (2) completing the project in-time (Batra et al., 2010; Siau et al., 2010). We measured project success with three reflective items based on previous literature (Lee and Xia, 2010; Wallace et al., 2004). Table 1 gives a complete list of the questionnaire items.

<table>
<thead>
<tr>
<th>Construct</th>
<th>Item</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Understanding</td>
<td>SU1: Your opinion was transparent for your team members.</td>
<td>Adapted from Biocca et al. (2001)</td>
</tr>
<tr>
<td></td>
<td>SU2: The opinion of other team members was transparent for you.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SU3: Your team members understood what you meant.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SU4: You understood what other team members meant.</td>
<td></td>
</tr>
<tr>
<td>Team Distribution</td>
<td>TD1: All development team members were located in …</td>
<td>Adapted and reversed from So and Scholl (2009)</td>
</tr>
<tr>
<td>Scale label: Same room…Different time zones</td>
<td>TD2: Compared to the development team, the product owner was located in …</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TD3: Compared to the development team, the Scrum master was located in …</td>
<td></td>
</tr>
<tr>
<td>Project Success</td>
<td>PS1: The product was launched in time.</td>
<td>Adapted from Lee and Xia (2010), and Wallace et al. (2004)</td>
</tr>
<tr>
<td></td>
<td>PS2: The functionality of the software product was in accordance with the customer needs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PS3: The software product fulfilled the qualitative requirements of the customer.</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Measurement Scales

For data analysis of the questionnaire, co-varianced based structural equation modeling (CB-SEM) was employed, using the software package IBM SPSS Amos 16. We evaluated established guidelines in order to evaluate whether to use CB-SEM or partial-least squares SEM (Gefen et al., 2011; Hair et al., 2011). CB-SEM is the appropriate approach for our analysis because only reflective constructs are included in our model (no formative constructs), and we do not use any hierarchical constructs. Furthermore, CB-SEM should be used to confirm and test well-established models and theories. Since we
transfer established concepts from literature on distributed, plan-based ISD and team performance to the agile ISD domain, a confirmatory analysis in CB-SEM is appropriate.

For the interviews, we prepared an interview guideline that was based on the survey results and previous literature on distributed agile ISD (e.g., Sarker and Sarker, 2009), shared understanding (e.g., Cannon-Bowers et al., 1993), and project success (e.g., Siau et al., 2010). We coded the structured handwritten notes using the software MAXQDA. The coding scheme was initially based on our research model. During analysis, we extended the coding scheme inductively.

5 Results

5.1 Survey Results

We were able to retrieve 87 fully completed questionnaires (response rate 22%). We had to discard six questionnaires due to nonsensical answers, resulting in 81 questionnaires that were used for the analysis. The average experience in developing following agile ISD methods is 2.57 years (s.d. 1.638) and the average team size is 9.6 (s.d. 5.581). More than 95% of the participants use Scrum or a combination of Scrum with related ISD methods. All three project roles, as prescribed by Scrum, are well represented in our sample. The participants were asked to rate the agility of their last completed distributed project on a 7-point Likert scale. The results show that our participants are mainly involved in highly agile projects (mean 5.6, s.d. 1.1, range 3-7). We also assessed common method bias with Harman’s One Factor Test (Podsakoff et al., 2003; Podsakoff and Organ, 1986) and with the Latent Variable Approach (Lindell and Whitney, 2001; Malhotra et al., 2006). In addition, non-response bias was evaluated (Sivo et al. 2006). Both potential threats to validity are unlikely to be an issue in our sample.

In order to evaluate the validity of our measurement model, we conducted a confirmatory factor analysis using IBM SPSS Amos 16 and IBM SPSS 19. Table 2 shows the results of the measurement model evaluation in terms of reliability and convergent validity of the scales. Established guidelines (Gefen et al., 2011; Hair et al., 2011; Straub et al., 2004) suggest that Cronbach’s Alpha and Composite Reliability should be larger than 0.7, whereas a value of at least 0.5 is needed for Average Variance Extracted (AVE). All three thresholds are exceeded for our constructs in the analysis, indicating good reliability and convergent validity. We also evaluated the latent variable correlations and the evaluation of the Fornell-Larcker Criterion in order to ensure discriminant validity. The square root of the AVE of all three constructs is higher than the correlations to the constructs, indicating good discriminant validity.

<table>
<thead>
<tr>
<th>Construct</th>
<th>Cronbach’s Alpha</th>
<th>Composite Reliability</th>
<th>AVE</th>
<th>Mean</th>
<th>S.D.</th>
<th>1.</th>
<th>2.</th>
<th>3.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Shared understanding</td>
<td>0.85</td>
<td>0.90</td>
<td>0.60</td>
<td>5.886</td>
<td>0.96</td>
<td>0.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Team Distribution</td>
<td>0.90</td>
<td>0.88</td>
<td>0.75</td>
<td>3.469</td>
<td>1.68</td>
<td>-0.06</td>
<td>0.86</td>
<td></td>
</tr>
<tr>
<td>3. Project Success</td>
<td>0.87</td>
<td>0.92</td>
<td>0.71</td>
<td>5.844</td>
<td>1.25</td>
<td>0.40</td>
<td>0.15</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Note: AVE = Average Variance Extracted, S.D. = Standard Deviation. Diagonal elements of the last three columns represent the square root of the AVE. Off diagonal elements are the correlations among latent constructs.

**Table 2. Evaluation of Reliability and Convergent Validity**

The loadings of the items on the respective construct should be larger than 0.7 (Hair et al., 2011) and substantially higher in comparison to the loadings of the items on other constructs (Straub et al., 2004). Both requirements are fulfilled in our measurement model (see Table 3), except for one item (SU4), which we decided to retain in the analysis because it is very close to the 0.7 threshold and it theoretically contributes to the construct.
Table 3. Loadings and Crossloadings of Indicators

<table>
<thead>
<tr>
<th>Construct</th>
<th>Item</th>
<th>SU</th>
<th>TD</th>
<th>PS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared understanding (SU)</td>
<td>SU1</td>
<td>0.781</td>
<td>-0.045</td>
<td>0.316</td>
</tr>
<tr>
<td></td>
<td>SU2</td>
<td>0.800</td>
<td>-0.046</td>
<td>0.322</td>
</tr>
<tr>
<td></td>
<td>SU3</td>
<td>0.829</td>
<td>-0.048</td>
<td>0.334</td>
</tr>
<tr>
<td></td>
<td>SU4</td>
<td>0.667</td>
<td>-0.039</td>
<td>0.269</td>
</tr>
<tr>
<td>Team Distribution (TD)</td>
<td>TD1</td>
<td>-0.046</td>
<td>0.795</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>TD2</td>
<td>-0.056</td>
<td>0.964</td>
<td>0.014</td>
</tr>
<tr>
<td></td>
<td>TD3</td>
<td>-0.048</td>
<td>0.833</td>
<td>0.012</td>
</tr>
<tr>
<td>Project Success (PS)</td>
<td>PS1</td>
<td>0.287</td>
<td>0.010</td>
<td>0.713</td>
</tr>
<tr>
<td></td>
<td>PS2</td>
<td>0.337</td>
<td>0.012</td>
<td>0.837</td>
</tr>
<tr>
<td></td>
<td>PS3</td>
<td>0.385</td>
<td>0.014</td>
<td>0.956</td>
</tr>
</tbody>
</table>

In order to evaluate the structural model, we conducted a maximum likelihood estimation with the standardized composite constructs. Figure 2 shows the results. As expected, the data analysis revealed a highly significant effect of shared understanding on project success, confirming H1. Surprisingly, we did not find a significant moderation effect of team distribution on the effect of shared understanding on project success, rejecting H2. Our model explains 20.7% of the variance ($R^2$) in project success.

Figure 2. Structural Model

We created an interaction plot (Figure 3) in order to illustrate the moderation effect. As theoretically hypothesized, we find that team distribution moderately dampens the positive relationship between shared understanding and project success ($\beta = -0.110$), but this effect is not statistically significant ($p$-value = 0.268).

Figure 3. Interaction Plot

We calculated established fit indices in order to evaluate model fit (Gefen et al., 2011; Hair et al., 2010). The values of the fit indices are presented in Table 4. The results show that our structural model has very good model fit because all fit indices exceed the recommended thresholds.
Lastly, we conducted a post-hoc power analysis in G*Power 3. The results show that the power is greater than 0.97 so that our data is able to detect small sample sizes (Erdfelder et al., 1996).

### 5.2 Interview Results

The semi-structured interviews allow us to provide plausible explanations of the survey results by taking the context into account. Our sample included seven informants, including Scrum Masters, Product Owners, and developers. In the following analysis, we refer to our interviewees as I1 to I7. All interviews were conducted in Germany and in face-to-face discussions. Since we were not allowed to record the interviews, we use verbatim quotes to justify our conclusions from the qualitative data. First, we describe the perceived challenges of team distribution in the specific context of our case company, followed by strategies for overcoming these challenges.

#### 5.2.1 Perceived Challenges of Team Distribution

One perceived challenge in a distributed Scrum setting is ensuring constant and frequent direct communication, which is vital for any successful agile ISD setting. For example, I1 mentioned that coordination challenges occurred that were caused by lacking communication between separate sites. Our informants report a higher barrier for communication when team members are distributed because one has to pick up the phone or send invitations for a meeting. Moreover, due to lags in the video transmission, facial expressions may not fit to the spoken words and one reacts differently as when it is synchronized, so communication is not as fluent as when sitting co-located.

Language barriers were stated to be another main challenge for project success in a distributed Scrum setting. Those barriers entail fewer and more distorted communication.

“There is simply not as much communication.” (I2)

This is caused by different degrees of understanding and the fact that team members feel constrained, especially team members who are not confident and experienced with their language skills.

“Due to varying language skills, misunderstandings occurred and persons had difficulties in expressing themselves.” (I5)

Our informants also state that there were potential misunderstandings, which may remain unnoticed for longer periods of time. Our interviewees observed or experienced themselves that there is an inhibition threshold, or “shame”, when talking to the group in a different language because of difficulties to express themselves. This is mainly observed at the beginning of the collaboration. The interviewees also report that when team members realized that other people have similar problems, they felt better about their own weakness and communication increased. Language is also considered to be difficult because learning to understand irony, accentuation and word choice is hard when team members are originally speaking different languages.
The third major challenge is cultural difference. I3 realized that it is difficult to adapt to behaviours that are different from the own culture and society. For example, I5 thinks that other cultures deal differently with criticism because some cultures do not mention it directly and others do. I6 also observed cultural differences in terms of reliability, openness and arrogance, which are subjective perceptions as remarked by I6. There are also differences in the communication behaviour, for example, some persons like to talk in a more direct manner or are politer than others.

Our data also reveals implications of team distribution on the well-being of the team. When doing a daily scrum by telephone conference, it is not perceived in the same way than when doing it co-located. Other team members are reported to “disappear” when they are not physically present. Our informants mention that there is a higher probability for conflicts because problems may be easily attributed to other team members that sit far away.

“It is easier to blame remote team members for problems. It is also easier to solve problems when sitting co-located.” (I3)

Lastly, I5 feels that the setup of communication tools is difficult at the beginning of projects and takes time. For example, tools have to be adjusted and translated to one common language.

5.2.2 Used Strategies for Overcoming Challenges of Team Distribution

In the specific context of our case company, Scrum helps to solve the key challenge of ensuring constant communication among all participating team members. The regular meeting practices encourage frequent and direct communication, which lead to shared understanding. Used media for connecting distributed team members for those meetings and during day-to-day work at Alpha were e-mail, phone, videoconferencing (Skype), instant messaging, screen sharing tools and a permanent site-to-site network connection. The site-to-site network connection was set up so that the team members were able to communicate constantly, which enabled communication that is judged to be comparable to co-located team members.

“All necessary communication tools were available, as if team members were sitting co-located.” (I1)

As the remote teams were well equipped with technical communication media, it was possible to start a videoconference (e.g., Skype) at any time. Our interviewees experienced that the communication with the business department is the same compared to co-located ISD, only the transmission channel of messages changed. In general, our interviewees argue that videoconferencing should be preferred because one sees the facial expressions and gesture of other participants. When team members see each other, it feels more like sitting in the same room.

“It helps to see the facial expressions of the other person, even if you know each other.” (I3)

The second key challenge which may inhibit project success is language. In order to resolve language challenges, only persons who speak the same language as the onshore team members were recruited for the outsourced team. Appropriate language skills are a prerequisite for recruiting.

“Only persons who spoke the same language were recruited.” (I3)

Our interviewees feel like team members are generally willing to learn a new language, especially in technical domains. Another solution for language problems is stated to be a shift towards e-mail in order to avoid wrong pronunciation and misunderstandings, mainly in the beginning of a project. Facial expression and gesture perceived in videoconferences also help to solve language issues.

In order to overcome challenges attributed to cultural differences, a Scrum master is reported to be needed to resolve conflicts and misunderstandings. A Scrum master ensures the efficiency of the team; this holds for distributed teams as well. Coaches may also help to communicate in a direct way with each other in order to develop shared understanding. Developers have joint tasks and have to communicate, which helps to establish shared understanding. Team members have to be encouraged to communicate in a direct manner in case they hesitate.

“You need more discipline and more stringent communication.” (I4)
The well-being of the team is ensured by including phases in which distributed team members are able to get to know each other in order to improve shared understanding, relationships, and learning. I2 thinks that it is important that people begin to trust each other. At the beginning of the project, all new team members should work co-located for the first few sprints in order to develop shared understanding and enable project success.

“It is a fundamental difference whether people have seen each other before or not.” (I6)

If persons do not know each other and the first communication is by e-mail, there is a high probability of misunderstandings, which may lead to conflicts. The communication and shared understanding is seriously better after sending the persons to the other country for a few weeks.

“After team members were sent to the other team for a couple of weeks, communication improved significantly. A phone call felt more natural then.” (I6)

Not just at the beginning, but also throughout the project, personal contact is deemed to be important from time to time. It is not possible to get to know each other when only communicating by phone. Our data indicates that as a result of those visits, people knew each other better and there was a lower language barrier, which ultimately results in higher success.

“When you met in person before, you know better how the other person means certain sentences.” (I3)

It is perceived as important to work as a team. For example, team spirit was reportedly improved by regular team events such as having dinner at restaurants, which was triggered by the Scrum master. New team members should be integrated quickly, for example by a joint lunch. The informants report that there is a higher inhibition threshold for blaming other people when people know each other and have good relationships.

6 Discussion

Drawing on Shared Mental Models Theory, we conducted a field study in a software product development company in order to investigate the role of shared understanding in the distributed Scrum development process. On the basis of 81 completely returned questionnaires, we confirmed our hypothesis that posits a highly significant effect of shared understanding on project success. In addition, we hypothesized that team distribution negatively correlates the effect of shared understanding on project success, for which we surprisingly did not find statistical support in our results. We complemented our quantitative survey data with seven semi-structured interviews in order to provide context-specific explanations of the survey results.

We find that the negative effect of team distribution on mutual understanding and project success was mitigated by the following three strategies. First, the remote teams were well-equipped in terms of communication tools in order to ensure constant communication with the offshore team members. Second, distributed teams worked together co-located for the first few sprints in order to get to know each other. Third, when recruiting new team members in an offshore context, only persons that speak the same language as the onshore team were recruited.

Our study has implications for theory and practice. We provide quantitative results of the effect of shared understanding on project success in the agile ISD domain which was so far lacking. Furthermore, we integrated Shared Mental Models Theory in the growing knowledge base of agile ISD, thereby contributing to the missing theoretical glue of agile ISD. In accordance with previous results (e.g., Paasivaara et al., 2008), we find that Scrum is also applicable in a distributed context. Our results show that co-location is not a necessary prerequisite in order to generate shared understanding in an distributed agile ISD context, which extends previous research that argued co-location is highly important for the creation of shared mental models (Yu and Petter, 2014). We also found empirical evidence that frequent visits and different communication modes mitigate the problem of creating shared understanding, which extends and validates previous qualitative findings (e.g., Layman et al., 2006; Paasivaara et al., 2008). Our results are also partially in line with previous findings that do not find
evidence for more misunderstandings in terms of priority and scheduling of tasks for people of different sites (Herbsleb and Mockus, 2003). Furthermore, our results show the applicability of previous findings to the distributed Scrum domain because studies found significant effects of related concepts such as shared knowledge on team performance (e.g., Nelson and Cooprider, 1996).

We encourage future research that conducts longitudinal studies of distributed Scrum teams. Our data is only a snapshot of the development process and we expect valuable insights when investigating the learning effect of distributed teams when working co-located for the first few sprints. Also, investigations of other moderators of shared understanding and project success, for example experience of team members, are a valid opportunity for future research. Furthermore, more research is needed that identifies other influential factors on project success. For example, the role of trust (Majchrzak et al., 2005) or group cohesion (Willer, 2009) for success of agile ISD in general and more specifically in a distributed context is not clear. Also, fundamental Scrum concepts such as self-organization and continuous learning (Schwaber and Beedle, 2002) should be taken into account. Lastly, future research is encouraged that provides deeper insight into the antecedents of mutual understanding in the agile ISD process. Our qualitative interview data as well as related literature (e.g., Te'eni, 2001) indicates that communication may be one of those antecedents, but empirical evidence is still missing.

For practice, we highlight the importance of establishing shared understanding within distributed Scrum projects. Practitioners should take special care in providing the necessary environment for the development of shared mental models of team members. We offer three prescriptions that help to mitigate the challenges of distributed Scrum teams: use of appropriate communication tools, temporary co-location of all team members, and recruitment on the basis of language skills. When following those recommendations, the negative effects of team distribution may be mitigated as in Alpha.

Our study has several limitations. First, we collected data on the individual level. We encourage future research to build on our study by collecting data on the team level, including objective data for measuring success of the Scrum teams. Unfortunately, this was not feasible in our case due to access restrictions but since our study is one of the first confirmatory studies on the inner workings of distributed agile ISD teams, we were still able to provide valuable insights. Second, as our study design was observational in nature and not longitudinal, we were only able to report on a snapshot of the development process and we recorded the information that is present in our population, but we were not able to manipulate variables. Thus, we cannot determine cause-and-effect relationships between our variables; we are only able to make inferences about possible relationships. Further longitudinal, panel, or experimental studies are necessary for strengthening the causal claim and confirming our results. In terms of the qualitative interview-based data, we may have missed some insights because we were not allowed to record the interviews, but we took special care in capturing the essential information by structured handwritten notes. Lastly, we focused on Scrum in our study so we cannot generalize our results to other agile ISD methods in general. Future research is needed that investigates the applicability of our findings to other methods.

7 Conclusion

To sum up, we confirmed shared understanding as important success factor of distributed project success when developing according to Scrum. Our insights are based on quantitative and qualitative data obtained in a field study of a software product development company. Our results show that when following certain strategies, team distribution may not necessarily be an inhibiting factor for shared understanding and project success. In sum, we provide novel insights in the agile ISD development process and extend the theoretical underpinnings of agile ISD.

Acknowledgements

The German Research Foundation (DFG) funded parts of this study under record no. RO 3650/8-1.
References


