



Foreignerds

Agile Software Development

Contents

1. Introduction.....	1
1.1 Scrum and GSD.....	1
1.2 Problematization and Purpose.....	3
1.3 Case Study.....	4
1.3.1 Case Study, Company X.....	4
1.3.2 Case study, Company Y.....	4
2. Theory.....	5
2.1 Scrum.....	5
2.1.1 History & Definition.....	5
2.2 Scrum Theory.....	6
2.2.1 Transparency.....	6
2.2.2 Inspection.....	7
2.2.3 Adaptation.....	7
2.3 Scrum Team.....	7
2.3.1 Product owner.....	7
2.3.2 Development team.....	8
2.3.3 Scrum Master.....	8
2.4 Scrum events.....	8
2.4.1 The Sprint.....	9
2.4.2 Sprint Planning.....	9
2.4.3 Daily Scrum.....	10
2.4.4 Sprint Review.....	10
2.4.5 Sprint Retrospective.....	11
2.5 Scrum artifacts.....	11
2.5.1 Product Backlog.....	11
2.5.2 Sprint Backlog.....	11
2.5.3 Definition of Done.....	12
2.6 Challenges in GSD.....	12
2.7 Hossain et al.'s (2011) Framework.....	13
3. Method.....	17
3.1 Research Strategy.....	17
3.2 Research design.....	17
3.2.1 Participant observations / Ethnography.....	18
3.2.2. Pre-study.....	18
3.2.3 Semi-structured interviews.....	19
3.2.4 Selection process and sampling.....	21
3.3 Data collection.....	21
3.3.1 Observations.....	21
3.3.2 Interviews, Interview guide and Operationalisation.....	23
3.4 Analysis of data.....	24
4 Empirical Findings and Analysis.....	25
4.1 Adjustments on research framework.....	25
4.2 Ethnographic context.....	26
4.3 Communication.....	27
4.3.1 Reduced opportunities for synchronous communication (CA1).....	27

4.3.2 Analysis of Reduced opportunities for synchronous communication (CA1)	29
4.3.3 Face-to-face meetings difficult (CA2)	29
4.3.4 Analysis of Face-to-face meetings difficult (CA2).....	31
4.3.5 Cultural misunderstandings (CA3)	32
4.3.6 Analysis of Cultural misunderstandings (CA3).....	33
4.4 Coordination.....	34
4.4.1 Reduced informal contact can lead to lack of critical task awareness (CB2)	34
4.4.2 Analysis of Reduced informal contact can lead to lack of critical task awareness (CB2).....	36
4.5 Control.....	37
4.5.1 Difficult to convey vision and strategy (CC2).....	37
4.4.2 Analysis of Difficult to convey vision and strategy (CC2).....	38
4.5.3 Perceived threat from training low cost “rivals (CC3)	39
4.5.4 Analysis of Perceived threat from training low cost “rivals” (CC3)	40
5. Theoretical & Practical Implications	41
5.1 Implications for the Hossain et al (2011) Framework.....	41
5.1.1 Communication	41
5.1.2 Coordination	42
5.1.3 Control	42
5.2 Practical Implications	44
5.3 Implications for GSD Theory	45
5.4 Limitations	46
5.5 Future research	47

1. Introduction

With the new millennia came a diverse development in services and products. The main reason for this was the further development of the Internet and related technology. Traditionally, these types of services and products have been produced in developed countries. However, during the last couple of decades production has moved to developing countries. The primary reason for offshoring and outsourcing IT- development, also known as *Global Software Development (GSD)*, can be ascribed to the low cost of labor and to capture competencies which are not available in the domestic country (Sutherland, Schoonheim & Rijk, 2009; Hossain, Bannerman & Jeffery, 2011). A popular development method in GSD has been the use of *Agile* software development. Earlier, most software was developed through a *Waterfall* methodology, which is more rigid than Agile methodology. The Waterfall model of development requires you to follow the requirement elicitation from start to finish without any modification through the whole project. In the iterative Agile methodology, the requirements can change over the course of the project, meaning the methodology is responsive to both initial and additional requirements during the ongoing software development (Schwaber, 1997).

The use of Agile methods is based on a belief that such methods increase productivity and avoid issues such as schedule delays and lack of motivation from the team members (Cardozo, Neto, Barza, França & da Silva, 2010). A common Agile methodology or framework is *Scrum*. It is an Agile development method that has come to be a framework for developing complex products (McKenna, 2016). Scrum is a project management method, which is generally assumed to increase productivity through adaptability and flexibility (Cardozo et al. 2010). In Scrum, the team aims to achieve a smaller number of goals in shorter periods of time that is repeated in iterations, i.e., sprints (McKenna, 2016). These goals are small parts, or increments of the final product.

1.1 Scrum and GSD

The current tendency among companies is to practice GSD, which assumes globally distributed teams, whereas Scrum assumes collocated teams (Hossain, et al. 2011). Even though this contradiction exists, the Scrum method is becoming more frequently used in GSD projects by practitioners and scholars than other agile methods.

During the last couple of years there has been a rise in studies regarding Scrum in GSD. Sutherland, Viktorov, Blount & Puntikov (2007) mention that prior to their study there was a large amount of research conducted on project management (including Scrum practices), distributed development, and outsourcing practices as isolated domains. But studies of combinations, such as

Scrum practices that are both distributed and outsourced, were more scarce. The majority of studies claim that there are many successful aspects of using agile methods when working with distributed teams (Paasivaara, Durasiewicz & Lassenius, 2008; Sutherland et al. 2007; Sutherland et al. 2009; Cho, 2007). However, there are also researchers that has opposed this view and claim that it is still debatable whether agile methods can be used in distributed settings, especially when working in larger agile teams (Williams & Stout, 2008). Industry experience reports suggest that Scrum practice in itself can mitigate commonly recognized issues in GSD (Holmström, Fitzgerald, Ågerfalk & Chonchuir, 2006), for example, by improving communication, trust and motivation (Hossain, Babar & Verner, 2009).

Because of the growing number of studies being conducted, Hossain, Babar & Paik (2009) performed a systematic literature review where they compiled 20 primary papers to evaluate the risk factors and how Scrum practices might mitigate those risks in GSD. The literature has thus documented that there is a need for empirical support for how the Scrum framework can mitigate challenges that occur in GSD, since the majority of the literature was industry experience reports.

From the results of the systematic literature review, Hossain et al. (2011) developed a framework as a basis for future research and for practical use. This framework outlines current research and views on the alleviating effects of Scrum practices on commonly recognized challenges in GSD. Hossain et al. (ibid.) states that challenges can arise in three broad categories; *Communication*, *Coordination*, and *Control* due to temporal, geographical or socio-cultural distance existing in distributed teams. The framework provides a guideline on eight alleviation mechanisms that can mitigate commonly recognized challenges in GSD.

Since the development of the framework, researchers have used the framework to study different aspects of GSD challenges and Scrum. Bannerman, Hossain & Jeffery (2012), conducted an in-depth study on the Coordination-aspect of the framework by following a change from traditional Waterfall development methods towards an iterative Scrum approach, based on the GSD framework. It was found that Scrum offers a distinctive advantage in mitigation of geographical and socio-cultural challenges, but not temporal distance coordination challenges. Noordeloos, Manteli & Van Vliet (2012), conducted a similar study on the transition from Waterfall to Scrum. They found that almost all aspects of GSD work was improved, e.g., that agile methodologies promote more frequent interactions between customers and developers. Khan & Azeem (2014), conducted a study prior to the research framework regarding the socio-culture in GSD projects using Scrum and identified intercultural challenges to look out for. Lastly, Rahman & Das, (2015) conducted

an inductive study based on interviews in their research, where the authors aimed to validate and to extend Hossain et al.'s (2011) research framework.

1.2 Problematization and Purpose

Scrum practices can, as the studies above discuss, mitigate some of the issues occurring in a GSD environment, but not all. For example, Scrum procedures can mitigate cultural misunderstandings through frequent communication i.e, Daily Scrum-meetings. However, Scrum cannot mitigate all challenges, e.g., perceived threats from low-cost rivals is a commonly recognized challenge that no study has yet proved that Scrum by itself can mitigate. It is, therefore, important to see how these issues, which are still prevalent when using Scrum are being addressed. Furthermore, most studies have had a focus on the feasibility of Scrum and not the distinctive contributions of using that framework. The Hossain et al. (2011) framework is as such founded upon experience reports and data from interviews with practitioners and is, therefore, lacking in-depth studies. If it was included, it would further strengthen the implications of this framework for practitioners and researchers using Scrum in a GSD environment.

This ethnographic study, which is conducted on location, differs from prior research firstly by being an in-depth study of distributed Scrum practices of a Swedish company in India. Secondly, this study examines a company that started its operation in India, 11 years ago. This company has been using Scrum practices between the two countries throughout its operation and has hence, learnt a successful best practice that can be compared to contemporary literature through observations and interviews. The implication of this, is the possibility to evaluate Hossain et al. (2011) with a particular case in a particular scenario. This is something that the articles, the framework is based on has lacked prior to this study. By applying Hossain et al.'s (2011) framework in a practical scenario by studying two different offshore development projects from Sweden to India, the authors can identify challenges in the case studies, i.e., the framework can be empirically tested in a distributed Scrum-project in a GSD-context in practice. Hence, *the purpose of this study is to both evaluate and provide suggestions for expansion of the Hossain et al. (2011) framework with ethnographically collected empirical support, which prior to this was primarily based on experience reports. This study also aims to identify GSD challenges and mitigation strategies that occur in the setting of an experienced organization conversant with Scrum methodology in a GSD context.*

1.3 Case Study

In order to evaluate the Hossain et al. (2011) framework and identify GSD challenges and mitigation strategies in relation to a Scrum-based organization, Indpro AB, a software development company founded in Sweden and based both in Bangalore and Stockholm has been studied. Indpro applies Scrum practices to all of its customer projects and this study assessed two of those projects. All of Indpro's customers are based in developed countries, primarily Sweden. Indpro has more than a decade of experience in India and has consequently developed its own best practice and delivery model incrementally. Due to the sensitive nature of our study, all participants except Indpro have been promised confidentiality and been anonymized in the report. This includes customers and interviewed employees of Indpro.

1.3.1 Case Study, Company X

Company X is a Swedish web-based retailer and was one of the first customers of Indpro. The original project was intended to last six months but due to the success of the first project they now have a dedicated team from Indpro in their development unit. That unit consists of four development teams with a team of four Indpro members stationed in Bangalore and the rest in Sweden. All four teams share the same Product Owner (Scrum Terminology will be explained further in Section 2.1 *Scrum*) but have separate Product Backlogs. None of these teams have a designated Scrum Master, but it is the Product Owner who assumes that role when necessary. Indpro has been involved with Company X since the beginning of their operations and was the original developer of their website and e-commerce platform.

1.3.2 Case study, Company Y

Company Y is a Swedish IT-company that monitors software environments. Indpro provides Company Y with a dedicated team consisting of three members from the office in Bangalore. The teams in Sweden and India share the same Product Owner and the same Product Backlog but have different Sprint Backlogs. Company Y originally contracted Indpro for a single project, in which Indpro developed a new product which complemented their core product. Company X was delighted with the result and elected to extend the collaboration. The team supplied by Indpro now works dedicated to Company Y with developing and improving their core product. The team in India consist of three members including a Scrum Master (see section 2.3.3).

2. Theory

2.1 Scrum

This chapter begins by defining the concept and history of the Scrum Methodology (section 2.1-2.5). This is followed by an overview of the GSD Challenges (section 2.6). Finally, the research framework for this study is presented (section 2.7)

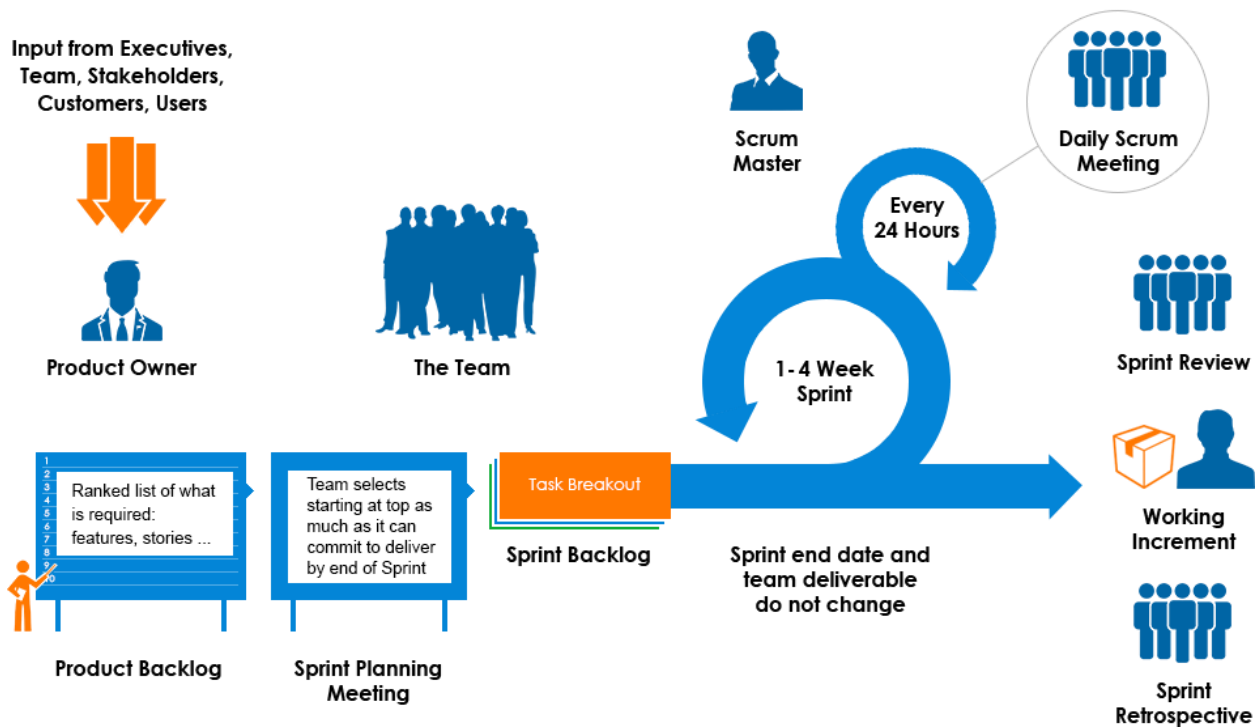
2.1.1 History & Definition

The agile software development framework Scrum was first described by Takeuchi and Nonaka (1986) in the terms of product development in 1980's Japan. They proposed a new way to develop products using a holistic approach instead of the previous sequential approach. The authors described the holistic approach with a parable to rugby where a team passes the ball within the team to move forward (Takeuchi & Nonaka, 1986). They mentioned the term Scrum in reference to the strategy in rugby to restart play. The analogy was made between the two because of the similarities in that both are adaptive, quick, self-organizing, and have few breaks (Schwaber & Beedle, 2002). Later on Sutherland & Schwaber (1995) presented a paper which became the foundation of the modern Scrum methodology.

Scrum is defined as a framework where people can address complex adaptive problems. This whilst creatively and productively deliver the highest possible valued products. It means that Scrum is not a process or technique merely for building products, rather it is a framework where one can employ various processes and techniques. Hence, Scrum is used for product management, development practices and to improve efficacy (Schwaber & Sutherland, 2016). The benefits of using Scrum is based on collocated and tightly collaborating teams (Kniberg, 2015), and it can therefore be problematic using Scrum in a distributed environment. Because of this issue, Kniberg (Ibid.) encourages practitioners to start with one big team with distributed members and after a while separating the teams after their geographic location.

The Scrum framework encompasses roles, events, artifacts, and rules. They all serve a specific purpose and the rules of Scrum bind them together and governs their relationship and interaction between them. In Graphic 1 from Hristovski's (2017) an overview of the Scrum Process is shown.

The Scrum Team consists of the Product Owner, the Scrum Master and the Development Team which will be described further in section 2.3 *Scrum Team*. In section 2.4 *the Scrum Events*; The Sprint, Daily Scrum, Sprint Retrospective and Sprint Review will be reviewed. In section 2.5 *Scrum Artifacts*; the Product Backlog, the Sprint Backlog and Definition of Done will be presented.



Graphic 1 from Hristovski (2017)

2.2 Scrum Theory

Scrum is derived from empirical process control or empiricism. Empiricism means that experience and decision making, based on what is known creates knowledge (Schwaber, Sutherland and Beedle, 2013). Scrum addresses complex software development and is a tool that seeks to achieve control over the empirical process. Not control in a sense of to predict the outcome, rather to control the process to achieve the most valuable outcome (Schwaber & Sutherland, 2010).

There are three pillars that uphold the empirical process control: *Transparency*, *Inspection* and *Adaptation*. This is done by a set of simple practices and rules.

2.2.1 Transparency

Transparency means that the process and the outcome must be visible to the whole team and those responsible for the project (Schwaber & Sutherland, 2016). Furthermore, it requires that the team has shared standards and a common understanding of what they see. McKenna (2016) mentions that everything has to be out in the open for all to see regarding the project. All the accomplishments and failures must be shared and he labels it as oversharing. He puts forth cultural identities which might hinder the adoption of this pillar, i.e, the American culture loves a winner, but there is a tangible fear of being wrong. What's more, in a command-and-control culture the fear of being

wrong is magnified. India as a high distance power culture (Ranjan Kumar & Sankaran, 2007), can consequently be regarded as a culture with an element of a command-and-control culture. Hence, transparency in Indian culture is an important aspect for Scrum methodology. In summary, transparency aims to visualize significant aspects of the process and thereby establish common standards in the Scrum team.

2.2.2 Inspection

Scrum users must frequently inspect artifacts and progress towards the sprint goal so unacceptable variances can be detected (Schwaber & Sutherland, 2010; McKenna, 2016). However the inspection should not be so frequent so that it gets in the way of the work (Schwaber & Sutherland, 2016). The inspection process should be performed carefully by skilled inspectors to be of value (Schwaber & Sutherland, 2016).

2.2.3 Adaptation

If the inspector finds any aspects that deviate from the agreed limits and this will result in an unacceptable product, the process or material that is being worked on must be adjusted. This must be done as soon as possible. There are four formal events that are considered in Inspection and Adaptation; Sprint planning, Daily Scrum, Sprint Review, and Sprint Retrospective. These will be explained further in section (2.4) *Scrum events*.

2.3 Scrum Team

2.3.1 Product owner

The Product Owner is officially responsible for maximizing the value of the product and the performance of the Scrum Team (Schwaber & Sutherland, 2016). This person is the interface between stakeholders and the Development Team, and is responsible for managing and controlling the Product Backlog (Diebold, Ostberg, Wagner and Zandler, 2015; Schwaber & Beedle, 2002). The role of a Product Owner also includes the responsibility to acquire initial and additional funding which is done through setting ROI objectives and release plans (Schwaber, 2004). The Product Owner is a single person who must ensure that the Product Backlog is clear and transparent to all members which is done through prioritizing the items (usually called stories) in the Product Backlog (Diebold et al., 2015). The stories do not inform the Development Team on how to accomplish the increment but rather on what needs to be created. As the Product Owner is the sole person allowed to change the Product Backlog, it is critical that the development team has access and possibilities to discuss the Product Backlog with the Product Owner (Kniberg, 2015).

2.3.2 Development team

The second role is the Development Team, which contains members with cross-functional skills (Davis, 2013). This team is responsible for producing and delivering a potentially releasable product increment of what they consider done (Schwaber & Sutherland, 2016). This team is self-organizing, in that the group itself determines how to turn stories from the Product Backlog into a releasable product increment (Ibid.). The only constraints that a team has are organizational standards and the Product Backlog they work with (Schwaber & Beedle, 2002). They take on as many stories from the product backlog that they think they can deliver during each sprint, which usually lasts a 2-3 week period (Ibid.). According to Schwaber & Sutherland (2016) the development team should be small enough to stay flexible and large enough to complete major sprints. They argue that a team should have between three to nine members to make coordination manageable yet still have the required skills available.

2.3.3 Scrum Master

The Scrum Master holds responsibilities to three main stakeholders: the Product Owner, the Development Team and the organization as a whole. The Scrum Master is the servant-leader of the Scrum Team (McKenna, 2016; Schwaber & Sutherland, 2016). The Scrum Master should act as a coach and facilitator for the Scrum Team and also make sure that the team adheres to the Agile-, to be more precise, Scrum-framework (McKenna, 2016; Davis, 2013; Diebold et al., 2015). He or she should do everything necessary to make the team successful, including protection of the team and removal of impediments that hinder their ability to perform at a high level and facilitate Scrum events which are requested or needed (McKenna, 2016; Schwaber & Sutherland, 2016). Being a coach to the Scrum Team implies that the Scrum Master should help the team come up with solutions themselves instead of solving it him or herself (McKenna, 2016). The Scrum Master has responsibilities toward the organization as a whole, comprising of leading and coaching the organization in its implementation and adoption of Scrum practices and collaborating with other Scrum Masters (Scrum-of-Scrums) to improve the application of Scrum practices in the organization (Ibid).

2.4 Scrum events

To create regularity and to avoid having meetings not included in the Scrum process, there are prescribed events taking place in the Scrum framework. All events are set into a specific set of time. This includes whole sprint itself, which is set to a maximum of four weeks before iteration (Schwaber & Sutherland, 2016). Events included in each sprint are; Sprint Planning, Daily Scrum, Sprint Review and Sprint Retrospective. All of the above will be explained further below accompanied with a general overview of the sprint in general.

2.4.1 The Sprint

As previously mentioned, the team works in sprints over several iterations until the targeted outcome from the product backlog is reached. The sprints usually last 2-4 weeks, though it can differ from team to team (Diebold et al., 2015). Importantly though, the sprint cannot be longer than a calendar month as the definition amongst the team members can change, complexity may arise and the overall risk may increase (Schwaber & Sutherland, 2016). During the sprint, there can be no changes that will endanger the sprint goal, as the quality of goals cannot decrease, however, the scope may be clarified during the process and lead to the Product Owner and the Development Team to re-negotiate the Sprint Planning as more is learned (Schwaber & Sutherland, 2016). The sprint can only be abandoned by the Product Owner, that is if the sprint does not meet the aim of the project any longer. After every sprint, an increment is produced (Diebold et al., 2015). When a sprint is started, it cannot be lengthened or shortened and every event within it is time-boxed and has a maximum duration. Furthermore, every event must be designed in a manner that allows inspection and critical transparency (Schwaber & Sutherland, 2016).

2.4.2 Sprint Planning

In the Sprint Planning both the Scrum Master, the Product Owner and the Development Team is present. The aim with the Sprint Planning is to define the Sprint Goal of the sprint (Diebold et al., 2015), but the team must also plan how they will achieve the aim during the sprint period (Diebold et al., 2015; Schwaber & Sutherland, 2016; McKenna, 2016). A Sprint Goal is a short statement of what the team plans to accomplish during the sprint (Davis, 2012). The time for the meeting is set to a maximum of 8 hours for a 30 day sprint, but is usually shorter for shorter sprints (Schwaber & Sutherland, 2016). The Product Owner sets what is of the highest priority from the Product Backlog, which he or she is presenting for the Development Team. The team will then negotiate the terms of the content, purpose, meaning and intentions of the Product Backlog and when the team knows enough, they select as much they believe they can convert into a functional product at the end of the sprint (Schwaber, 2004). The events from this will be put as stories in the Sprint Backlog, and the team will, with the help of the Scrum Master, plan how they will manage their time (McKenna, 2016). The tasks in the Sprint Backlog will emerge further as the sprint evolves (Schwaber, 2004). Now the team, together with the Product Owner and the Scrum Master, has answered; what will be delivered in the end of the upcoming sprint (Sprint Backlog) and how they will work to achieve a functional product at the end of the sprint.

2.4.3 Daily Scrum

Within the sprint, there is a daily iterative meeting held by the Scrum Team, called a Daily Scrum (McKenna, 2016). This is typically a 15 minute stand up that is supervised by the Scrum Master, to inspect the work that has been done since the last stand up and what needs to be done until the next Daily Scrum. The Scrum Master merely proceeds to make sure that the Development Team can have the meeting, but it is up to the Development Team to conduct the Daily Scrum (Schwaber & Sutherland, 2016).

The stand ups often form based on three questions for all members to answer. *What did I accomplish yesterday that helped the Scrum Team meet the Sprint aim?, What will I do today to help the Scrum Team meet the Sprint aim? And, have I encountered any impediment that could interfere with me or the Scrum Team to meet the Sprint aim?* (Diebold et al., 2015; Schwaber & Sutherland, 2016). This way, the Scrum Team can inspect progress toward the Sprint Goal (Schwaber & Sutherland, 2016). The reason for these 15 minute meetings are to ensure collaboration among team members, to enable efficient work, to resolve problems, to improve communication, eliminate the need of other meetings, highlight and promote quick decision-making, and improve the Development Team's level of knowledge (McKenna, 2016; Schwaber & Sutherland, 2016).

2.4.4 Sprint Review

At the end of each sprint, a Sprint Review meeting is held. Here the Development Team presents what they have developed in the sprint process. Participating is the Product Owner and other stakeholders, and the team presents the viable product, preferably by doing a demo (McKenna, 2016; Schwaber 2004). Here the Development team gets feedback from the Product Owner and stakeholders, this is to ensure that they can change direction if the product is not satisfactory, but even, release the product if stakeholders and the Product Owner is satisfied according to what is stated in the Product Backlog (McKenna, 2016; Schwaber & Sutherland, 2016). The acceptance is based on a common Definition of Done (Diebold et al., 2015), a term that will be explained further in Section 2.5.3. Hence, this informal meeting is meant to present the functionality and bring people together to determine what to do next. The result of the meeting is a revised Product Backlog that will define the probable items for the next Sprint Backlog in the upcoming Sprint (Schwaber & Sutherland, 2016).

2.4.5 Sprint Retrospective

After the Sprint Review is the Sprint Retrospective. This is a time to reflect, inspect and identify what can improve in the next sprint. The Scrum Master ensures that everybody from the Development Team is involved and understands its purpose. The team should find improvements that can be implemented in the process of the next Sprint, and even modify the current Definition of Done (Schwaber & Sutherland, 2016). McKenna (2016) makes the analogy between this meeting and a closed-door meeting in team sports, when the team holds a meeting without any press or coaches, in an effort to identify what needs to be addressed and how they plan to address them. Subsequently, the Development Team holds the meeting, with preferably only the Scrum Master according to McKenna (2016). The Product Owner can participate if one can ensure that the Development Team feels comfortable and can be transparent enough in their self-inspection.

2.5 Scrum artifacts

2.5.1 Product Backlog

The Product Backlog is a constantly evolving list of requirements needed for a product (Schwaber, 2004). A Product Backlog is dynamic in that requirements are added to the list whenever the product-environment evolves (Schwaber & Sutherland, 2016). This list consists of functional and nonfunctional requirements. The items or requirements added to a Product Backlog needs to be clear and is sometimes stated as user stories whereby the Product Owner prioritizes these by assigning a value to each (Davis, 2013). This makes it easier for the Product Owner to organize the Product Backlog. The Development Team adds story points to each story to state the amount of effort needed to complete the story. The Development Team and Product Owner also engage in the process of Product Backlog refinement, which is the act of revising and reviewing added items to the Product Backlog (Schwaber & Sutherland, 2016).

2.5.2 Sprint Backlog

Stories chosen by the Development Team for the sprint, a plan for delivering the increment, as per the established Definition of Done, and a plan for realizing the sprint goal is set up in the Sprint Backlog (Schwaber & Sutherland, 2016; Davis, 2013). The Sprint Backlog consists of stories decomposed into tasks (McKenna, 2016; Davis, 2013), and intends to make visible the work necessary to meet the sprint goal (Schwaber & Sutherland, 2016). Before moving a story from the Product Backlog to the Sprint Backlog the Development Team must ensure that the story complies with their Definition of Ready, which is a collection of everything necessary to make it possible for a story to be developed (McKenna, 2016). Before the Sprint Backlog, the Product Owner usually has a meeting with the team for a Product Backlog refinement meeting where they discuss the different stories and prioritize the backlog.

2.5.3 Definition of Done

Transparency is one of the core themes in Scrum and to ensure that it is adhered to, Scrum Team members must have a shared understanding of what it means to be done (Schwaber & Sutherland, 2016). Agreeing on a Definition of Done assures every team member their expectations will be met, and there won't be any surprises or misunderstandings during the sprint (McKenna, 2016). While the Definition of Done can differ between different Scrum Teams, it must be consistent within one team and for every story that the team works on, however, all parts of the definition may not be applicable for every. The Definition of Done is distinguished from the acceptance criteria set up by the Product Owner, which is specific to a single story (Ibid.).

2.6 Challenges in GSD

There are several reasons why companies increasingly seek to work in a more distributed fashion. GSD often offers a way to simultaneously achieve a more skillful and cost-efficient workforce as well as a higher proximity to markets, and a flexibility to respond to local differences (Bannerman et al., 2012). Furthermore, GSD can reduce time to market, increase productivity, improve quality, and provide cost efficiencies for software developing organizations (Jiménez, Piattini, & Vizcaíno, 2009). Even though there are many advantages of using GSD, there are plenty of challenges to overcome. It can be harder to participate, understand and communicate in a distributed environment. If the quality of the telecommunication technology is poor, it can increase the cultural and knowledge distance between the sites (Paasivaara & Lassenius, 2010). The issue with participation makes it harder for the distributed members to contribute in meetings (Ibid.) There is a growing body of literature on challenges in GSD, most are based on three categories of challenges; Communication, Coordination and Control which may arise due to temporal, geographic and socio-cultural distances that are encountered in GSD (Ågerfalk, Fitzgerald, & In, 2006; Hossain et al. 2011; Bannerman et al., 2012).

Temporal distance is the measure of the dislocation of time between two different people with the goal to interact. This may create issues such as a difficulty to hold simultaneous meetings, reduced work hours together with distributed colleagues and delayed responses (Ågerfalk, Fitzgerald, Holmström, Lings, Lundell & O'Conchuir, 2005; Hossain et al. 2011; Bannerman et al., 2012). Subsequently, GSD processes may be significantly affected (Ågerfalk, 2006).

Geographic distance is the issue of not being able to meet up on the same location (Abrahamsson, Salo, Ronkainen & Warsta, 2002), e.g., making it difficult to have face-to-face meetings (Ågerfalk

et al., 2005). This can lead to lack of critical task awareness, “teamness” and reduced trust (Abrahamsson et al., 2002; Hansen & Baggesen, 2009; Moe & Šmite, 2008; Bannerman et al., 2012).

Socio-cultural distance may cause inconsistent work practices, different perceptions of authority, and lack of mechanisms for creating a shared understanding among team members, as well as misunderstandings and reduce cooperation (Hossain et al. 2011; Bannerman et al., 2012).

These distances, which can be prevalent between different teams and team members can affect the processes of Coordination, Communication and Control (Almeida & Albuquerque, 2011).

One of the main challenges which can arise when working with GSD is the subject of effective Coordination and Communication among teams (Al Zaidi & Qureshi, 2014). According to Al Zaidi & Qureshi (Ibid.) Coordination is a set of actions which teams undertake to achieve a common goal, thus it is imperative that the teams have a shared understanding. This is difficult in a GSD environment since many of the coordination mechanisms that work in a collocated setting are absent (Bannerman et al., 2012). The distances presented above can create issues in the Coordination category, for instance through reduced informal contact due to geographical distance and different work practices due to socio-cultural distances. These distances can impinge on the shared understanding of the distributed teams. According to Almeida & Albuquerque (2011) a critical factor for the success of a project in a distributed environment is effective communication. Challenges for the communication category can appear because of the lack of synchronized work hours and not having the possibility to have face-to-face meetings (Ågerfalk et al., 2005). The Scrum methodology emphasizes on communication while reducing Coordination and Control overhead (Almeida & Albuquerque, 2011), so issues in the control category are bound to occur. Issues here can be the difficulty to convey vision and strategy due to geographical distances and different perceptions of authority between the onshore and offshore teams due to different socio-cultural backgrounds (Hossain et al., 2011).

2.7 Hossain et al.’s (2011) Framework

Hossain et al.’s (2011) framework is developed to provide mitigation strategies that Scrum brings to generally occurring GSD challenges. Below there are three challenge categories; Communication, Coordination and Control, where different sub-groups of challenges occur, i.e.; CA1-CA3, CB1-CB4 and CC1-CC5. For almost every GSD-challenge, there are one or more mitigation strategies. These are GSD_P1-GSD_P8. Every challenge in the framework (CA1-CC5) is related to a particular source characteristic; Temporal distance, Geographical distance, and Social Cultural distance. For example, *CA2 Face-to-face meeting difficult* is a geographical distance generally

recognized in GSD, which is mitigated by Scrum methodology itself, by ICT-mediated communication (GSD_P2 & P3) and *Visits* (GSD_P4), according to the literature that the framework is based on.

Challenge Category	Challenge Description	Mitigation Strategy
Communication (CA)	CA1. Reduced opportunities for synchronous communication	GSD_P1. Synchronous work hours, GSD_P3. ICT-mediated asynchronous communication
	CA2. Face-to-face meetings difficult	GSD_P2. ICT-mediated synchronous communication, GSD_P3. ICT-mediated asynchronous communication, GSD_P4. Visit.
	CA3. Cultural misunderstandings	GSD_P3. ICT-mediated asynchronous communication. GSD_P4. Visit. GSD_P5. Frequent (or Improved) communication.
Coordination (CB)	CB1. Increased coordination costs	GSD_P1. Synchronous work hours. GSD_P3. ICT-mediated asynchronous communication
	CB2. Reduced informal contact can lead to lack of critical task awareness	GSD_P2. ICT-mediated synchronous communication GSD_P3. ICT-mediated asynchronous communication GSD_P4. Visit. GSD_P5. Frequent (or Improved) communication GSD_P6. Iteration GSD_P7. Review GSD_P8. Planning

	CB3. Inconsistent work practices can impinge on effective coordination.	GSD_P5. Frequent (or Improved) communication. GSD_P6. Iteration
	CB4. Reduced cooperation arising from misunderstandings.	GSD_P3. ICT-mediated asynchronous communication GSD_P4. Visit. GSD_P5. Frequent (or Improved) communication GSD_P6. Iteration GSD_P8. Planning
Control (CC)	CC1. Management of project artifacts may be subject to delays	
	CC2. Difficult to convey vision and strategy	GSD_P4. Visit. GSD_P5. Frequent (or Improved) communication GSD_P7. Review GSD_P8. Planning
	CC3. Perceived threat from training low cost "rivals"	
	CC4. Different perceptions of authority can undermine morale	GSD_P5. Frequent (or Improved) communication
	CC5. Managers must adapt to local regulations	

The table below (Table 1) shows the mitigation mechanisms the framework proposes to overcome the commonly recognized challenges arising from Geographical, Temporal and Socio-cultural distances. These mechanisms are not only applicable to Agile development but also to other non-Agile methods. However the underlying principles of Agile practices imply that Scrum practices can achieve additional benefits from the use of these mechanisms. GSD_P5 to GSD_P8 are mechanisms that are inherent of Scrum practices in themselves.

Category	ID	Description
Synchronized work hours	GSD_P1	Increase overlapping working hours between sites to enable synchronous communication for meetings; for example, adjust working hours at sites to create some overlap or participate in meetings from home
ICT-mediated synchronous communication	GSD_P2	Practices that enable synchronous formal or informal communication between teams; for example, use individual or conference phone calls, teleconference, video conference, web conference, or application
ICT-mediated asynchronous communication	GSD_P3	Practices that enable asynchronous communication between team members; for example, email, Instant Messaging, or Wiki
Visit	GSD_P4	Face-to-face meeting made possible by travelling between sites. Two main kinds: seeding visits to build relationships, and; maintaining visits to sustain relationships
Frequent (or Improved) communication	GSD_P5	Enable frequent formal and informal communication among team members through tools and/or face-to-face meetings
Iteration	GSD_P6	Activities that involve cyclical repetition enable multiple incremental opportunities to monitor progress and resolve issues
Review	GSD_P7	Formal or informal activities that enable reflection on prior activities, assessment of completed work, and the opportunity for stakeholders to provide feedback to the teams
Planning	GSD_P8	Activities that establish the scope of work, resourcing, scheduling, and the processes to be employed

Table 1 from Hossain et al. (2011)

As Hossain et al. (2011) mentions, this research framework implies there is a generic GSD context. GSD can take many forms in different contexts. As the research framework is yet to be empirically tested and is based on experience reports, certain issues might not be in the framework or issues in it may not be prevalent in a practical scenario. Lastly, it is unlikely that there can be one “ideal” method for every context in GSD. When studying the case of Indpro with this framework, the entirety of the framework may or may not be relevant. However, the case of Indpro is suitable to explore an experienced vendor using Scrum in a GSD context for almost 11 years. The case of Indpro can be used as a model for future Scrum-related research and other mitigation strategies in a GSD context.

3. Method

This chapter focuses on the selection of research methods appropriate for this study. The chapter begins with a Research strategy (section 3.1) and continues with an explanation of the Research design (section 3.2). This is followed by Data collection (section 3.3) and finally by Analysis of data (section 3.4) that describes in which way the collected data was systematically analyzed.

3.1 Research Strategy

The study empirically evaluates the Hossain et al. (2011) framework and extends the framework based on the empirical data collected through an ethnographic case study on a Swedish company primarily based in India. This study takes an inductive approach to research where the authors will refer the implications of their findings from the field study to the theory (Bryman & Bell, 2015). It can however not be definitely stated that the research is completely of an inductive nature. Rather, it has aspects of inductive research in it, as opposed to a deductive research approach. Thus, the study will also give implications from the theory to practitioners.

3.2 Research design

Previous research regarding mitigation effects of Scrum for GSD challenges has been carried out by many researchers e.g Bannerman, Hossain & Jeffery (2012); Noordeloos, Manteli & Van Vliet (2012); Khan & Azeem (2014) using Hossain et al. (2011)'s framework (see Section 1 *Introduction*). The unique contribution of this case study, as opposed to the majority of experience reports that the Hossain et al. (2011) framework is based on, is the in-depth nature achieved through an ethnographic study that is conducted on location in India. The members of the organization in India works as distributed and dedicated Scrum teams towards Swedish customers. Because of the in-depth nature required for this study, a qualitative research approach will be applied based on both semi-structured interviews and participative observations.



Graphic 2

Above (Graphic 2) is a timeline of the six weeks spent in Bangalore, India. Two weeks of pre-study were followed by four weeks of interviews. During all six weeks observations were conducted. All of these will be motivated and explained further in the sections below.

3.2.1 Participant observations / Ethnography

Participant observations, which in latter years scholars have started calling ethnographic research, derives from anthropology. Historically, researchers have usually used ethnography to get access to a group of people and study its culture by watching and listening to the uncovering of events and taking notes during an extended time (Bryman & Bell, 2015). The roles of the authors of this study in the observed organization is what Gold (1958), referred to as, Participant-as-observer. Which is the role the researcher takes as a fully functioning member of the social setting whereby the members in it are aware of the researcher's status as a researcher. As such, the authors were engaged with the observees in their daily lives, work and were transparent about the research aim and objectives. This approach was embraced by the authors of this study. Bryman & Bell (2015) mentioned the research of Perlow (1997) as a typical account of ethnographic research. In her study concerning work-life issues of engineers she mentions working and observing in their working environment as well as asking to get invited to their homes and was always open to having discussions as soon as the chance arose. This approach of doing ethnographic research is applicable to this study as the authors shared an apartment with two developers part of Company X, in addition to spending time with the observed subjects on and off work. It is important to note that the authors of this study did not partake in the actual working tasks of the observees, and functioned merely as researchers and passive observers in the daily work activities. Bryman & Bell (2015) calls this passive observation, i.e, not participating in the value providing activities for the organization. The observations lasted for 6 weeks in Bangalore, where the authors spent 5 full working days a week at the office. This time was managed with both planned observations during meetings, as well as observing daily work and informally interviewing and discussing work with the observees. Time was also spent writing this thesis and conducting semi-structured interviews which is explained further below.

3.2.2. Pre-study

The authors planned a pre-study for the first two weeks after arriving in order to determine how the Scrum methodology and GSD challenges were approached and managed in comparison to the framework. Furthermore, it was studied exactly how the teams were structured. After these two

weeks of pre-study, the authors found themes which were viable to study. These themes corresponded to challenges present in the framework and was used as a foundation to develop an interview guide and strategy for the observations (which will be discussed further in Section 3.3 *Data Collection*). This way, the authors could get a preliminary view of the organization, e.g., how it worked, how teams were structured, and in what way the Scrum methodology within Indpro differentiated from the literature. Also worth mentioning is that before arriving in India, the authors had two teleconference meetings with the deputy CEO and one face-to-face meeting at the Stockholm office with both the deputy CEO and the CTO. This was to get an initial view on how they worked with GSD and Scrum but also to get to know which teams and projects that worked distributed, in what way, and to know which teams the authors could get access to. Furthermore, these meetings were important in order to get a general picture of how the teams were organized in the two planned case studies before going to India. This way it was possible to get to know the limitations and the spectrum of access early in the process, as well as to determine in what way this study could contribute to the current body of knowledge. To make the employees comfortable and relaxed in our presence, the authors spent the first day memorizing the names of everyone in the open-landscape office where the authors sat and invited them home to get to know them further. The identified themes from the pre-study was used to modify the framework accordingly and to make the study viable, as the limitation of this study could not cover all of the diverse challenges and mitigation-strategies in the framework. This modified framework (see more in 4.1 *Adjustments to Framework*) was then used as a reference point that was put in relation to the ethnographic findings at Indpro. This way it could be studied in-depth, in what way the findings might differ or conform to the literature.

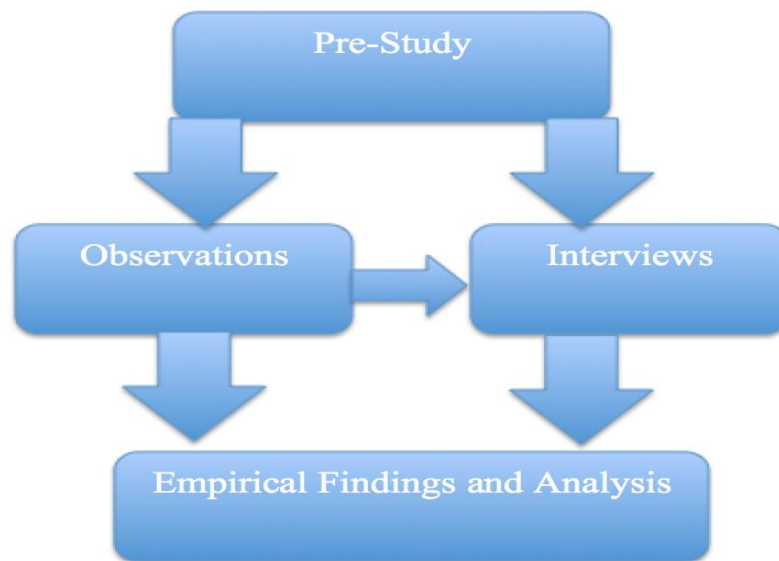
3.2.3 Semi-structured interviews

Semi-structured interviews were used to complement the empirical data from the observations, and used as an opportunity to go beyond what was observed from the people, groups and the organization in general. Eisenhardt (1989) suggests that the choices of methodology should complement each other in the best way possible to answer the purpose of the study, hence, the data collected for this study will mainly be collected from observations, complemented with interviews to gain more depth.

For this study a strategic selection of interviewees was preferred. This means that the respondents were selected in order to provide answers for the mitigation effects Scrum has on GSD from the viewpoint of an experienced vendor working between Sweden and India. In a strategic selection, individuals are selected based on their knowledge in a specific context (Bryman & Bell, 2015), Saunders (2009) argues that this approach is appropriate when the study works with a limited

population of people and is desirable when the respondents have much knowledge in the area of the study. The process of strategic selection has been applied through choosing Indpro as the company of study in general, because of its distributed business activities between Sweden and India for more than 11 years.

The interviews were based on the pre-study and were therefore planned to start on the third week after arriving in Bangalore (See Graphic 3).



Graphic 3

It could be argued that the interviewees felt more comfortable around the authors in the setting of the interview after spending time together on and off work. But on the other hand, it is always difficult to avoid affecting the interviewee during interviews and this will likely have an affect on their answers. Subsequently, if this study was would have been conducted by different authors, it could be argued that the outcome from the interviews would likely have been different. All of the interviews were recorded and transcribed, to make sure the interviewees answers would not be distorted, making sure that the data from the interview was as reliable as possible.

This is a qualitative study using semistructured interviews with a limited population. This affects the external validity and subsequently makes it difficult to replicate as compared to a quantitative study. However, this study can provide introductory research for future qualitative studies in this area (Saunders et al., 2009). To ensure as high validity as possible, an example of the basis for the interview guide is attached as an appendix (*Appendix 2*). It should be noted that the interview guide

was adapted according to which role and which person that was interviewed and based on what was found in the pre-study.

3.2.4 Selection process and sampling

After meeting with management, two suitable case studies were selected. This selection was based on two factors, first, which projects the authors could get good access to, and second, which projects and individuals within it had current and prior experience of working with Scrum in a distributed manner. The strategic selection in this case study was to interview all Indian team members of the two case studies and four members of the management.

An important aspect to take into consideration is that if this study was conducted in collaboration with a different company, the outcome would likely be different. As stated, the selection of Indpro presumably has an effect on the results of this study. Bryman & Bell (2015) argues that this is a common critique of qualitative research and its generalizability. For this study, the findings of mitigation strategies that Scrum brings to GSD cannot be generalized to other contexts. The generalizability of this study can rather be assessed from its theoretical conclusions from the empirical data (Bryman & Bell, 2015). The 12 interviews conducted was a way to dig deeper from what was found in the ethnographic data from the 6 weeks of observation. Kvale (1997) argues that one should not conduct too many interviews in a qualitative study to avoid issues interpreting large volumes of data. The ethnographic nature of this study yields a large amount of data, and the authors therefore concluded that the material from the observation in combination with 12 interviews was sufficient for this Master's thesis.

3.3 Data collection

3.3.1 Observations

As this qualitative study has an inductive research strategy that examines two case studies ethnographically, the purpose was not completely set in the beginning of the six week visit. A challenge in working with this approach is the risk of collecting data that is irrelevant, because the spectrum when observing can become too broad. An example of this is Kunda (1992) that describes how he was being swamped with information, partly because of not seeking to define his focus of study. To avoid this challenge, both authors studied the research framework thoroughly before starting to observe, to make sure that there were some limitations in what to look for. This way, the observations were operationalized and bound to theory, as well as making sure to avoid data overload. The observations took place in many different areas. Mainly in the shared workplaces for the two studied teams (Company X and Company Y) in Bangalore where the developers from each company share the same office, though they are both located in two different office landscapes. Both

authors tried to move between the two office landscape to spend an equal amount of time as possible on both case studies. Furthermore, observations took place in two conference rooms at the office where meetings were held. These notes were structured in a table, where the left column was the actual description of what was going on in the room and what was said, in the right column the authors could make comments and a brief analysis of the occurrences (See *Appendix 1*) this approach was inspired by Wedlin (2017), who describes the importance of separating analysis and interpretations from what is actually observed. During the non-meeting observations, field notes were transcribed before leaving the office each day. This method was used to avoid making the participants self-conscious of their actions and, therefore, cause a Hawthorne effect, which is that the act of observing could change their behavior when being observed (Bryman & Bell, 2015). Ditton (1977) identifies the Hawthorne effect as a way to avoid anxieties. The authors met for around 15 minutes at the end of each working day, discussing what they had done during the day regarding the research and what they had observed (see Appendix 3). This Scrum inspired meeting was held to inform each other of what they had seen during the day, as well as to help remember important aspects of what had been observed. In regards to documenting the empirical data, the authors first is that the authors first wrote down descriptions and comments of what they had seen individually, and after that discussed the occurrences with each other. This was to avoid interference with one's own impressions and interpretations.

Table 2 below shows how many hours the authors spent observing different aspects of the Scrum practices at Indpro.

Type of observation	Hours (approx.)
Daily Scrum	4.5
Sprint Planning	2.5
Sprint Review	2
Sprint Retrospective	2
Backlog Refinement	2
Observation at the office	220
Observation outside the office	65

Table 2

3.3.2 Interviews, Interview guide and Operationalisation

In addition to the participant observations there is 12 semi-structured interviews conducted in this study: five interviews from the Company X case study, and three from the Company Y case study. One interview is with a member that has previously been part of both teams and now is a part of management. There are three more interviews conducted with management of Indpro. For Company Y the whole Indian Scrum Team was interviewed, and for Company X the whole Indian Scrum Team as well as two Swedish developers from two separate Scrum Teams were interviewed (see Table 3 below for a summary of all the interviewees).

Interviewees		
Company X		
Role	Number	
Software developers Sweden		2
Software developers India		3
Total		5
Company Y		
Role	Number	
Software developers India		2
Scrum master		1
Total		3
Management		
Role	Number	
Senior Technical Architect		1
CSO		1
CEO		1
CTO		1
Total		4
Grand Total		12

Table 3

Before conducting the interviews, as for the observation, the theories were reviewed and studied to make sure questions based to this study's purpose were asked. The pre-study worked as a reference in developing the interview guide. This was especially important for the follow-up questions in the semi-structured interviews, because having studied the practices and conversations of the interviewees the authors, knew which challenges and examples to ask about in the semi-structured interviews. The interview guide was developed with this in mind, as Bryman & Bell (2015) argues that there are no particular rules for an interview guide for semi-structured interviews, but there should be prepared questions to guide the discussion within a few themes. The themes for

the interviews was based on impressions from the pre-study and more specific questions were based on the observations as well as the research framework from Hossain et al., (2011).

When the interviews were performed, the authors met with the interviewees in one conference room at the Bangalore office. All interviews were recorded after receiving consent from the interviewees. Afterwards all interviews were transcribed, this methodology due to several reasons. Bryman & Bell (2015), mentions that recording interviews lets the interviewer keep better attention on the discussion when relieved of the burden of taking notes. They mention that this ensures the answers received are not misinterpreted, and also gives the authors a possibility of going back to the answers to review them.

3.4 Analysis of data

The reference frame that has been used for the analysis of this study is analytic induction, which according to Bryman & Bell (2016) is a process of analysing and collecting data alternately. This means that the analysis had already begun when arriving in Bangalore and has as such, affected which parts of the framework that was put into play and what questions that were asked during the semi-structured interviews. After the observations and interviews had been conducted, the authors used the research framework as a reference frame when transcribing the interviews, i.e., analysing the material whilst writing the transcription. Since the interviews followed up on the pre-study and the observations, the themes or rather theory-bound questions had clear connections to the research framework. The authors could therefore systematically find similar themes in all interviews. The observation notes were not structured in the same manner as the interviews, since the interview guide was influenced by the theory after the pre-study. Naturally, the observation notes became more generic and was used to take a step further in the interviews. But also worked as a way to build a context and a way to obtain a greater understanding in general regarding the challenges and mitigation mechanisms of Scrum in GSD for the Scrum Teams in Bangalore.

4 Empirical Findings and Analysis

The *Empirical Findings and Analysis* chapter first presents the adjusted research framework which was based on the pre-study (section 4.1). Then provides a brief definition of the specific ethnographic context (section 4.2). The chapter is thereafter structured according to the three categories found in the research framework, in the sections 4.3 - 4.5. Those sections include *Empirical Findings* and the *Analysis* conducted by the authors.

4.1 Adjustments on research framework

The framework was after the pre-study modified accordingly to the identified themes of challenges that was observed in Bangalore, according to the course of action presented in section *Pre-study* 3.2.2. In the table below are the challenges that have been studied in the framework for this study i.e., the areas where the authors are able to provide empirical data to both evaluate and provide suggestions for expansion of the Hossain et al. (2011) framework

Challenge Category	Challenge Description	Mitigation Strategy
Communication (CA)	CA1. Reduced opportunities for synchronous communication	GSD_P1. Synchronous work hours, GSD_P3. ICT-mediated asynchronous communication
	CA2. Face-to-face meetings difficult	GSD_P2. ICT-mediated synchronous communication, GSD_P3. ICT-mediated asynchronous communication, GSD_P4. Visit.
	CA3. Cultural misunderstandings	GSD_P3. ICT-mediated asynchronous communication. GSD_P4. Visit. GSD_P5. Frequent (or Improved) communication.

Coordination (CB)	CB2. Reduced informal contact can lead to lack of critical task awareness	GSD_P2. ICT-mediated synchronous communication GSD_P3. ICT-mediated asynchronous communication GSD_P4. Visit. GSD_P5. Frequent (or Improved) communication GSD_P6. Iteration GSD_P7. Review GSD_P8. Planning
Control (CC)	CC2. Difficult to convey vision and strategy	GSD_P4. Visit. GSD_P5. Frequent (or Improved) communication GSD_P7. Review GSD_P8. Planning
	CC3. Perceived threat from training low cost "rivals"	

This adjusted research framework was used as a lens through which the empirical evidence was found and analyzed, to be able to provide empirical support to evaluate it. In the next sections the Empirical findings will be presented and analyzed.

4.2 Ethnographic context

The context for the ethnographic study was at the office of Indpro in the World Trade Center building in Bangalore. The office is located on the sixth floor of a tall building, a floor which was shared with five different companies. When entering the office one first encounters a reception, walking further inside the office there are three main landscapes which are mostly divided by which clients the software developers work for. For the two cases companies that the authors followed, each sat in a different landscape within the office. Furthermore, in the office there were two conference rooms, where most Scrum meetings were held (e.g. Daily Scrum, Scrum Planning etc.). Finally there was one smaller kitchen with facilities for water and coffee, one office for HR and one office space for management.

4.3 Communication

4.3.1 Reduced opportunities for synchronous communication (CA1)

In the office in Bangalore, the time difference to Sweden is usually 4.5 hours ahead in the winter time, and 3.5 hours in the summer time. During the authors stay in Bangalore the time difference changed in the last week of the stay to summer time. Usually this meant that the Indian programmers started their day a couple of hours earlier than their Swedish counterparts. The Indian developers preferred to arrive at the office about 2-4 hour later than for the developers working for Company X and Company Y, e.g. a Swedish developer arrives at 8AM UTC+01:00 and an Indian developer at 11AM UTC+05:30. This was of course something that differed from person to person, but as a general observation, this was the case. One manager commented on this observation:

“We [that are located in India] are a couple of hours ahead, which is quite good, actually. So the meetings have to be planned later in the day [here in India]. But they [the software developers] starts so late here [India], and likes to start late, that it has become an advantage for us... Just yesterday an issue came up that needed to be resolved quickly. The Indian team could then solve it when the Swedish developers and the Swedish customer was still asleep. This was thanks to the time difference. So I see it as an advantage rather than a challenge. Though I imagine it is different if the time difference is longer than between India and Sweden, or if the time difference is the other way around”

An Indian software developer working for the customer Company Y also argued that the time difference can be used in different ways, and that they try to use it to an advantage.

“The time difference, we use it in a flexible way, when we finish our day, we send questions to the Swedes if we need to, and then we get the answers before we start the next day. So we use the time-difference to our advantage. So we are increasing our efficiency. If we have to talk to them during the day we just ping them on slack.”

During the authors time in Bangalore, of course, time difference could prove challenging when working distributed. The Scrum teams working with Company Y started out as a large Scrum team, consisting of about 12 people in one team. But later split up into different teams based on geographic location, which was the case during this study. This meant that the Indian team still works from a common Product Backlog but has their own Scrum events. The teams communicate between one another through Scrum-of-Scrums meetings that are held between the teams different

Scrum Masters. But even though they are not always working together, it can occur dependencies between the Swedish and Indian teams. The Scrum Master commented on this:

“I can say that time difference is one of the most important things. But also initially we all three were new guys. so, we had a lot of dependencies and there were a lot of problems that we used to face. So if we ran into a problem and needed any help from those guys we had to wait for half a day. And then those guys came online and then we used to, we had to clear our queries. But now we have become more experienced in the product and now we know how to solve our own problems.”

A software developer working for Company X had a similar experience with the time difference:

“Sometimes it will be a bit of a lag, when we don’t get answers directly.”

At the office this was something that could occur when an issue arose late on a Thursday in India, and there was some dependency towards the Swedish developers and they had already left work. The query couldn’t get answered until the next day. But a software developer for Company Y commented on this issue when asked about it.

“It is all about planning really, these type of things mainly arise when the teams are new or when you are new to a product. Working with Scrum and going through the constant iteration and the planning that comes with it, sometimes take away such challenges.”

Indpro has teams for other projects where they adapt the working hours to fit between the time zones. They have also tried to go from distributed Scrum teams to more geographically independent teams, i.e. working with distributed Scrum-of-Scrums. One C-level manager spoke about the importance of planning when managing time difference:

“We have tried to work with both one team with Indians and Swedes, but now we work separated mostly because of time-zone related issues that occurred before. The good thing with a distributed Scrum Team was that it is easier to spread knowledge and to get a team-ness, but it was an issue with the time difference... We want a fast interaction between the teams (now, when using separated teams), that’s why we use slack, an instant messaging platform. If our developers want help with for instance a code review [from India to Sweden or vice versa] as the last thing they do, we will of course encounter problems. It is

important that the developers plan their work meticulously when they work in a distributed fashion.”

4.3.2 Analysis of Reduced opportunities for synchronous communication (CA1)

At Indpro they used *Synchronized work hours* (GSD_P1) for some teams. But mainly, both developers and managers stressed the importance of the developers ability to plan their time and be proactive on issues that will or can arise during a sprint, and to take action early. The iterative nature of Scrum is something that can help developers notice issues in good time to take necessary action. For example the Daily Scrum meetings forces the team members to reflect and verbalize what they have done during the day to help the team meet the sprint goal, as well as what issues that has arisen that can prohibit the team from reaching that sprint goal. This is the self-inspecting part of Scrum (McKenna, 2016; Schwaber & Sutherland, 2016). Even though Scrum has this self-inspecting nature, the challenges that arises in GSD, e.g. time difference, is not something that a collocated Scrum Team encounters. For Indpro, these issues appear regularly between different Scrum Teams working on the same Product Backlog. Thus during these regular iterative meetings, more meticulous planning is required to not risk the time difference to cause unnecessary delays and dependencies between the teams, during for example a review of written code. This is something that is not included in Hossain et al.’s (2011) framework under *Reduced opportunities for synchronous communication* (CA1), the importance of planning one’s work to be able to communicate without leading to delays and dependencies. *Synchronized work hours* (GSD_P1) is a mitigation strategy that reduces such issues, according to the literature as well as the empirical findings from Indpro, though it is a method that demands more sacrifices from employees to work uncomfortable hours, as well as being more costly for the company e.g. due expensive inconvenient hour add-ons. The empirical findings from the case study suggests that planning your work, and when to communicate can even be beneficial in regards to proactivity, when having longer active hours of developers working and solving imminent issues. *Planning* (GSD_P8) could be another solution to mitigate these types of issues according to the empirical findings.

4.3.3 Face-to-face meetings difficult (CA2)

As is the nature when working distributed Scrum, there was a synchronous Daily Scrum meeting everyday between Sweden and India. The two Swedish software developers that was temporarily visiting India was still working with their ordinary Scrum Teams back in Sweden. This meant that the experience of working distributed on a regular basis was a new occurrence for them. Generally, one of them told the authors, software developers can often work distributed (e.g. from home instead of coming to the office) due to the nature of their work. Though, the experience of working distributed on a regular basis was a new occurrence for them. Noteworthy was the different degrees

of how actively they engaged in discussions in Daily Scrum meetings, Sprint Planning meetings and Retrospectives and Reviews that were held in the same room with people in the same time and space, compared to Scrum meetings via synchronous communication. Both tended to engage more in discussions in the meetings held in the same room in Bangalore than the meetings via video to Sweden. When asked about this occurrence, one of them said:

“I talk a lot more when I’m actually in the same room, I talk less over Hangouts (a video conference tool)”

People talking less over telecommunication was also something that management had noted, but rather regarded it as something positive in terms of the meeting culture and making meetings more productive. One C-level manager said:

“There are two sides of the story really, on the one hand you skip a lot of unnecessary information, it becomes more effective and you tend to follow your agenda for the meeting. On the other hand, it is harder to really get a grasp of the big picture, maybe”

During the authors observation, a common view among Indian developers was the ineffective meetings that Swedes generally have. One said during the interviews:

“Swedes generally have long meetings and have really good discussions where everybody really can express their opinion. You get opportunities to really discuss things. But sometimes I get the feeling there is a lack of conclusions and effectiveness.”

Moreover, during the different Scrum meetings there was different communication technology used. Company X had a conference room that was always on standby with a microphone and camera covering the entire room. At the office in Bangalore telecommunications was of course also being used to connect to distributed teams. Sometimes there were problems hearing and understanding one another. Both Indians and Swedes claimed that it is harder to understand each other over video communication rather than speaking face-to-face.

The lack of face-to-face meetings sometimes came with minor challenges of understanding each other compared to the collocated meetings being held. This was generally due to sound and video

quality, and the lack of perception of body language. When both the Swedish and Indian developers were asked about this they claimed that this was being mitigated by modern communication technology as this developer expressed:

“Yes obviously if a person is present at that location. It is easy to get your query resolved. But, I feel that [where] technology has taken [us], has reduced that hurdle, now we have screen sharing tools [and] we can easily ask if a person is free. We can just share the screen and get that resolved. But yeah, definitely being at the same geographical location has an impact.”

When the authors spoke to management about this, they had a vision that was brought from experience about how to use telecommunications in the future when working distributed.

“We always try to use the best technology available. And this is something we are always trying to improve. Imagine a conference room where the telecommunications automatically starts when you enter the room and that it can perceive every body gesture or facial expression. This is important because you need to tell how someone really feels about something, and it is something that can easily get lost when you don't see people's true reactions.”

Company X had a conference room that was always live and where it was tried to create an atmosphere for the distributed members of being in the same room when communicating over Google Hangouts. There was white board in the back of the room where people could write code or ideas on, though it could be hard to see what was written on the whiteboard due to low resolution.

4.3.4 Analysis of Face-to-face meetings difficult (CA2)

The empirical findings from the case study supports the Hossain et al. (2011) framework that *ICT-mediated synchronous, asynchronous communication* (GSD_P2 & GSD_P3) and *Visits* (GSD_P4) are suitable mitigation strategies to overcome “*Face-to-face meeting difficult (CA2)*”. There are aspects of using the ICT-mediated communication that affects the way Scrum methodology, and software development in general, in a way that what would not be present when working collocated that need to be highlighted. The first observation is the laconic, or rather compendious meetings when for instance having Daily Scrum meetings over Hangouts or Skype. The feeling amongst Indian software developers was that Swedes generally has unproductive meetings, but the management at Indpro had a belief that the ICT-mediated communication would make the meetings more efficient by reducing informal conversations in a meeting. To have fixed time meetings is

something that is inherent in Scrum according to Schwaber & Sutherland (2016), where e.g. a Daily Scrum is generally not longer than 15 minutes, and its purpose is to ensure collaboration among team members, to enable efficient work and to resolve problems (McKenna, 2016). In a way, the empirical support from the case study suggests that ICT-mediated synchronous communication ensures Scrum efficiency. It should also be stressed that this efficiency is not ensured without top quality technology. The observations at Indpro clearly show that efficiency is based on having great sound and video quality, to capture all parts of body language as well as a general understanding of what is being communicated. In summary, the framework provides appropriate mitigation mechanisms for this challenge, however, it seems to be rather important to follow Scrum methodology as well as providing communication technology which create a meeting set and setting that is as close to reality as possible, in order to be able to provide equally or more efficient teams as what is the case for collocated members using Scrum.

4.3.5 Cultural misunderstandings (CA3)

In the daily distributed work there could occur misunderstandings, or at least, difficulties understanding your counterpart at work. E.g. one day one developer for Company X was working on tasks regarding Swedish streets and area names, which can be hard if you have no understanding of that country or culture. That time, a Swedish developer was visiting the Bangalore office and the two could easily cooperate to solve this problem. After observing this occurrence along with other similar cultural misunderstandings, one developer was asked about such misunderstandings.

“One classic example is ‘buy two get one free’, and this happened very early [for him working with Swedes]. I was working on e-commerce and I had to implement that on the page. What I did do was $2+1=3$ that means if you buy 2 you get a third for free. So in the cart you will see 3 products, I thought like that, and later I got know it was incorrect. It should only be 2 products.”

This example visualizes that a shared understanding can be challenging working between two different cultures. To mitigate cultural differences, visits was once again something Indpro used to develop the relationships between Scrum Teams, as well as the developers and the companies. One developer at Company X commenting:

“Yes it helps really much [visiting Sweden]. Not only in the cultural way but also from a professional view, you get know technical details and get help easy and get to know in which direction the company is moving. And also things like when something is moving

from one hosting service to another. The teams [are] usually working in different technologies and being in Sweden you can see what they use [is], and discuss technical things with the other teams.”

Misunderstandings were more common before, according to the more senior developers. Thanks to video technologies such as Hangouts and Skype, as well as text messaging software Slack, the general picture was that misunderstandings are not as common as they were before.

“I think we see less of these issues [cultural misunderstandings] now than before. I think it is because of our long experience working with them [Swedes], our Swedish corporate culture along with regular visits and also becoming more accustomed to the culture. I think Scrum is good too, compared with waterfall. Because of the frequent meetings. In a way you have to talk often to get a shared understanding of the task we are working on.”

4.3.6 Analysis of Cultural misunderstandings (CA3)

Cultural misunderstandings are bound to happen when people from different cultures interact. To reduce cultural misunderstandings it is of importance to create an environment where the concerned parties have a shared foundation and understanding. By using the Scrum methodology, where one of the main pillars is Transparency, issues of cultural differences can be reduced. McKenna (2016) mentions that everything in Scrum has to be out in the open for everyone to see. To decrease the distance prevalent between the Indian culture which has a high distance power culture and the Swedish one (Ranjan Kumar & Sankaran 2007), it is critical that everything is overshared as McKenna(2016) coins it. To mitigate issues in this category, the framework from Hossain et al. (2012) proposes the use of *ICT-mediated asynchronous communication* (GSD_P3), *Visits* (GSD_P4) and *Frequent (or Improved) communication* (GSD_P5). The empirical evidence from this study conforms with these proposals, though the authors of this study would like to include the use of *ICT-mediated synchronous communication* (GSD_P2). By having dynamic, real-time communications the concerned parties can, if any misunderstandings occur directly address them by asking and confirming each other's statements. In addition to this extension the authors believe that the importance of *Visits* (GSD_P4) could not be stressed enough. The visits in each other's countries can help the developers learn and absorb the soft values and culture, and which will in extension help in the creation of a shared understanding. Furthermore the authors discovered during the observations that the matter of cultural differences was more prominent in Company X than Y, the authors believe this to be the consequence of the B2C nature of Company X, whereas Company Y exclusively work with B2B. By catering to the end-consumer, the Indian team of Company X has to have deeper knowledge about the culture, especially the softer cultural

values, for instance the “*buy two get one free*” issue that a developer had. In Company Y which only works with other businesses, and has a very technical orientation, these issues are not prevalent. The reason for this, according to the authors is that both the Indians and Swedes have the same technical education, i.e they speak the same technical lingo without the need to include softer cultural values to work more efficiently. As the mitigation strategies presented by Hossain et al. (2012) must be modified according to the specific context where it is being applied, the strategies might be used to different degrees relative to the need of the specific environment or context.

4.4 Coordination

4.4.1 Reduced informal contact can lead to lack of critical task awareness (CB2)

The lack of information about future tasks, product and strategy has implications on the communication between the teams. With the current structure the India-based team must communicate through the Product Owner when they need help. This becomes further problematic when the Product Owner is on leave or otherwise unavailable. A software developer in India comments on this challenge:

“And it’s a bit difficult to communicate if the Product Owner is on leave or something like that, even though we have other channels to communicate with, we don’t know who you should talk to about something”

Company X has provided the teams with different tools to communicate with each other but since the Indian team doesn’t know who to contact directly they can only send out an open query. A team member said:

“At the moment we can of course contact the other teams through slack but, as it is at the moment, we get replies a lot later, because they have their own things to focus on.”

A Company X software developer from Sweden further comments:

“There sometimes arise situations where a certain team willingly or unwillingly withholds information which will affect the other teams, so it makes it harder for the other teams, especially the Indian team to obtain that information.”

During the interviews it was disclosed from the Indian team members that they would like to contribute more to the company but that wish is hindered by the lack of knowledge about future work.

They feel that sometimes they only get to know about new developments after it has already been developed and therefore they cannot share any inputs. They think that more contact with the other teams would've been sufficient to prevent issues like this but they are aware of the complicated trade-off that management has to consider to not make it as time-consuming as it was before when there was only one team. This lack of informal contact, when the Indian team works from home soil also leads to the Indian team at Company X not feeling like a part of the client company, which they technically are not but they believe would help with their work with a software developer commenting:

“When you work in Sweden you will interact more, you feel that you are in the production area. Like the main area. Now here in India you can only watch through hangouts and [software] like that.”

Moreover the authors noted that instead of using the designated team names, the Indian teams for both case companies, used “Company X team” and “Company Y team” when conversing. Both the Indian Scrum Teams for Company X and Company Y had earlier been part of bigger teams with Swedish members, and was working fully distributed inside one Scrum team. When this study was conducted the Indian teams worked as separate Scrum Teams and was in some cases connected through Scrum-of-Scrums with their Swedish counterparts. This was to make the teams more efficient and less dependent on each other. There was still dependencies between the teams, since they worked from a shared product backlog for Company Y. One of the challenges arising from distributing work was knowledge sharing. To overcome this Indpro regularly sent developers from both Company X and Y between Sweden and India to visit each other to overcome challenges due to lack of face-to-face meetings, as well as overcoming other cultural and temporal barriers such as limited knowledge sharing. During the ethnographic study, the two Swedish developers visited Company X and two Indian developers visited Sweden shortly after their departure. Developers from Company X claimed that developers visited from one country to the other about every six months, generally. The developers really felt it was helpful to visit the other country, for cultural reasons and to get to know how different cultures work, as well as getting to know the people working on the same projects from the same product backlog, as well as to spreading and gathering knowledge.

“Initially when I had joined indpro, just after 20 days we went on a Sweden trip for one month. It was more of knowledge sharing and how people work in Sweden, It was more of knowing each other. That was a really good experience because those guys in Sweden

wanted us to just come there, feel how they work and feel the product on which we are going to work. And then, decide what part we had to take once we come back to India.”

Visits were clearly helpful according to the software developers both from Sweden and India. However, even when visiting each other regularly, they still often worked in divided teams between the countries, thus making it challenging to know the bigger picture or what other teams was working on.

Since they work with Scrum, which emphasize transparency, most Scrum Teams regularly have meetings one another. For instance they have bi-weekly demo meetings with the other teams to inform of their progress. But the Indian team would have liked to know about the other teams progress during the sprint and not after. They suggest that a team member should stay in contact with the other teams and a project manager for Company X mentions:

“It would be better if each team has someone with knowledge about other teams, who I could ask.”

The Indian team of Company X mentions that this lack of knowledge lowers the efficiency of the group, and when asked about efficiency a manager at Indpro mentioned:

“When you don’t follow the Scrum Methodology there will be implications on the efficiency”

The manager mentions that by following the Scrum Framework efficiency, planning and transparency will not be issues. The reason for not completely following the Scrum Framework, is that the customers organizations are not usually suited or aligned to fully use Scrum.

4.4.2 Analysis of Reduced informal contact can lead to lack of critical task awareness (CB2)

One observation was the importance of following the Scrum methodology to the letter. The Hossain et al (2011) frameworks mitigation strategies GSD_P5 - GSD_P8 are mitigation strategies that are directly related to the use of Scrum practice. These are *Frequent Communication, Iteration, Review* and *Planning* that is inherent in Scrum. The examples above show that issues can arise regarding knowledge sharing when working distributed, but also when working with software development using Scrum. One of the main pillars of Scrum methodology is Transparency, i.e. that everything shall be visible to the entire team and that there shall be a shared understanding within

the development team (Schwaber & Sutherland, 2016). The software developers said that they wished to be more included and to know what was going on in the bigger picture as well as within different teams. One solution for this is to work with Scrum-of-Scrums to a greater extent, which is an inherent part of Scrum. Management at Indpro said that it is challenging to convince customers, or rather, make sure that customers fully adopt their way of work to full-fledged Scrum methodology. Hossain et al. (2011) mentions that there are plenty of different contexts where GSD is prevalent, so using Scrum as well as other mitigation strategies differ based on the specific context. This means that there is no ideal theory for all contexts. With this in mind, and at the same time observing Indpro's way of managing GSD challenges, it becomes clear that the main mitigation strategy not inherent to Scrum to overcome *Reduced informal contact can lead to lack of critical task awareness* (CB2) is to focus on more frequent visits between the sites to create a common culture and understanding. For the inherent parts of Scrum there seemed to be a better common understanding and transparency in Company Y, where they had Scrum of Scrum meetings, than what was the case for Company X where they did not have such meetings.

4.5 Control

4.5.1 Difficult to convey vision and strategy (CC2)

For Company X, one issue when only having one big Scrum Team was that it was difficult to coordinate the team and give each team member meaningful tasks while simultaneously arranging common meetings and other communications became very time-consuming. The Scrum meetings and other common meetings could sometimes take longer time than the recommended amount and the majority of the information shared during those meetings was redundant for the Indian team. Company X therefore chose to split and divide the teams based on function, which led to increased productivity and efficiency. The division of labour has however presented issues regarding strategy and vision on an operational level. A Software developer said:

"...you don't get know what the other teams are working on right now. Before, when in one big team everybody knew."

Another software developer comments:

" Now I don't know anything, I'm clueless about the other teams. About what things are going on over there. Before I knew what kind of work was coming up in the future, like if they changed something but now I get know it after they have developed some of the main things."

The management of Indpro was well aware of these types of challenges, and the feeling of inclusion differed depending on which client the developers worked with. The developers worked with customers in a dedicated fashion i.e. working almost as full employees to almost all extents, except being formally employed by their clients. This, among other things, was a way to include them in the customer's vision and strategy. A C-level manager commented on this aspect.

“The responsibility is mainly on the clients [to convey their vision and strategy to the developers]. For example, all discussions on Company Y are very open, it is quite a small company where everybody can get a feeling of what is happening [in the organization]... Whereas Company X is bigger and has grown, there it naturally arises bigger challenges considering there are more teams working on smaller parts for a bigger holistic picture.”

A difference between Company X and Company Y was the use of Scrum-of-Scrums, which the latter used between the Indian team and the Swedish counterpart, and was something that Company X did not use. Although the developers in both teams expressed wishes to be a bigger part of the strategy in decisions, the developers from Company X were the ones that expressed greater feelings of isolation from other teams. The main differences between the two Companies was firstly, Company Y being a smaller company with fewer teams. Secondly, Company Y was using Scrum-of-Scrums between the teams, and Company X was not using that part of Scrum methodology at the time of the observation.

4.4.2 Analysis of Difficult to convey vision and strategy (CC2)

The empirical evidence from Indpro suggests that it is challenging to convey one's vision and strategy all the way down in the development chain. The framework suggests that *Visits* (GSD_P4), *Frequent (or Improved) communication* (GSD_P5), *Review* (GSD_P7) and *Planning* (GSD_P8) are commonly used mitigation strategies for this issue. Inherent in Scrum methodology is the use of Scrum-of-Scrum meetings, which were used for Company Y, but not Company X and where the developers in the former did not experience the same level of lack of exposure to vision and strategy. Important to note though, is that the company that was larger in terms of developers as well as number of Scrum teams experienced more challenges in terms of conveying vision and strategy. But it was evident when talking to Company Y about what the Scrum-of-Scrum meetings had brought in terms of transparency and awareness of other teams as well as a bigger picture of what was going on in the company in general, compared to how it was before those meetings were implemented. Transparency is inherent in Scrum, or rather, one of the main pillars of what Scrum methodology brings when used right (McKenna, 2016). So using Scrum to the fullest, can make sure that transparency will be prevalent to a larger extent, for instance through using Scrum-of-

Scrum meetings. The recently mentioned mitigation mechanisms that the Hossain et al. framework (2011) brings up are of importance to convey the strategy and vision from e.g. Sweden to India. The most prevalent of these mentioned in the framework that was observed at Indpro is arguably *Visits* (GSD_P4). As visits only takes one so far, it seems to be of importance to always make sure transparency exists between clients (Company X & Company Y) and vendor (Indpro) as well as between teams, for instance through Scrum-of-Scrums, when arranging visits but also in everyday work. In other words, using Scrum-of-Scrums may ensure greater transparency and may bring the mitigating effects that for an example *Frequent (or Improved) communication* (GSD_P5), *Review* (GSD_P7) & *Planning* (GSD_P8) gives to GSD according to Hossain et al (2011)'s framework. This stresses the importance of using all inherent parts of Scrum methodology i.e. Scrum-of-Scrum in this particular case for Company X.

4.5.3 Perceived threat from training low cost “rivals (CC3)

During the interview with the main offshore coordinator (MOC) at Indpro in India, it transpired that one of the main issues arising from working with clients from another country was inactivity from the clients. Clients would sometimes not answer urgent and operational queries from the developers in India which could leave them stranded for a couple of days without the possibility to proceed. Furthermore the MOC mentioned that he sometimes got the feeling that certain members from the client company intentionally went inactive due to feelings of insecurity and perceived threats from the developer in India. MOC:

“... when Indian developers are trying to communicate with Swedish developers the Swedes aren't cooperating sometimes, intentionally. So the Swedes are feeling the competition and insecurity.”

When asked about this, a manager at Indpro commentated that it is a prevalent issue and of which they are aware of and thus they therefore prefer to work with expanding companies instead of downsizing companies. The manager states:

“To avoid this we are trying to only work with growing companies but we realize that as a small company we don't have the luxury of rejecting clients that easily, though we try to emphasize our competence instead of the cost reduction they will experience ”

Another manager mentions:

“I think that is quite a normal reaction when their employers contracts an outsourcing partner. In my opinion the main responsibility reside with the client company’s management to inform their employees of the nature of the cooperation, that they, we in this case, are there to supplement rather than replace them and their work.”

4.5.4 Analysis of Perceived threat from training low cost “rivals” (CC3)

In this challenge category the framework did not provide any strategies which might mitigate the issue of *Perceived threats from low cost rivals* (CC3). According to the managers the issue is prevalent and understandable from the Swedish developers in the client company. One manager mentions that the ultimate responsibility resides with the client. The manager believes that they should inform their employees that the Indian team from Indpro is not there to replace but to complement their work. Even though the framework did not provide any mitigation strategies, the authors realized that Indpro applied their own strategies to solve this issue. Firstly, they try to only conduct business with growing companies, so as to not take tasks from the previously established teams in Sweden. Secondly, Indpro’s insistence to place their developers in the same team as the Swedes in the startup phase of projects. While the authors believe the purpose of this insistence from Indpro’s side is mainly to facilitate knowledge sharing and bringing their developers up to speed, there is another implication of that strategy, which the authors realized during the observation period. By working in the same team as the Swedes and in combination with the mitigation strategies *Frequent (or Improved) communication* (GSD_P5) & *Visits* (GSD_P4) from Hossain et al.,(2011)’s framework, the Swedish developers at the client side perceived the Indians as team members rather than competition. By using Scrum, which assumes *Frequent (or Improved) communication* (GSD_P5) which they used in this case, it could be argued that Scrum in combination with Indpro insistence, and usage of regular visits, mitigates this challenge. These strategies helped the Indians become part of the ingroup instead of outgroup which they were in the beginning. At first as the MOC mentioned, sometimes the Swedish developers would deliberately not cooperate, and with the Indians being an outgroup, the Swedish team regarded them with animosity. However, through the strategies applied and with the help of the management at both case companies, Indians became part of the ingroup. From these actions, Indpro could seamlessly create their own separate Scrum team, as Kniberg (2015) advocates.

5. Theoretical & Practical Implications

This chapter begins by presenting the implications of this empirical study, both to Hossain et al. (2011)'s Framework (section 5.1) and practical (5.2). This is followed by Implications for GSD theory (section 5.3) and the Limitations of this study (5.4) and finally by proposals for Future Research (5.5).

5.1 Implications for the Hossain et al (2011) Framework

Below are the implications for the research framework that has been found in this empirical study. In bold are suggestions for added mechanisms. Underscored are mitigation strategies that have been pinpointed as extra prevalent or that practitioners should further examine, e.g. ICT-mediated synchronous communication, (GSD_P2) where the suggestion is to focus on ICT-communication tools that are state of the art to receive all the possible benefits.

5.1.1 Communication

Reduced opportunities for synchronous communication (CA1)

The Hossain et al. Framework suggests that *Synchronized work hours* (GSD_P1) is a viable mitigation mechanism, and the authors agree this mechanism can solve issues regarding synchronous communication. As the Indians usually started their working days later than their Swedish counterparts (in local time), this had an effect of automatically getting more synchronized working hours, therefore this study suggests a different mitigation mechanism to be more suitable. The findings from this study suggest that *Planning* (GSD_P8) should be prioritized instead of the current mechanism proposed by the framework. By planning more precisely, the time difference can be used to the Indian teams' advantage and issues regarding time difference can be effectively mitigated. *Planning* (GSD_P8) prior to this study has not been regarded as a mitigation strategy for the time difference issue in GSD and is therefore suggested to be introduced to the framework. While it must be noted that this should be explored further in future studies, it was a successful mitigation strategy from these two case studies at Indpro.

Face to face meeting difficult (CA2)

The empirical findings support the use of *ICT-mediated synchronous communication* (GSD_P2), *ICT-mediated asynchronous communication* (GSD_P3), *Visits* (GSD_P4) as mediation mechanisms. However the authors noticed during the study that the impact and the effectiveness which they provide is dependent on the ability of aforementioned technologies to capture sounds, expres-

sions and body language. The most important finding is thus not only to have ICT-mediated synchronous and asynchronous communication, it is also to ensure as high quality of the technology as possible.

Cultural misunderstandings (CA3)

The empirical evidence gathered in this study conforms with the proposed solutions in the Hossain et al. framework, which was *ICT-mediated asynchronous communication* (GSD_P3), *Visits* (GSD_P4) and *Frequent (or Improved) communication* (GSD_P5). By examining the evidence from this study the authors realized that another mechanism was used to overcome cultural misunderstandings. That mechanism was *ICT-mediated synchronous communication* (GSD_P2), and by using that mechanism people can when misunderstandings occur, address them directly. Beside that addition to the framework, the authors advocate a strong emphasis on *Visits* (GSD_P4) to help create a shared understanding.

5.1.2 Coordination

Reduced informal contact can lead to lack critical task awareness(CB2)

Hossain et al.(2011)'s framework proposes the use of *Visits* (GSD_P4) *Frequent (or Improved) Communication* (GSD_P5), *Iteration* (GSD_P6), *Review* (GSD_P7) and *Planning* (GSD_P8). The findings from this study concur with their recommendation. The authors of this study would like to stress the importance of *Visits* (GSD_P4) to mitigate this issue as well as making sure to follow the Scrum methodology more accurately to get the level of transparency needed to ensure task awareness. Mostly in Indpro's case it regarded the awareness of other Scrum Teams tasks rather than their own. By visiting often and using Scrum-of-Scrums this issue seemed to be mitigated to a higher extent than scenarios where visitations were more scarce.

5.1.3 Control

Difficult to convey vision & strategy (CC2)

The implications here for Hossain et al. (2011)'s framework from the empirical study indicates that the strategy used most in practice by Indpro was the use of *Visits* (GSD_P4). The framework recommends, beside *Visits* (GSD_P4), the use of *Frequent (or Improved) Communication* (GSD_P5), *Review* (GSD_P7) and *Planning* (GSD_P8) to mitigate this challenge. The authors agrees to the use of those strategies but would like to underscore *Frequent (or Improved) Communication* (GSD_P5) as the main contributor to overcome the challenge of conveying vision & strategy. For this case study it was prevalent that the Indian team members that had frequently visited

the onshore country, i.e., Sweden. After these visits the Indian team members gained a more comprehensive understanding of where the projects were headed. If they did not visited frequently, they felt that they knew less about the vision and strategy.

Perceived threat from low cost rivals (CC3)

The Hossain et al (2011) Framework did not provide any mitigation mechanisms for this category but through the study the authors did find certain mechanism which could be used to mitigate this challenge. These mechanisms are *Visits* (GSD_P4) and *Frequent (or Improved) Communication* (GSD_P5) which can help the offshore-team (Indian team) be considered team members instead of competition by the Swedish team. This is because this issue was not prevalent any longer after the teams met and got to know each other in the Indpro case, therefore it is important for the new developers to have the opportunity to meet the developers early in the process and that they continue to have frequent communication between the sites. This was to ensure a common understanding and a common goal and to avoid being seen as rivals.

In the framework below are suggestions in bold for added mechanisms. Underscored are mitigation strategies that have been pinpointed as extra prevalent or that practitioners should further examine

Challenge Description	Challenge Description	Mitigation Strategy
Communication (CA)	CA1. Reduced opportunities for synchronous communication	GSD_P1. Synchronous work hours, GSD_P3. ICT-mediated asynchronous communication GSD_P8. Planning
	CA2. Face-to-face meetings difficult	<u>GSD_P2. ICT-mediated synchronous communication,</u> <u>GSD_P3. ICT-mediated asynchronous communication,</u> GSD_P4. Visit.

	CA3. Cultural misunderstandings	GSD_P2. ICT-mediated synchronous communication GSD_P3. ICT-mediated asynchronous communication. <u>GSD_P4. Visit.</u> GSD_P5. Frequent (or Improved) communication.
Coordination (CB)	CB2. Reduced informal contact can lead to lack of critical task awareness	GSD_P2. ICT-mediated synchronous communication GSD_P3. ICT-mediated asynchronous communication <u>GSD_P4. Visit.</u> <u>GSD_P5. Frequent (or Improved) communication</u> GSD_P6. Iteration GSD_P7. Review GSD_P8. Planning
Control (CC)	CC2. Difficult to convey vision and strategy	<u>GSD_P4. Visit.</u> <u>GSD_P5. Frequent (or Improved) communication</u> GSD_P7. Review GSD_P8. Planning
	CC3. Perceived threat from training low cost "rivals"	GSD_P4 Visits GSD_P5 Frequent (or Improved) Communication

5.2 Practical Implications

Besides the theoretical implications of this study, the authors have found several practical implications. One primary implication is that the vendor should adhere further to the Scrum Methodology and request that their client do the same. By doing this, both client and vendor can experience a more productive and efficient collaboration.

Due to the distributed nature of the collaboration, having synchronous meetings and face-to-face meetings is difficult, meaning that Indpro must apply other tactics to overcome these potential problems. The evidence gathered suggests that it is fundamental that high quality ICT systems is used and that every team, in Sweden and India is encouraged to plan their tasks more efficiently. To deal with issues regarding cultural differences, one of the main pillars of Scrum, Transparency, might mitigate some of these by providing shared values and a common understanding, but it cannot solve all. Frequent visits between the vendor and client teams is imperative so that the team members can get to know each other and the different cultures prevalent. During these visits the different teams should also work on the same tasks to facilitate knowledge sharing between the two groups. The study also indicates that the level of cultural misunderstandings depend on which type of business the client conducts. Company X, which caters to the end consumer (B2C) tended to experience more cultural misunderstandings with the Indian team than Company Y which solely conducted business with other companies (B2B). The number of visits needed is therefore based on the specific context.

To ease the difficulty of conveying the client's vision and strategy and the potential lack of task awareness, the authors suggest stricter adherence to the Scrum Framework to introduce Scrum-of-Scrums in Company X where these issues were more prevalent. To counter the threat that the Swedish developers may feel when an Indian team joins, Indpro can work to clarify to the client that it is of importance to work together to make sure that the employees are informed of the nature of the collaboration to ensure its efficiency and productiveness when starting a new project with external consultants. Furthermore Indpro should continue to insist that their employees be part of the Swedish Scrum teams at the beginning of the project to facilitate knowledge sharing and create a feeling of “teamness” which might ease the Swedish developers animosity towards their Indian counterparts. After this has been achieved, the Indian members should be transferred to a new, geographically separated team in India, which can work with limited dependencies on the Swedish teams.

5.3 Implications for GSD Theory

Bannerman, Hossain & Jeffery (2012) studied, in their paper *Scrum Practice Mitigation of Global Software Development Coordination Challenges*, the implications of the Hossain et al. (2011) Framework empirically and their findings are similar to what was found in this study with a few exceptions. For example in *Reduced informal contact can lead to critical task awareness (CB2)*, their findings suggested that the inherent parts of Scrum; fostering frequent communication, iteration, review and planning, significantly contributed to mitigating this challenge. Noordeloos et

al. (2012) also suggest inherent mitigating characteristics from Daily Scrums, with its iterative nature, are of help to ease such challenges. The findings from this study was rather, to ensure the full use of Scrum to witness the full benefits of mitigating this GSD challenge among others, i.e. in for Company X where Scrum was not fully applied the mitigating effects on GSD debilitated. The general findings from this study, in regard to ICT mediated synchronous and asynchronous communications tools contrasts with Bannerman et al. (2012) findings. They argue for the importance and benefits of these type of communication tools, which the authors of this study do not oppose, but has rather argued for the quality of such tools. A possible reason for this difference can be that this thesis is written around five years later after a vast development of such tools, where they both are taken more for granted in organizations. The implications for the GSD theory in general is arguably to focus on the quality in sound, video and experience when working with ICT mediating technology, rather than to argue to have it at all. The findings from this study must stress the importance of high quality communication technology. Bannerman et al. (2012), argues in their empirical study, just as the Hossain et al (2011) framework suggests, that the mitigation strategies for the time difference is *Synchronized work hours* (GSD_P1). This study supports this notion, but with addition of *Planning* (GSD_P8) as a mitigating strategy. It shall be stressed that the generalizability of this finding must be studied further (see more in section 5.5 Future research) to ensure that this is valid for more cases as well as cases where the time difference is larger than 3.5-4.5 hours.

5.4 Limitations

This contribution has some limitations that need to be taken into consideration. Firstly, and as Hossain et al. argues, the framework has its limitations from being based on mostly industry experience reports. Hence, the practical implications suggested in this study are based on such material. This thesis provides empirical backing for some of the challenges in GSD as well as some mitigation strategies. Important to note is that due to limited amount of time and resources, the entirety of the framework could not be covered. Furthermore, there were only two cases that could be ethnographically observed for a time period of six weeks. On a similar note this research has been of a somewhat inductive nature which besets the generalizability of the findings to other contexts and for other companies. Rather, this study provides support for the framework of providing empirical support for a framework primarily based on experience reports. Another limitation of the framework and as Hossain et al. (2011) argues that it assumes a generic GSD-context and it may obscure project-specific variations in the mitigation mechanisms. In addition, the time difference for this study was 3.5-4.5 hours. In settings where the time difference is larger, it is likely to

affect the outcome of observing and studying such challenges. Another example of different settings that may affect the study is the degree of distribution of the Scrum Teams. For instance, there could be settings where a company only works with fully distributed members inside a Scrum Team, to another context where only one member or one team is distributed. These are examples of settings that is probable to change the outcome of a study like this. Thus it is important to note that there is no best practice to use this framework or a generic approach of dealing with GSD challenges. These are only general proposals of generally observed challenges and suitable mitigation mechanisms for such issues.

Other limitations include the fact that the authors were invited by management by Indpro and this could affect both the interviewees and observed subjects as well as the study as a whole by not acknowledging all challenges due to unintentionally or rather unconsciously missing important challenges that could have otherwise been identified.

5.5 Future research

Firstly, the authors would for future research like to suggest additional studies similar to this in other geographies and between different countries; these results could then be compared and culture-specific themes could be identified.

During this empirical study the authors realized that one fundamental issue when working in a distributed environment is the issue of knowledge sharing. Secondly, the authors would therefore like to suggest research in the field of knowledge sharing in a distributed environment.

Also, a quantitative study to validate the framework, which is currently based mostly on experience reports is needed. This to both validate the framework as well as this study's suggestions for improvements.

Lastly, a study testing the framework in a distributed environment where the time-difference is larger than in this study and where the whole framework is covered.

CONTACT US :-



+1 (201) 381 - 5152



contact@foreignerds.com



251 Little Falls Drive, Wilmington
DE, USA 19808

Thankyou

