Software Development across Time Zones
A Study of Globally Distributed Software Development in Small Enterprises

Uppsala University

Alexander Rangevik
Master program in Management, Communication and IT
2013

Supervisor: Prof. Pär Ågerfalk
Abstract

Globally distributed software development has become an important topic for small and medium enterprises. However, the distinct requirements of small and medium enterprises (SMEs) are still not fully understood. To contribute to this growing discussion, I will present an ethnographical field study of a Swedish software company engaged in a distributed development practice with its American branch. By applying a theoretical view based on communication, coordination and control over time, space and socio/cultural difference, I show to what extent a distributed software development model affects SMEs. Where problems related to formal and informal communication, knowledge sharing, articulation work and programming methodologies are discussed. My findings suggest that organizational adjustments are needed for SMEs in order to take advantage of distributed software development (DSD), that specialized tools supporting coordination and articulation work is not so common in practice and that certain adjustment to agile methodologies could be made in order to better support DSD.

Keywords: Globally Distributed Software Development, Small Medium Enterprises, Coordination, Communication, Control, Agile methodology
Acknowledgements

First I would like to thank the company that made this study possible, both for their priceless support and warm welcome to the American office. I also want to thank my dear classmates and American friends for their unfailing support during the writing process. Finally, I would like to thank my supervisor for his great advice.
# Contents

Contents ........................................................................................................................................... 5  

1  Introduction .................................................................................................................................. 7  
   1.1 An Era of Globalization and Distribution of Work......................................................... 7  
   1.2 Problem Discussion............................................................................................................. 8  
   1.3 Purpose .............................................................................................................................. 9  
   1.4 Research Question ............................................................................................................ 9  
   1.5 Disposition ....................................................................................................................... 10  

2  Literature review ......................................................................................................................... 11  
   2.1 Globally Distributed Software Development .................................................................. 11  
   2.2 Small and Medium Enterprises ....................................................................................... 12  
   2.3 Communication, Coordination and Control ..................................................................... 12  
       2.3.1 Communication ....................................................................................................... 13  
       2.3.2 Coordination .......................................................................................................... 14  
       2.3.3 Control .................................................................................................................... 15  

3  Research Method ......................................................................................................................... 16  
   3.1 An Ethnographic Study ...................................................................................................... 16  
       3.1.1 Data Sources ........................................................................................................... 16  
       3.1.2 The Case Company ................................................................................................. 17  
       3.1.3 My Role .................................................................................................................. 17  
   3.2 Theoretical Lens and How the Analysis was Made ......................................................... 18  

4  Ethnographic Study in Los Angeles ......................................................................................... 20  
   4.1 Company History and Current Situation ........................................................................ 20  
   4.2 Company Structure - four Departments ......................................................................... 21  
   4.3 Time Zones and Locations ............................................................................................. 21  
   4.4 An Agile Methodology ..................................................................................................... 22  

5  Findings and Analysis ............................................................................................................... 24  
   5.1 Communication ................................................................................................................. 24  
       5.1.1 Meetings and Time Zones ....................................................................................... 24  
       5.1.2 Sharing Knowledge and Lack of Informal Communication .................................. 26  
   5.2 Coordination ..................................................................................................................... 28  
       5.2.1 Modularization or Shift Work .................................................................................. 28  
       5.2.2 Lack of Team Awareness ......................................................................................... 30  

5
5.2.3 The Communication Funnel and Team Spirit ......................... 32
5.3 Control ................................................................................. 33
5.3.1 Managing Project Artifacts ............................................. 33
5.3.2 Allocation of Roles and Team Structure ......................... 35

6 Concluding Discussion ............................................................. 38
6.1 Communication, Coordination and Control ......................... 38
   6.1.1 Communication .............................................................. 39
   6.1.2 Coordination ................................................................. 40
   6.1.3 Control ........................................................................ 41
6.2 A prospect of future research ............................................. 42

7 References ............................................................................... 43
1 Introduction

This chapter serves as an introduction to the field of distributed software development and the main problems within the practice. Then, I will move on to discussing my personal view of the subject, highlighting the particular research question that I will investigate and how my research will contribute to the field of study.

1.1 An Era of Globalization and Distribution of Work

Over the last several decades, economic forces have turned national markets into global ones. The trend toward globalization of business, and software developing in particular, has created new forms of competition and cooperation that reaches across national boundaries. Tight budgets, time constraints and limited workforce benefit from exploring the global marketplace (Šmite, Moe & Ågerfalk, 2010). Obtaining an increased pool of knowledge by recruiting the most talented people around the globe, reducing time-to-market and reaching local clients are some benefits of going beyond the national borders (Carmel & Agarwal, 2001). This change has a significant impact not only on distribution and marketing of products, but also on the way products are constructed, tested and delivered to customers (Herbsleb & Moitra, 2001).

Software development, in contrast to other engineering disciplines, is recognized as a highly complex task that heavily relies on human interaction. Distributed software development (DSD) across temporally, geographically and socio-culturally dispersed teams creates a unique and challenging situation with major problems related to communication, coordination and collaboration. As a result, engineers, managers and executives are facing a growing number of challenges on many levels; from technical to social to cultural.
There is strong evidence, based on both statistical modeling of development and survey results, that distributed software development activities take much longer time than comparable co-located activities, and that coordination and communication are major factors in this delay (Herbsleb & Moitra, 2001).

As stated above, working with distributed software development with geographic dispersion creates obvious overhead. However, the potential advantage of widely dispersed time zones could theoretically extend the productive hours of the day from the current eight or so hours to somewhere near the limit of 24 (Herbsleb & Moitra, 2001), which in turn could speed up the development process three times. This may be a distant goal, but even the possibility of accelerated problem solving is worth investigation for small and medium enterprises (SMEs). That allows dispersed teams to share advances or communicate roadblocks for a problem, to finally conclude with an integration of modules into a final solution.

1.2 Problem Discussion

It is no longer unusual for large software projects to have teams in multiple locations (Herbsleb, 2007). On the other hand, it is less common for small enterprises (max 50-60 employees) to function in a collaborated manner when it comes to software development between their offices located in two or more countries. In the cases of small enterprises that use a distributed software development model, we often see Western European countries outsourcing to Eastern European ones, due to a high level of technical competence and lower salaries. (Richardson, Ita, et al., 2008)

The question is if there are other situations where small enterprises could benefit from a DSD model even if the setting is not as the one described above? The proximity to a certain market may be the main driver to initiate a DSD model for small enterprises, for instance between Europe and America. But could this increase in work hours gained from the time difference act as a secondary and positive outcome? For example with non-parallel tasks, where programming ac-
tivities are done in one time zone, there could be testing of that code in the next time zone and so forth. The flexible nature of small enterprises may have benefits when it comes to the organization of teams due to their more informal structure and communication channels (Richardson, Ita, et al., 2008).

The trend of globalization in combination with DSD portray that there are benefits to this kind of distributed work, especially for larger organizations. However, is the DSD model suitable for all organizations and sizes? Are factors such as role distribution, domain knowledge, time and geographical differences too complex for small organizations to deal with in order to fully take advantage of a DSD setting?

1.3 Purpose
The distinct requirements of SMEs in DSD settings are still not well understood (Boden, Bernhard & Volker, 2007). Thus, they will be the focus of my research, and more specifically, in this case study, newly started branches. For instance, in this case, startups without prior routines and experience of DSD. The more opportunistic nature of SMEs put them in different contexts than large companies, with perhaps more sophisticated outsourcing models and established practices. With this backdrop, what are some of the opportunities and problems with globally distributed software development for small enterprises without prior experience of DSD? Is the previous research done within the field of DSD, for larger enterprises, applicable even in the setting of SMEs?

1.4 Research Question
The goal of this study is to gain an understanding of the problems and opportunities that occur in distributed software projects in small enterprises; this in order to improve the outcome of DSD processes for newly started SMEs. This thesis will focus on those features of software projects that influence the need to coordinate, communicate and
maintain project control, which includes the investigation of communication mediums, coordination tools and processes. With this in mind, the research question is: *How are small enterprises’ processes in communication, coordination and control affected by distances in time, room and culture?*

1.5 Disposition

The first chapter serves as an introduction to the field of distributed software development and the main problems within the practice. The second chapter, the literature review, aims to provide an understanding of the major research findings in globally DSD projects, by examining work conducted by previous researchers. The third chapter, research method, will motivate to my choice of research method and its particular benefits for this type of study. The forth chapter will give an introduction to the company of study. The fifth chapter, empirical findings and analysis, will provide an insight of the challenges related to DSD that I found during the study. The sixth and final chapter will conclude the key findings of my work and give a prospect of future research.
2 Literature review

As stated before, the distributed software development field in combination with an agile approach is an area of research that has not yet been widely investigated. The practice is still ahead of the research, and understanding of the phenomenon of DSD in various settings is still an ongoing process among researchers.

This chapter aims to provide an understanding of the major research findings in globally DSD projects, by examining work conducted by previous researchers. Work that states that characteristics such as communication, coordination, overall project control, knowledge sharing and team awareness are, among others, contributing factors to successful DSD.

2.1 Globally Distributed Software Development

The way I choose to describe DSD in this thesis is in the more broad sense. That includes not only the “pure” development activities but also tasks that surround it - from idea, to pre-studies and requirements specification to deployment and maintenance. In other terms, this definition includes most of the development life cycle.

The major problem that has been identified within the globally DSD literature is distance, which introduces barriers and complexity into the management of DSD projects (Carmel & Agarwal, 2001; Damian & Zowghi, 2003; Herbsleb & Mockus, 2003).

Distance can be recognized in the context of three dimensions, temporal, geographical and socio- and cultural. Temporal distance introduces hinders for direct communication and cooperation (Ågerfalk & Fitzgerald, 2006). Geographical distance limits the physical proximity between team members (Carmel, 1999). Finally, socio- and cultural distance impacts the level of understanding between remote teams, as well as their activities and efforts (Rutkowski et al, 2002). Distance in
all three dimensions impacts the way teams communicate, coordinate and structure work activities (i.e. who is doing what, where and when).

2.2 Small and Medium Enterprises

In this globalized world economy, SMEs are now entering the DSD field, even though their choices often have to do with opportunities that present themselves rather than careful planning (Richardson, Ita, et al, 2008).

There are several characteristics that are different when it comes to the activities in small companies compared to their large counterpart. The hierarchy is typically more flat. Because of a closer and smaller work environment, the employees know each other to a higher degree, even if they work at different locations and possibly never met face-to-face. (Richardson, Ita, et al., 2008) The projects are usually small, involving fewer people and having a lower level of complexity. The internal communication can be informal, even between managers and employees. Ongoing communication and reduced personnel turnover support trust being built over time. Employees tend to develop a shared understanding, and cooperation among these individuals is common, especially if they have established personal bonds and know each other well. (Kelly & Noonan, 2008)

2.3 Communication, Coordination and Control

In order to investigate the impact of a Global DSD setting in my study, the theoretical view will focus on the three processes communication, coordination and control, which have been well covered in the literature (Battin, 2001; Boland & Fitzgerald, 2004; Espinosa & Carmel, 2003; Ghosh, Yates & Orlikowski, 2004). These key processes will serve as categorizers for the major findings in the field and act as a theoretical frame for my research.
2.3.1 Communication

Communication is essential in all software development (Curtis, Krasner & Iscoe, 1988; Beck, 1999) and especially crucial in a distributed environment, where the preferred face-to-face communication is limited. In this type of environment, the medium for communication is shifted from face-to-face into a technical setting, and the communication is therefore more restricted. (Ågerfalk, 2004) During the software development process, the team members exchange a large amount of information and knowledge. The communication process in DSD is therefore heavily reliant on the tools used to enable interaction between actors.

When the members are located in different countries, other problems arise such as misunderstandings and misinterpretation. This is due to different mother tongues and cultural differences. (Jiménez, Miguel, Vizcaíno, & Piattini, 2010) Both national and organizational cultures play important roles when it comes to the understanding of communication related issues. National culture covers an ethnic group’s values, norms, language and style of communication, while organizational culture covers the working unit’s norms and values, which includes the culture of systems development (Carmel & Agarwal, 2001).

Due to the geographical distance between team members, there are limited situations for informal, spontaneous conversation. Informal “corridor talk” helps people stay updated on what is going on around them, what other people are working on, what expertise people possess, what states the various parts of the projects are in, and many other essential information that enable developers to work together efficiently. There are also risks of missing reuse-opportunities of code or solutions that otherwise could be spread through informal conversation between colleagues, which could save cost and time. (Herbsleb & Moitra, 2001) With geographic and temporal distance, vital informal communication becomes less frequent and poses obstacles for efficient coordination and collaboration (Herbsleb, Paulish, & Bass, 2005).
2.3.2 Coordination

The SME setting often implicates working in small teams that allows for more agile methods of dealing with coordination activities and articulation of work (Boden, Bernhard & Volker, 2007).

By coordination I mean the management of dependencies among tasks. If tasks that were carried out at different locations shared no dependencies, global projects would not pose a significant challenge. People at given sites would not need to communicate. Coordination is therefore the act of integrating each task between organizational units towards a common objective (Carmel & Agarwal, 2001). Software development obviously requires coordination, and the distributed setting emphasizes this since the introduction of time, space and cultural dimensions (Ågerfalk et al., 2005).

Coordination is one of the main time consumers in DSD (Herbsleb, 2001), whereby reduced coordination is advantageous. Reduced coordination costs could be an outcome of teams not working at the same time (Espinosa & Carmel 2004). Although, it is fair to assume that the side effects of reduced coordination suffers from increased time to fix misunderstandings, reworking and alignment of work etc. (Espinosa & Carmel, 2004) However, at the same time, coordination costs could be reduced through shift work. There is an increased effort to conduct synchronous team meetings. Team members would have to work flexible hours in order to coordinate with their co-workers through real-time conversations, thus increasing the cost and effort of coordination. (Battin, Robert, et al., 2001)

A distributed work setting lead to physical separation, lack of informal contact and a decreased feeling of “teamness” with remote colleagues (Battin, Robert, et al., 2001; Herbsleb, & Mockus, 2003; Kiel, 2003). Due to the lack of face-to-face contact between colleagues, team members may not be as aware of their remote co-workers’ work activities. This could lead to misunderstandings, vital information going unnoticed and code conflicts arising. (Ågerfalk, et al., 2005)
2.3.3 Control

The final process that wraps around the two aforementioned communication and collaboration is the aspect of overall project management and control (Ågerfalk, et al., 2005). Control is constituted by the management of goals, policies, standards, or quality levels regarding the development process as a whole (Carmel & Agarwal, 2001). In other words; the management and reporting activities that make sure development activity is processing and the formalized structure that ensures development of software on budget, in time and of required quality (Ågerfalk, et al., 2005).

Control is also closely related to management of project artifacts and therefore the practice of development methodologies. Project artifacts in this context relate to parts of projects that could be divided into more or less independent parts. Therefore, in order to minimize coordination costs, remote teams could split their work across feature content into independent modules (Ebert & De Neve, 2001; Sahay, 2003; Bass & Paulish, 2004). This allows less dependence on remote teams in decision-making processes and reduces problems in the integration phase of the systems (Herbsleb & Grinter 1999). Lings et al. (2007) suggest that, for DSD, ensure that the architecture of the system is consistent with the distributed structure of the team. This leads to the question of articulation work; individuals must somehow partition work into units, thereafter divide it amongst them and finally reintegrate it when the work is completed (Nardi, Whittaker, Bradner, 2000), henceforth called modularization work (Conchuir, et al., 2009), this is however also a coordination aspect.
3 Research Method

In this chapter I will motivate to my choice of research method and its particular benefits for this type of study. I will give an insight on how the study was conducted and the special circumstances of it. I will also present the theoretical lens through which I considered my findings.

3.1 An Ethnographic Study

Given that some research on information systems has focused on the social and organizational contexts of this field (Ngwenyama et al., 1999), ethnographic research emerged as one important means of studying these contexts (Myers, 1997a; Prasad, 1997). Ethnographic research is one of the most in-depth research methods available to aid the understanding of people, the organization and the broader context within they act. The context in this case, globally DSD, is deeply connected to human behavior, where the key aspects are communication, cooperation, trust, responsibility and articulation work. The chosen method is, therefore, more appropriate for this context than, for instance, interviews or surveys alone. The latter methods possibly create more manufactured data than ethnography do (Silverman, 2010).

3.1.1 Data Sources

Observations were mainly carried out at an off-site location in America, but several visits were also made to the head office in Sweden, where additional interviews with employees were held. The reason the study was carried out at the off-site location was that I wanted to investigate the DSD phenomenon from an off-site perspective. This also serves the research question better since my findings deviate more from my expectations as a Swedish developer in a culturally different
environment, as America, compared to doing the research from the Swedish site solely.

During the whole study, field notes (a) were written on a daily basis in order to preserve the true nature of my observation - I wanted the material to speak for itself as much as possible. However, despite the advantages of being present physically at the office, there were areas that were harder to cover in my observations, for instance when computers were used for communication, e.g. Skype or chat. These areas were complemented by interviews (b), and as Koch (2009) and Verbeek (2005) say that the technology becomes a stepping between the researcher and the experience, and could therefore benefit from further investigation. Lastly, studies of different technical solutions (c), such as online tools for collaboration and coordination, were performed and documented throughout the study.

3.1.2 The Case Company
The ethnographic study was carried out during three months at an American branch of a Swedish web development company, located in Los Angeles. The company had recently started to investigate the possibilities of a DSD model between the American and the main office in Sweden.

The American office was situated in a two-story house where I was accommodated on the second floor, while the office was located on the first floor. Living at the “office” resulted in an in-depth relationship with the environment and the people working there. Being the first and last person in the office every day made the observations comprehensive in regard to hours spent with the employees. This also corresponded well with my choice of research methodology and research question.

3.1.3 My Role
During the study I had an active role (as opposed to a strictly observing one) as the technical project manager of the American office. As for technology matters, I was the main communicative link between the offices as well as to the American clients. However, as Czarniaw-
ska (2007) points out, that an observer can see different things than actors, I had to be aware of the limitations of not perceiving the contexts from an observing point of view. Therefore, the data constructed from my personal experience was often discussed with my co-workers in order to confirm what I was experiencing, but also to see how it deviated towards their understanding of the situations.

3.2 Theoretical Lens and How the Analysis was Made

The ethnographic study was initiated with an explorative view upon the research field of DSD.

As my research progressed the data I collected from recurring activities were analyzed by trying to see patterns (Miles & Huberman, 1994) and why they emerged. These patterns then facilitated the forming of categories, which were analyzed in relation to the literature with a focus on coordination, cooperation, communication, articulation work, and project management during DSD.

This led me further into an investigation upon the literature in the field and its key conclusions. The findings by some of the major authors in the area (Battin, 2001; Boland & Fitzgerald, 2004; Espinosa & Carmel, 2003; Ghosh, Yates & Orlikowski, 2004) have concluded that DSD could be oriented through three main processes. As mentioned in the previous chapter these are Communication, Coordination and Communication.

These processes were then summarized further into “A Framework for Considering Opportunities and Threats in Distributed Software Development” (Ågerfalk et al., 2005). As Ågerfalk et al. (2005) state, “proven methods for successful DD have not yet been formulated, and the presented framework may be an important tool in identifying the most pressing research issues”. As the name indicates, the framework could be used to consider threats and opportunities in various DSD settings in respect to the three distance dimensions temporal, geographical and socio/cultural. This framework gave me a basis for the view upon DSD in large companies, and a way to investigate if the specific findings differed for SMEs.
When I then present the results, in my analysis of the empirical findings and conclusions, I want the reader to be aware of that the material is based on several voices. First it is my reproduction in which I try to synthesize what other persons say and do – based on interviews and observations, the empirical themes. The other one is citations from these interviews, i.e. these person’s subjective voices. The third is my subjective voice through which I conclude the discussions related to the empirical findings in relation to the literature in the field.
4 Ethnographic Study in Los Angeles

This chapter will give an introduction to the company of study, its structure, unique environment and work methodologies.

4.1 Company History and Current Situation

The company of study is Swedish based and operates in the web and film production industry with world recognized brands and clients. In 2011, a merger was made with a film production company based in Los Angeles, United States. The merger was mainly based on reasons related to proximity to the American clients the Swedish office already had.

The merger with the American company was primarily to create a closer communication channel with the American clients. This could also be seen as a way of taking advantage of the time zone difference in order to have the possibility of 16-hour workdays and to deliver projects with short deadlines and fast feedback. However, the routines and processes between the Swedish and the American office are in the early stages of a fully functional DSD model, and part of why this study was made.

The film department was already established in the Los Angeles office before the merger, and the web department was created as a result of it. The employees at the American site were not used to the processes related to software development, so as the office grew bigger the need to establish a foundation of work routines emerged. As a result, this study was initiated to understand the needs of the organization and to further develop its routines for DSD.
4.2 Company Structure - four Departments

The company in Sweden is constituted by four departments; Creative, Film, Web and Hosting/Support. The American office has a Film and a Web department: The management of this web department is, however, located in Sweden.

The creative department’s primary objectives are to create strategies, bring forth ideas, design-concepts, wireframes and user experience designs. They are also the closest link to the client, and therefore hold workshops and establish target groups in close collaboration with the client.

The film department’s objectives are the pre-production and production phases; such as planning, stage-management, script writing, sound, photo.

The web department’s responsibilities are production and maintenance of web related projects. Activities within the department consist of two units: back-end development and front-end development. The back-end development is mainly focusing on PHP and .NET programming languages while the front-end development mainly focusing on HTML5, CSS and JavaScript. With incorporating HTML5 as a technical platform for both mobile and desktop solutions, the field of work is very broad and encompasses most things related to web development.

The hosting, support and technical maintenance part of the company is located both in Sweden and in Mexico. The Mexico location is advantageous from a performance point of view when it comes to the hosting of the American clients’ sites. It also allows for faster support and maintenance for the Los Angeles office.

4.3 Time Zones and Locations

The company is relatively small, consisting of around 50 people, where 40 of these are located in the Swedish office. As mentioned earlier, the purpose of the merger with the Los Angeles office was primarily to take advantage of the proximity to American clients for better communication and support for existing clients. However, the
temporal dispersion in this case could prove to be advantageous when it comes to the development activities performed, and is therefore worth investigating. Possible outcomes of successful DSD between the sites could be 16-hour workdays and increased problem investigation.

4.4 An Agile Methodology

Each project is handled in its own way depending on the type of project and how much resources the clients have to spend. In the cases where the clients are direct customers (end clients) of the company, an agile workflow is preferred. In these projects, a set of features is agreed upon between the company and the client and then there is another set of “extra” features that are added if there is more time when the initial features are done.

Each week there is a Scrum (Scrumallience, 2013) sprint meeting where the company goes through the number of work hours for each feature. Before these meetings it is possible for the client to change features from the agreed list, in consultation with the development team to make sure the time frame is not altered - unless the client agrees on adding more funds to the project.

Each day there is a short meeting to make sure that there are no roadblocks and that the work is progressing as planned. In case of roadblocks, the technical project manager tries to solve these issues so that the developers can continue working.

In smaller projects, there is usually a more traditional approach where certain requirements are specified into features. These features are then locked with a fixed price tag and produced until finished, without clients changing features during the development period. However, the internal workflow follows an agile approach where small self-organized and cross-functional teams of 5-7 people work with either the whole or parts of projects.

The web department is divided into two teams; one in Sweden and one in America. The Swedish office had a technical project manager and four developers on front- and back-end. The Swedish technical
project manager was usually the one that distributed work over to the American office. The American office had two developers; one that acted as technical project manager and communicative link with Sweden and back-end developer and one that is more focused on front-end development.
5 Findings and Analysis

In this chapter I will provide an insight of the challenges related to DSD that I found during the study. The material gathered is summarized into themes that elaborate on specific topics. Each topic is concluded with a commentary where I analyze and make conclusions. In the commentaries I describe certain ways of dealing with these challenges, in order for SMEs to better handle these specific situations.

5.1 Communication

5.1.1 Meetings and Time Zones

The meetings with the Swedish office usually took place during the mornings in Los Angeles, where only people from the web department were present. The morning meetings were set to start somewhere around 07:15 and 07:30 (PST).

Initial contact was made through Skype’s text messaging function to confirm the other people’s presence. The technical project manager of the Swedish office calls, using Skype’s video function, to check in with the Los Angeles office, which usually occurs during the day’s last working hour in Sweden. The image quality varied from decent to good, but the sound was usually clear and free of lags.

In the beginning of the study, the morning meetings occurred quite frequently, from every day to every second day in most cases. However, these meeting became less frequent as the projects and everyday work progressed, and the need for questions and clarifications became less necessary. In cases where more complex coding procedures needed to be explained, the technical project manager on the Swedish site used to share his desktop through Skype so that people in the American office could see, in real-time, what was happening on the remote computer and thereby more easily grasp intricate problems.
In the case of broader matters (as opposed to technical), such as discussing client meetings, there was usually a project manager present at both sites as well. These meetings were usually held in a bigger conference room equipped with a large television set with camera and microphone to support a larger amount of people. The function of these meetings was mainly to relay feedback from client’s requirements in order to supply the Swedish creative department with what they had to work with in terms of ideas for the project pitches. These types of meetings usually had a higher rate of decision making because of the people that attended from the Swedish office; project managers and managing directors were usually present to speed up the decision process. The managing director of the Swedish office tends to be a person that likes to be involved in both small and big matters in the company, which resulted in that many of the decisions went smoother and faster than they would have if they had to pass a middle man. This was especially crucial when there were only one to two overlapping office hours.

However, when the office hours were not sufficient the meetings were usually held in the evenings, Swedish time. This required off-hour work for some of the employees at the Swedish office.

One of the developers at the American office pointed out that in order to get as many overlapping hours with Sweden as possible, she had some morning meetings from her home before going to the office. This gave her and the Swedish office a longer time frame after the meeting in case something more came up before the overlapping hours were over.

**Discussion**

Dividing pre-studies and client material between the sites was in general a bad idea since it required a lot of synchronous meetings with more seasoned employees or managers in order to quickly commit to decisions. The seasoned employees and managers were not always available when needed for meetings, due to several other commitments. Because of the large time difference of nine hours, there was
limited time for synchronous meetings. This implicated that people at the Swedish office had to work after office hours and the American employees before office hours. People at the American office had to start their working day at 07.00 in order to have at least one hour of overlapping time for synchronous conversations. Some people at the office in Sweden came in later in the mornings only to be able to work later during the day – to create more overlapping hours for coordination. The managers that were needed during more client heavy periods sometimes had to work during their nights after work - a situation not sustainable in the long run. It was not an optimal idea to divide other work than development activities between the offices since it involved people from many different departments and levels in the organization, which created too big of an overhead. For this to work in the long run, the whole organization needs to rethink the way they work. From a local perspective to a global perspective – better processes for handling over-time and flexibility for their employees, formal allocation of time for meetings etc. In general, more resources regarding the activities that surround a DSD model are needed. If the culture within the organization is not ready for those requirements, it can put a strain on the communication and the organizational culture as a whole, with overworked employees as a result.

For large global companies it is almost a requirement that their culture is more oriented towards flexibility than in the case of SMEs. It is not uncommon for Indian developers to work during their night in order to match the daylight hours in Western countries.

5.1.2 Sharing Knowledge and Lack of Informal Communication

Since the office in Los Angeles was relatively new and had not yet got all the preferred roles within its web department, there could sometimes occur situations where knowledge within certain fields of development was lacking. This created situations where questions regarding certain programming practices had to be sent to the Swedish office, which had more experienced developers within each field (architecture, hosting and back-end).
When employees from the Swedish office visited the American one, there was an increase in knowledge sharing, which was mainly based on the fact that there was more time and a higher rate of understanding in face-to-face conversations. For instance, when standing together behind a screen and looking at problems to solve, or just pointing and talking, creates a higher level of interaction. One of the developers at the American office said that she appreciated having the more technically experienced co-worker from Sweden around during visits from the Swedish office: “I can learn new things (when there is Swedish employees visiting) that are not possible to learn through just Skype or mail conversations, mainly because there is no time for it.”

When these face-to-face situations were not possible, knowledge regarding programming practices was usually widely spread on the Internet, through written guides and videos. In some cases, this eliminated the need to ask colleagues for help, especially when working with well-established platforms and practices. However, the face-to-face knowledge sharing, a.k.a “coffee talk”, was still a better way to communicate and solve problems together.

Both synchronous and asynchronous information sharing was needed and complimented each other. Asynchronous information sharing was mainly used in forms of emails to clarify work requirements and instructions, while the synchronous counterpart usually had a knowledge sharing purpose or for project meetings - or sometimes just to “catch-up” in order to maintain team spirit.

The time and geographical difference between the sites made knowledge sharing both cumbersome and time consuming and relied on people’s personal willingness to work before and after regular work hours – rather than during the limited hours with overlapping time that often has to be used for more formal meetings and coordination matters.

**Discussion**

The lack of informal relationships with remotely located colleagues made it hard to know which colleague possessed a certain skill or expertise. It could also create a lack of trust, since informal relationships
have a harder time developing over distance and time zones, which could impact collaboration negatively. Therefore, tools of information exchange, such as forums, video conferences or chat systems that team members could use in order to speed up knowledge sharing and create informal relationships, appeared to be necessary. However, it seemed difficult to find tools that were both easy to integrate in the workflow, and that provided the required functionality in order to form informal relationships. Some tries were made by setting up an iPad in both locations’ coffee rooms, which shared a video stream with the other site. Its purpose was to serve as a team building mechanism, but the feeling of being supervised became an issue among the employees.

In general, synchronous information sharing over Skype had a higher degree of efficiency than emails, and was preferred when discussing complex coding solutions, in order to avoid misunderstandings. The more synchronous the conversation was, the higher potential efficiency it had – ranging from email and live chat to telephone and Skype’s video function with screen sharing. The screen sharing events were helpful because they gave direct feedback to the viewer and reduced misunderstandings, which sometimes could occur if information was given solely by voice or text.

5.2 Coordination
5.2.1 Modularization or Shift Work
During one of the Swedish web application projects, the American team joined in after the production had been in development for a while, approximately half way through. The Swedish team had mainly worked on the back-end part, building up the framework and structure of the website. The main reason for the American team to join in on this project was to speed up the development process on existing modules, and it seemed logical to work in shifts rather than to divide the rest of the project into modules.

On one occasion, the American office was sent wireframes to complete certain sections of the front-end. However, some parts of the
wireframes were based on older versions of the design ideas. Thus, no match between the design of the wireframes and the code that the Swedish shift had committed to the repository, for the American developers to continue with. This was realized a couple of hours during the American workday, when the Swedish office was no longer available for questions, which halted the production for the whole American work-shift. This was just an example that illustrates that not all projects are good for shift work. Therefore, breaking up the projects into independent parts, to be assembled later on, made more sense - especially if the parts were divided with aspect to the competences that was available at each site.

**Discussion**

In the situation mentioned above, with its nine-hour time difference, it was hard to maintain a synchronous collaboration model that was needed for shift work. Because of the time zone difference, it was barely two people working at the same time between the offices, which resulted in fewer opportunities for coordination. Additionally, it was difficult to foresee if roadblocks were going to occur, thus shift work often proved more troublesome than modularization work. Therefore, shift work was more appropriate when the coding features were harder to divide into modules and between the remote teams. Thus, shift work acted more as a secondary solution in order to speed up development.

In terms of efficiency, the two ways of working were practically the same. Shift work, however, needed more ongoing coordination, usually every day, which created a more obvious time overhead than modularization work did. However, in order to work in a modularized way, the need of planning beforehand was required to a higher degree, since changes would be harder to make during the work periods dedicated to finishing bigger modules.

When possible, modularization work was in general a better choice, since the independent parts could be worked on for a couple of days or more. However, as Lings et al (2007) says, that for software development work, we need to ensure that the architecture of the system is
consistent with the distributed structure of the teams. Therefore, the modules that were distributed to the American office needed to be within the scope of the competence within that office.

During these kinds of development projects, the team in Sweden work more or less without documentation. Instead, they just follow major features that need to be done during certain time frames agreed upon with the client, and therefore relying mostly on verbal communication within the team. This creates a void of information when the Swedish team is not present, which makes shift work not as suitable for their agile methodology. However, there will always be problems related to information void, even though the work is modularized. For instance, it can become harder to integrate parts of code, if the teams work too independently, due to conflicting assumptions.

There may, however, be ways to solve this. Firstly, let teams be able to go in and change off-site co-workers code on in order to clear roadblocks, if possible, when working in shifts. For that to be doable, it needs to be more transparency in what is going on in the whole project and a better understanding of what other co-workers are working with. Secondly, to work more thoroughly with documentation, and step down on the agile approach, so that off-site team can rely on documentation to a higher degree than verbal communication, and rely more on modularization work.

5.2.2 Lack of Team Awareness

Most of the projects that both offices worked on together had a common repository where code was uploaded, and when working in shifts this was usually done at the end of the day. The reason for this was that the off-site team members would be able to continue working when their day started, i.e. the next shift. However, the communication regarding what parts that were being worked on during each shift was not transparent or visible to the off-site team. Additionally, the communication that was held between the technical project manager in Sweden and the developers in America were was of a technical nature and task specific. Therefore, not giving an overall understanding
of the whole system architecture and what the other team member were doing, this led a feeling of isolation for the American team.

A tool called Trello (Trello, 2013), used as a Kanban board (or backlog), kept track of people’s specific responsibilities of programming tasks, for example “create menu for page X”. While it served its purpose as a Kanban board, it did not give an overall understanding to where the whole project was headed, and how the team felt about the progress that was made.

Discussion

The need for communication to be more transparent in order to reduce roadblocks became crucial during some projects – especially questions regarding requirements and the prioritization of features. (This is related to the issue with lack of informal communication mentioned in the Communication section.)

The system that was used – an online Kanban board – to organize features and keep track of team members’ work did not provide enough transparency for the off-site team in order to get full understanding of the projects. Since the projects were mainly based in Sweden with Swedish clients, the projects primarily used the American resources to speed up the development process. For the in-house team in Sweden it worked well because they shared more informal conversations together that overlapped the communicative gap of the Kanban board.

In order to tackle the team awareness issue an open transparent discussion board should be available to each project so the need for bottlenecks is reduced, and so that the communication is not centered through a specific role. However, the complexity of the articulation work in the context of distributed projects was somewhat unexpected by the employees beforehand. My findings suggested some specialized tools for the support of articulation work and project management. When one of these tools (Basecamp) was presented as a solution to the Swedish technical project manager, he said: “We tried that one before and it was too complex and took too much effort to be used in everyday work”.
Specialized tools for the support of articulation work are perhaps not that common in practice, since informal flexible communication channels were used instead. The general thought about these tools were that they was too complex.

5.2.3 The Communication Funnel and Team Spirit

Previously, before the researcher came to the American office there were not any Swedish employees in the web department. The communication had gone through the Swedish technical project manager to the developers in America, in English, either through email, telephone or video meetings. These conversations usually happened daily or a few times a week depending on the type of project, shift work or modularized work.

During this period of the field study, there was need of a Swedish technical project manager in the American team, in order to deal with the current Swedish clients in America. This role was filled during the field study by the researcher who took the role as an interim technical project manager. The fact that there were no previous formal roles at the American web department made an ongoing transition towards funneling the communication through the new Swedish technical project manager natural. However, this resulted in that the other team members in America, who previously had contact with the Swedish office, no longer received emails regarding new or ongoing projects. Emails started to be written in Swedish instead of English, and sent solely to the new technical project manager instead to all the departments’ members. Since internal email conversations regarding projects were written in Swedish, at the Swedish office, they were usually just forwarded to the American office in Swedish, and not translated. Naturally, this reduced the effort of having someone translating and coordinating the information in the Swedish office to English, but it had implications.

The new technical project manager now became the funnel for most of the information going to the American office and had the responsibility to forward information to the other co-workers in the web department. This created questions regarding the other American devel-
opers feeling of importance and put the new technical project manager in an uneasy situation of becoming a funnel for most of the communication between the offices.

**Discussion**

Some informal communication could be held in the native language, but information regarding the work related projects should be conversed in a commonly understood language to reduce risks of decreasing team spirit and team awareness. Otherwise, situations that create informal roles of authority could discourage team members if not handled accordingly.

One solution is to create more formal roles for communication or at least make sure everyone is informed on how the communication structure is set up between the remote teams. Formal communication roles should therefore be established before the projects. For instance, the two technical project managers between the sites could be responsible for the main information exchange, which is in turn distributed to the rest of the team. However, this information should at the same time be transparent in case any of these persons are not present, as mentioned before.

5.3 Control

5.3.1 Managing Project Artifacts

During one project there was an extreme time pressure. An application made in Sweden was to be presented to a client during a demonstration meeting, but in the last minute the Swedish office realized that they could not use the solution they had been working on. A new one needed to be created in one day (overnight in Sweden) to be presented to the client the following day. This situation benefitted from the increased hours of work because of the big time difference.

The application was an iPhone web solution prototype, for a future native application. The purpose of this was to show the basic func-
tionality of the application and to give the client a good enough representation of the final product.

Instructions were given through a telephone call to the American office (night time in Sweden) and through some written requirements sent through Skype’s chat. The work had been modularized so that a Swedish developer worked with a standalone part of the application that could be incorporated easily with the other part that the American office worked with.

The American office started working with their part that was the front-end design. However, after a couple of hours the developers realized that the URL to the other part being made in Sweden was incorrect and not possible to integrate or include in the final solution, which was the original plan. The persons responsible were no longer available online and there was nothing to do but to continue working “around” the problem. When the Swedish office saw our messages in the morning they were able to solve the problem on their end just in time for the presentation.

The whole development was made on a staging server in Sweden and also presented on the staging server to the client. This actually created problems because the American office was not informed that the client was using the application from the staging server. At one point, changes were made in the source code from the American office, which “broke” the application so that the presentation to the client was interrupted.

There was no control over the project artifacts in this case, mostly because the task was handed over by a manager that was outside the usual work process, and not the Swedish technical project manager.

**Discussion**

The situation described above could have been solved either through an email saying that the code was off limits to the American office, or by moving the source code to a deployment server so that no accidents could be made during the presentation. Additionally, by having a stricter chain of events when initiating projects, where information
always was channeled through certain roles. For instance, only between the technical project managers, that could decrease problems with losing control. These situations were unfortunately not uncommon due the fast pace of work in the industry. With many client meetings and tight deadlines, solutions had to be created in a short amount of time. However, for this work style to work, a formal methodology would be beneficial; one that clearly states what is done (a.k.a the “definition of done”, in agile terminology) and what is still under development, and then the code should be moved to a secure place.

In a broader scope, there are ways to deal with DSD with both agile and waterfall approaches, but the most appropriate solution is probably a mix with what is best of both. For instance, make concept, requirements elicitation and high-level architecture follow a traditional waterfall approach. This could be done either with American or Swedish clients at site. This lets us get detailed and clear requirements before starting with the development. When the project enters the development stage, we move to a more agile approach for developing, testing and releasing, which is done collaboratively between the teams at each site – as suggested in “Modularization or Shift Work”. The situation today, where the pre-development activities are not isolated to one site, creates confusion and control issues.

5.3.2 Allocation of Roles and Team Structure

During one of the projects for an American client that included a pre-study for a complex web application, there were no project managers or art directors on site at the American office. This resulted in lack of time/costs estimations on site in America and expertise regarding the visual aspects of the application. What the American office had at the time was a sales person, without technical experience, and a technical project manager to handle the technical aspects.

The pre-study meetings and interviews with the clients resulted in a requirements specification that was sent to the Swedish office and their project manager to approximate hours needed to complete the main features of the application and an estimated cost. From the design point of view, the Swedish designers had to rely on the infor-
mation given by the sales person and the technical project manager’s “user stories” from the client, to get a feeling of what the client wanted to convey and to which target group. Since these stories were passed on to the Swedish office, there might have been information that was lost, not only due to the fact that the information was re-told but also because of the language difference.

This role distribution at the American office was not optimal for these kinds of situations. Therefore, the American office ended up using material both made by the Swedish creative department and an America freelance designer. This resulted in diverse design material that did not present the project in its best form.

The results could have been improved if the two offices had worked with the same material all way through the project. However, the sales person at the American office was more inclined to use a local art designer for the pitch for the sake of the client, which had expressed their wish of local suppliers for this project.

Discussion

In order to tackle the role allocation problem, there need to be a better distribution of roles, so that the both offices are equipped with the right competences in-house, to actually manage clients on their own at the two different sites. However, being a SME, with limited resources and not actually equipped to handle end-clients at the American site at the moment, these situations would perhaps be better off not attempting at all. As it is today, the American office lacks some of the roles needed to fully function on their own. What the American office did have was merely competence related to sales and development. It did not have, for example a creative department. This created problems with the overall control of the projects, where issues of responsibility easily arise between the offices – resulting in political discussions.

However, in the case of just development activities, the need is reduced to a technical project manager and developers. That preferably had both back-end and front-end competence, in various programming practices, in order to handle roadblock internally within the site.
Pre development activities and their roles are preferably located at the same geographical location as the client to reduce the kind of problems discussed above.
6 Concluding Discussion

This chapter will conclude the key findings of my work; bring forth the main points from the previous chapter and present prospects for future research. My analysis has concluded into to some main topics that should to be taken into consideration when implementing DSD in SMEs. These topics are related to the initial and thoroughgoing discussion about communication, coordination and control.

6.1 Communication, Coordination and Control

In chapter one, we started with the era of globalization and the growing interest for small companies to look beyond national borders in order to gain competitive advantages. This case study focused on those small enterprises that established branches in other parts of the world for proximity reasons to a specific market, rather than to gain development advantages. However, since they now had the possibility to benefit from extra working hours and accelerated problem solving, why not look into this opportunity? Yet, with this opportunity there are obstacles; obstacles related to difference in time, geography and culture. These obstacles impact the very core activities of software development – communication, coordination and control.

The goal of this study was to gain an understanding of the problems and opportunities that occur in distributed software projects in small enterprises, in order to improve the outcome of DSD processes for SMEs. With that backdrop, the research question was: How are small enterprises’ processes in communication, coordination and control affected by distances in time, room and culture?

The previous literature in the field had mostly focused on problems related to larger and established enterprises in DSD. Did these find-
ings apply to those of my study: a small organization without prior routines or experience with this way of working?

My findings are important because they show how this newly started branch in America dealt with the problems and opportunities related to DSD. They further show how the investigated SME may have handled DSD problems differently than larger companies, where previous research already has been made. These findings are important for companies in similar situations, which want to investigate how a global DSD model might affect them and what obstacles to expect when incorporating this way of working.

6.1.1 Communication

The more informal relationships between employees in SMEs create an ideal setup for the communicative style of agile development. However, once adopting a DSD model, distance between sites creates barriers for communication. Spontaneous “coffee talk” between sites is reduced to a minimum if the time zones are just slightly overlapping the workdays. Even though small organizations usually have more informal communication (Kelly & Noonan, 2008), they still face the same problems as bigger organizations do when it comes to communication – especially the crucial aspect of “corridor talk” (Herbsleb & Moitra, 2001). Richardsson et al. (2008) argue that, for small organizations, the employees know each other to a higher degree, even if they work at different locations and never actually met face-to-face. However, due to the cultural differences in the investigated company and the employees not being used to the implications of a global way of working, put strains on the communication and the relationship between the two offices. Which according to Carmel & Agarwal (2001) could be a consequence of DSD. The more laidback and relaxed work culture in California sometimes annoyed the more “hard-working” Swedish employees.

In this specific case study, the time difference proved to be the main problem for communication, and one could argue that in order to better work out the culture differences through more informal relationships between the offices, the one hour overlapping time was not
enough. This lack of overlapping time could lead to more formal relationships between the teams and possibly trust issues, and therefore longer times to finish tasks. It is questionable what this overlapping hour is best used for. Since it was used for coordination matters, rather than “coffee talk”, it may also lead to lack of team spirit and knowledge sharing.

There is a need of keeping the communication flat and transparent and avoid putting single roles as communication channels. Communication solutions, such as online project management tools, could be used to reduce the need for synchronous communication in cases of overall project information and articulation work. It increases team awareness and speeds up the problem solving process by knowing other team members’ knowledge.

However, my findings and interviews suggest that specialized tools are often perceived too complex for everyday use. It follows that there is a need for deeper understanding of what kind of coordination is required in order to make successful progress in DSD and what factors that drive the need for coordination. So that these factors can be predicted for any kind type of project – in order to improve tools that best match the circumstances and what their tradeoffs are.

6.1.2 Coordination
The main coordination problem revolved around shift work or modularized work and as Herbsleb (2001) was also in the case study one of the main time consumers. Battin, Robert et al. (2001) argue that shift work could reduce coordination costs. However, at the investigated company, shift work proved to need more coordination than modularization work – probably because of the lack of overlapping time for coordinating, which modularization work required less of. In accordance with Espinosa and Carmel (2004), this indicates that reduced coordination costs could be an outcome of teams not working at the same time. Projects that are easy to divide into interdependent modules could, therefore, advantageously be modularized between the teams to reduce coordination costs. The teams could, however, become isolated, and reintegration of the modules could be troublesome,
due to conflicting assumptions and misunderstandings, an issue even for larger organizations, as Ågerfalk et al (2005) points out. Furthermore, modularization work requires that the sites have enough competence to internally handle their own roadblocks.

Shift work is more suited for those projects that are not easily divided into modules and/or when the offsite competence is not enough to handle big modules independently. Shift work benefits from being less dependent on multiple competences and may be more beneficial in cases of smaller off-site teams. However, modularization work is beneficial if the competence is present (and roadblock can be handled internally) and there is modularity within the features of the project. Since modularization work requires less communication and coordination, it is preferable to shift work.

If we work toward well-matched processes to handle modularization across the sites, it may be possible to reduce the amount of needed communication. And perhaps, if we could design architectures and features with isolation in mind, small organizations could benefit more from a modularized way of working.

6.1.3 Control
It is clear that working in only one location puts less emphasis on a formal methodology and control – especially if the projects require smaller teams, which probably is more common in SMEs. And as Ågerfalk, et al. (2005) state, control is important to ensure development of software on budget, in time and of required quality. We can assume that this is important in small organizations as well. However, the formal processes to regulate these control mechanisms did rarely reach the off-site location in America. Work requirements that were sent from Sweden to the American office, and developed, seldom received feedback on whether or not it reached the required quality levels, for instance. Neither was any formal “definition of done” stated, probably because this was more formally agreed upon in the Swedish office. These processes, however, never reached the American site, which could have benefitted from them to reduce misunderstandings and code conflicts. Further, as the complexity increases with addition-
al teams on different locations, the need of more formal processes and methodologies arises. Maybe there are ways to deal with DSD that encompasses both agile and waterfall approaches – a solution that mixes the advantages from both approaches. For instance, make concept, requirements elicitation and high-level architecture follow a traditional waterfall approach at one site only, favorably done at the client’s location. This lets us get detailed and clear requirements before starting with the development. When the project enters development stage we move to a more agile approach for developing, testing and release, where work is divided between the offsite locations, using e.g. modularization work. This does not necessarily reduce the importance of control mechanisms related to development, but it reduces the complexity of the pre-study and creative work done in this phase, and let the overall project control maintain within one site.

6.2 A prospect of future research

There are several areas that would benefit from further research within the field of globally DSD. For instance, is there a point where the overhead in time and resources is bigger than the actual gain from distributed work in small companies? In that case, how could we evaluate this efficiently? As expected, actual work practices may differ from the perceptions that managers have about development strategies in their companies.

Further, ethnographic research of articulation work and DSD in general gives an understanding of the processes related to software development, and how the development processes may require adjustments to match DSD settings for small enterprises. This leads us to the question of how to further improve articulation work in DSD settings. Therefore, achieving a better understanding of the needs of developers further field studies seem necessary. We currently have very little to go on for finding answers to these crucial questions and have a need for good theories that could provide a basis for the thoughts about compromises of DSD and predicting its outcomes for SMEs.
7 References

Study data

1. Field notes, 2013 (Alexander Rangevik)
2. Interviews form both Swedish and American offices, 2013
3. Documentation of technical aspects of tools used, 2013

Literature


Electronic references