Benefits and Challenges of Distributed Development and their Role in Global Software Development

University of Oulu
Information Processing Science
Bachelor's Thesis
Joni Barsk
09.09.2018

Abstract

Globally Distributed Development has arguably become a phenomenon in the past decades. Organizations have tried to utilize benefits of Globally Distributed Development and gain an upper hand in the global market. Organizations have tried Global Software Development, since the nineties with varying degrees of success.

This thesis analysed the effects that could cause an organization to begin or to give up a Globally Distributed Development process. This thesis was made as a literature review. Scientific literature was analysed to answer predetermined research questions: What has an impact in Global Software Development, which practises could be successful in Global Software Development and what differences could there be between Global Software Development and Global Game Software Development.

Effects that could influence the decision either way, were distributed to benefits and challenges. Then they were further divided to processes and dimensions. Processes were: communication, coordination and control. Dimensions were divided to: temporal dimension, geographical dimension and socio-cultural dimension. A three by three grid was made from the processes and dimensions. Effects were placed within the table and analysed, to see how they might influence Distributed Development.

Conclusions included that benefits could arguably be further divided to Effortless and Effortful benefits. These benefits are individual, depending on the organization, but in general, it was argued that Effortless benefits should be utilized as best as possible and Effortful benefits should be prioritized. Conclusions of challenges didn't yield as conclusive results. Arguably, most of the challenges are somehow connected. meaning that alleviating one will likely have adverse effects in another challenge. Careful planning, execution and follow-up was recommended when organization tries to alleviate different challenges of Distributed Development.

Analysis of the benefits and challenges yielded further information. Results were divided to three parts: communication, control and coordination. Most significant piece of information was the importance of planning. GSD without a well-formed plan is going to fail. Distributed development has too many moving parts to allowing hap-hazard plans and executions of those plans. Furthermore, importance of communication was highlighted within the analysis, as an integral part of success in GSD.

Utilization of Outsourcing in Globally Distributed Game Development is a significant part of game development, as multiple components of game-design require skills that a programmer might not possess. Voice acting and ability to make music to name just a few. For game software organization, utilisation of outsourcing could arguably be significantly more important than to a regular software organization. Not only due to the vast array of talents needed, but also due to the short period of time they are needed. As a result, Distributed Game Software Development could be called as: single site software organization, with outsourced autonomous multi-site subsidiary task development teams.

Finally, this thesis summarizes the literary review, by arguing that communication and pre-development work are arguably the two most important factors when considering the suggestions for a successful globally distributed development.

Keywords
Software Development, Distributed Software Development, Distributed Development, Global Software Development, Globally Distributed Game Development

Supervisor
Dr. Jouni Markkula

Contents

1. Introduction

Arguably Global Software Development has become a growing phenomenon in the past decades. This literature review analyses the reasons behind this phenomenon: why global software development or GSD for short, is done and what kind of benefits or challenges a software company might expect to face when expanding to the distributed software development. In addition, this thesis tries to collect a group of recommended approaches and assessments of benefits that are to be used for a successful software company that attempts to expand to globally distributed software development. To formulate guidelines for successful global software development, it requires analysis of the factors that have an impact to the global software development. This includes understanding aspects of the distributed development; which GSD is a sub-type off. First, this thesis defines the positive and negative aspects of the distributed development. Afterwards distributed development benefits and challenges will be compared to the global software development, resulting in a definition what are the good characteristics for a successful GSD and what are the important challenges for a GSD to face. Finally, found distributed software benefits and challenges will be compared with the global game software development, to gain an understanding of what is global game software development compared to global software development.

In this thesis, distributed software development teams of the same team and the outsourced teams by contract are considered as similar groups connected to the original project. Thus, teams and contracted teams are considered indistinguishable from each other for the thesis. In general, there are different ways to define terms, such as: distributed teams, distributed development, organizational structure of distributed development and distance in distributed development. In this thesis, all these different definitions for the sake of use are combined in more simplistic form.

For this thesis, some of the definitions are being condensed to a represent a single term, so that the thesis is easier to understand and comprehend. If the difference of the term is great enough the general term will be still used, but it will be further explained to fit the context, so that the purpose comes clear. *Global Software Development* is a form of software development where development teams are distributed in multiple locations (Ågerfalk, Fitzgerald, Holmstrom Olsson, Lings, Lundell, & Conchúir, 2005). Team that is part of a distributed software development and is in one of the locations where the distributed software development is being developed, is called distributed team, virtual team or remote team. For this thesis, all these teams are called *distributed team*. Benefits, positive side-effects or other forms of events that result in positive outcome are considered as a *Benefit. Challenge* is any occurrence or other event that result in negative outcome for the development or for the developing organization. *Organization* is a general term to represent any managing or governing body of the distributed development and its' teams.

Research questions:

"What has an impact in Global Software Development?"

To truly understand what is required for a good Global Software Development it is necessary to understand what a GSD is. Thus, in this thesis we priorities to figure out what is Distributed Development and from there what of these aspects impact the outcome of Global Software Development teams.

"What are the suggested practices for a successful Global Software Development?"

In addition to understanding the aspects that can impact the outcome of the Global software development, this thesis also evaluates what are good suggested practices for a Global software development organization. Additional questions would include, what does the organization, teams and individuals need to keep in mind for a successful Global software development.

"How does or does not, Global Software Development differ from Global Gaming Software Development?"

When previous questions are answered, this thesis analyses the differences and similarities in benefits and challenges between Global Gaming Software Development and Global Software Development.

1.1 Research process

This literature review will include research from multiple literature sources to find positive and negative aspects that impact distributed development. Then combining these results in a unified group of benefits and challenges. These results are then analysed, resulting in a conclusion of what positive and negative effects impact the distributed development. From the analysis, a group of suggested practices for an organization is made, which, when followed should have a positive impact on the organization. This literature review will be concluded by a comparison of the differences of global software development and global game software development, to analyse the differences between the two.

2. Distributed Software Development

Software development can be defined as distributed, when parts of the development team are separated by geographical distance (Lagerfeld et al., 2005). The distance is one of the defining aspects when trying to consider the subtype of a distributed software development. Definition of sub-types follows the description given in research by Jan, Dad, Amin, Hameed and Shah (2016). Depicting the following definitions: Domestic, Off-Shoring and Shoring. Domestic confines the teams within the same country. Off-Shoring includes domestic, but also has elements of parties being from neighboring countries. Finally, Shoring, in which the parties are divided globally, Shoring can be considered being anything beyond the scope of Off-Shoring.

Distributed development has three major dimensional aspects. According to Ågerfalk et al. (2005), they are temporal distance, geographical distance and socio-cultural distance. *Temporal distance* dimension is time-based distance. Temporal distance increases when the time between information-amount sent and the information-amount received is prolonged. Shortest temporal distance is during face-to-face communication. As described in global software development, *geographical distance* increases as the physical distance between operating members increase. *Socio-cultural* distance is the amount of difference in people's norms, values, cultures, social behavior, ethnicity and communications styles. These differences can be divided into organizational and national culture. As the geographical distance increases, it can be assumed that the temporal and socio-cultural does too. Although it is also true that socio-cultural distance can vary greatly even in small areas, such as districts in the United States of America.

Research by Carmel and Agarwal (2001), indicates that for a development team to function, it requires: Coordination, communication and control. *Coordination* is an act, where multiple entities are tasked to work together in harmony to further their objective. *Control* is a process where the entity is managing behavior and results of the objective. Entity gathers data and acts accordingly on those results to further the objective. Control can be formal or informal. In formal control, the entity has quantifiable evidence to back up the process, limitations and its changes. Whereas in informal control processes, entity does not have quantifiable evidence, instead the limitations are given as obscure instructions, such as authority pressure to finish an objective quicker. For example, final release date of a project is a form of formal control, whereas a project manager's suggestion to do a task faster is a form of informal control. *Communication* is an act where information is passed from entity to entity and the information transferred stays complete and whole during the entire process. Communication is an integral part of success in control and coordination processes, since without it, distributed development would lose one their major assets: ability to transfer information. (Carmel & Agarwal, 2001.)

Approach of this kind is chosen, since it fulfills and answers of all the questions given in "What is a distributed development". Also, it emphasizes the features given in global software development: Distance, people and communication. Ågerfalk et al. (2008), gave another approach, focusing in the people and their interaction with the system itself, it included organization, team and task processes to witness the effects of the positive outcomes of the distributed development. Furthermore, the research done by Carmel and Agarwal (2001) is one of the most cited sources in distributed development, thus it can

be deduced that the definitions in the research are valid and appropriate for use in this thesis.

2.1 History

Distributed software development's importance has increased in global market. Phenomenon is considered significant way to stay competitive and profitable in the global market. Large groups of skilled-labour from low-cost economies have created a new pool of workforce, which organizations can utilize as a low-salary workforce. Accessibility of the internet and with the more modern inclusion of the highspeed bandwidth and easy transportability of software code has allowed the use of simultaneous groups of developers that can be distributed among globe (Ågerfalk et al, 2008). Compared more traditional industries the transportation of software is much easier. Furthermore, it can be argued that initiation of distributed software development sites is easier, when compared to the hardware production industry. Ease of software transportation means that setting up an off-shore workplace is cheaper, since transfer of heavy equipment is not needed. Software development teams' greatest need is proper working area and the hardware to do their work. In comparison, mechanical industry requires not only workspace, but also assembly line or other suitable constructions, which will house the building, assembly or development of the products that the entity is producing. Usually it requires a transportation or creation of heavy industry equipment, which are not cheap compared to a set of computers for a distributed team.

According to Akbar and Safdar (2015) "Since early ninetees, Global Software Development (GSD) had started affecting the global IT industry." (p.314) First well documented case of global software development was done by International Business Machines (IBM), who in the mid 1990's created a time-zone-efficient software development process, that followed the principles of follow-the-sun approach. IBM's process proved unsuccessful and was discontinued after initiation (Carmel, Espionasa & Dubistky, 2010). Later Siemens have done multiple attempts at creating a successful software by distributed development (Sangwan et al., 2007), one of which was a study to improve and understand the effects and issues of the global software development (Herbsleb, Paulish & Bass, 2005).

2.2 Global Software Development

Global software development is a sub-class development method of the distributed software development. GSD is under the sub-class of Shoring. The teams are distributed by distance, in multiple geographic sites. These teams can be part of the same organization, collaboration of a different organization or be completely outsourced. The difference in distance can be anywhere from the five continents, to small as 2 countries in same continent. (Jan et al., 2016; Sangwan et al., 2007.)

Sosio-cultural distance does not necessarily increase in linear ratio with the increase of geographical distance, since great differences can occur event at short distances, especially in socio-cultural distance. The Asian culture has proved difficult for some of

the western software corporations to deal with. This can be interpreted that the socio-cultural distance was too great for the distributed teams to work with. Thus, other high quality but cheap labour, must be found closer by (McKinsey Global Institute, 2003).

Follow-the-sun (figure 1) is an example of GSD utilization and utilization of time-zone-efficiency process. In the follow-the-sun process the project and its artefacts are passed on to the next distributed team that are located roughly 8 hours to the west. Then the distributed team works on the project for the duration of their work shift, then the project is again passed to the next team located roughly 8 hours to the west. Project and its artefacts are send to the west after work shifts, eventually creating a loop where the distributed team is ready to receive the project when it has completed its earth's rotation. Enabling a process where the project is being worked on for around-the-clock. Further creating a work flow where the organization can constantly focus on producing the product, without any downtime. (Carmel, Espinosa & Dubinsky, 2005.)

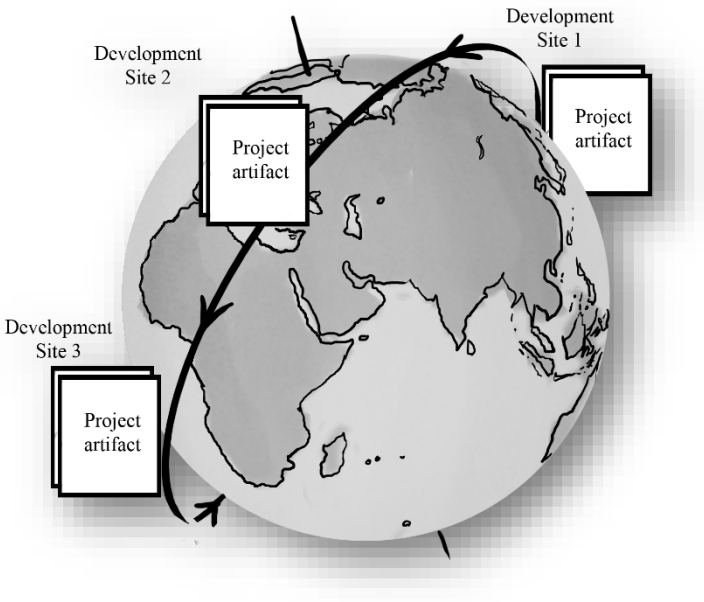

Figure 1: Follow-the-sun approach

3. Benefits of Distributed Development

Cost benefit is an often mentioned when referring to distributed development benefits (Carmel & Agarwal, 2001; Ålgerfalk et al., 2008). For this thesis, cost is a too abstract of a term to be used, when analysing the benefits of the distributed development. As such, and analysis need to be made to study the impact of different benefits to the distributed development. This includes, definitions of other characteristics that can be considered beneficial, but does not necessarily affect the cost in a straightforward manner.

Other benefits to the distributed development, such as "The 'Unknown' Benefits of GSD" (Ågerfalk et al., 2008, p.3). These beneficial side effects, as mentioned by Ågerfalk et al. (2008), include: Organizational benefits, Innovation and shared best practices, improved resource allocation, team benefits, improved task modularization, reduced co-ordination cost, increased team autonomy, process task benefits, formal record of communication, improved documentation and clearly defined process. Then, this thesis analyses the benefits of the distributed software development and analyses the impactfulness that a benefit might have to the organization. This is done by reviewing one or more research sources to ascertain from the context of how impactful a benefit is for the distributed software development. Benefits are divided in temporal dimension followed by geographical dimension and socio-cultural dimension. Dimensions are further divided by the processes that are used to indicate how the process is handled within the organization or in the distributed team. Some of the benefits are mentioned multiple times in different sections of the table, and thus will be analyzed multiple times in their respective dimensions and processes.

Beneficial and neutral effects and side effects are given in Table 1. Frame and frame titles are based on the table from Ågerfalk et al. (2005), copy of the table can be found in Appendix. Contents of the table are a collection of the effects mentioned in sources: Ågerfalk et al. (2005); Ågerfalk et al. (2008); Carmel and Agarwal (2001) and Herbsleb and Moitra (2001).

Table 1. Beneficial and neutral effects of Distributed Development.

Process	Dimension		
	Temporal Distance	**Geographical Distance**	**Socio-Cultural Distance**
Communication	- Time zone effectiveness - Reduced time to market	- Proximity to market/customer	- Innovation and shared best practices - Asynchronous communication preferred by non-native speakers - Formal records of communication - Improved documentation
Coordination	- Time zone efficiency - Modularization of work	- Access to large labour pool - Standardization in work practices and communication - Allocation of roles and team structure - Modularization of work	- Mix of skills and experiences - Language and cultural training - Standardization in work practices - Improved team autonomy
Control	- Time zone effectiveness - Deployment of distributed teams and corporations	- Improved resource allocation	- Clearly defined process

3.1 Temporal Benefits

Time-zone-efficiency (Communication). One of the advantages of asynchronous communication is the digital trail left by the communication. This by-product of communication can be then used to ease communication between outside members of the original communication interaction. When the developer teams are expected to hand-out the project to the next group in the line, asynchronous communication encourages better documentation and communication, which will shorten the step between being handed the project to work at the project, instead of just understanding, what did the previous team work on. This round-the-clock communication process gives the ability for sped up completion of the project, due to improved number of hours spent on the project per day. (Algerfalk et al., 2005.)

Reduced time to market (Communication). As with time-zone-efficiency asynchronous productions enables the possibility for an organization to produce project faster as it is being done round-the-clock. This increase in production will shorten the time for project completion. (Ålgerfalk et al., 2008.)

Time-zone-efficiency (Coordination). Distributed teams that are working in multiple sites or shifts are going to asynchronously develop or solve problems can coordinate their efforts among each other. Coordination efficiency increases when tasks are requested and

completed asynchronously during shifts. Asynchronous distributed team can complete the task, by the time distributed team that requested the task is back at work. This continuous production eases coordination costs as there is no downtime due to waiting, for a task to be completed. (Algerfalk et al., 2005.) In later research, according to Ålgerfalk et al. (2008) titles this differently, by naming it "reduced coordination costs".

Modularization of work (Coordination). In addition to rapid deployment, distributed teams also create modularization in the organization. Teams far away have reduced communication, which is reflected in their work processes. This will result in modularization, which will affect work processes, which is not necessarily a bad thing. Teams will know their boundaries and the teams automatically will understand the proper junction points in the developed software. Modularization of work is considered by Ågerfalk et al. (2005), as a byproduct of a good documentation and communication between the groups. When done correctly it will ease information flow within the organization and allow members of the module to be more creative as they have more authority over their own work, which in turn will reduce the project's managerial cross-over.

Time-zone-efficiency (Control). Paasivaara and Lassenius (2006) mention in their research that with an implementation of agile methods, development teams can also gain the additional benefit of short iteration cycles. This, combined with follow-the-sun approach can allow constant flow of information between every shift, towards the management office, which will in turn improve the management's knowledge of the current situation.

Deployment of distributed teams and corporations (Control) has two main points. Organizational structure or development team can be created to anywhere in the word, to meet the demands of a rapid market developments (Herbsleb & Moitra, 2001), but also the organization can deploy additional teams, due to having access to much greater skilled-labour force, compared to the area of recruitment of local organizations. All of this allows the organization to push their product to the market much faster, since the earlier an organization starts the earlier they finish.

3.2 Geographical Benefits

Proximity to customer or market (Communication). Creating a physical development site, near the customer, allows the organization to recruit members from the same culture as the local market. Further improving the positive feedback of the organization as new jobs are given to the local people. This positive media can allow further contracts to be made within the local customer base. Thus, improving the maker gains in the area (Ebert & Neve, 2001). Furthermore, the proximity to the culture with the new employees in the organization will allow, better understanding of the traditions and communications. It can have drawbacks, like cultural difference in working customs, but it also gives insight to the customer base. With the decreased distance to the market, the developing teams can now also create a software more closely fitted for the customer. Furthermore, impeded software development can gain from having a hardware development in close proximity to the software development for further coordination and control. This will shorten the necessary time for the market launch. The local site can also allow, better connections to marketing, local partners and improve communication and coordination with the local partners (Ålgerfalk et al., 2008).

Access to large labour pool (Coordination) was arguably one of the original reasons behind distributed development, such as offshoring. Instead of hiring personnel from other country and flying them to the development location, the organization instead creates a site in to the location where recruitment of highly skilled workers is plausible (Herbsleb & Moitra, 2001). Furthermore, salaries of these highly skilled workers vary between regions. Thus, organizations can be willing to set up distributed software development teams far away from the original development team of the organization in hopes of highly skilled and cheap labour. (Ålgerfalk et al., 2008.)

Standardization of work practices and communication (Coordination). The entire organization will benefit from the standardization and modularization due to asynchronous nature of distributed development. With proper control this simplifies responsibilities and actions within organization, creating better coordination. With modularization of teams, the development teams can have highly modularized work assignments. This allows modularization of work between development teams, resulting in decrease of overlap and improve hand-out quality between the distributed teams. Overlap of work happens when team members or a team is not fully aware of the boundaries of their work, which can result in excess work and social-friction between the teams and team members part of that assignment. Another benefit of standardization of work practices is the improved handouts, since modules created by the distributed teams are made so that their implementation to the main project or to other modules is more thought out. (Ålgerfalk et al., 2005) and (Ålgerfalk et al., 2008.)

Allocation of roles and team structure (Coordination). Organization in distributed development has access to large and cheaper labour pool. This addition to manpower can replace personnel with expensive and singular expertise. These now redundant experts can be rotated where they are needed, instead of being laid out. Keeping the personnel in the team, not only keep the expertise in the organization, but also creates stability for the workforce as a stable and steady workplace. (Ålgerfalk et al., 2005) and (Ålgerfalk et al., 2008).

Modularization of work (Coordination) is benefit from one of the root causes that is within geographically distributed development. Geographical distance creates boundaries between development teams, which will create boundaries between teams' responsibilities. According to Conway's law: "The structure of the system mirrors the structure of the organization that designed it." (Herbsleb & Grinter, 1999, p.85). These boundaries can allow parallel work processes to be done as well, as it allows to create clearly defined modules, where only the integration to the other software components is to be done in unison with other developer teams (figure 2) (Conchúir, Holmström, Ågerfalk & Fitzgerald, 2006). Whether the organization chooses the follow-the-sun approach or modularity by dividing individual software assignments to development teams. In both cases, due to the increased communication with other bodies of development teams. All teams will gain significant improvements in official communication and coordination within organization, since it is one of the only ways to communicate between teams. This helps with backtracking, rechecking requirements and distributing information to nonparticipants of the original communication. Whereas informal discussion usually is only spoken and not written to anywhere. Redistribution of informal agreements is less convenient, since they need to be written first. (Ålgerfalk et al., 2005.)

Improved resource allocation (Control). As with *allocation of roles and team structure* distributed teams of an organization can respond to multitude of development challenges by redistributing experts to the distributed team in need of help. Furthermore, this resource allocation enables this expertise to spread to multiple distributed teams in the organization, creating more skillful personnel and more effective teamwork. (Ålgerfalk et al., 2005) and (Ålgerfalk et al., 2008).

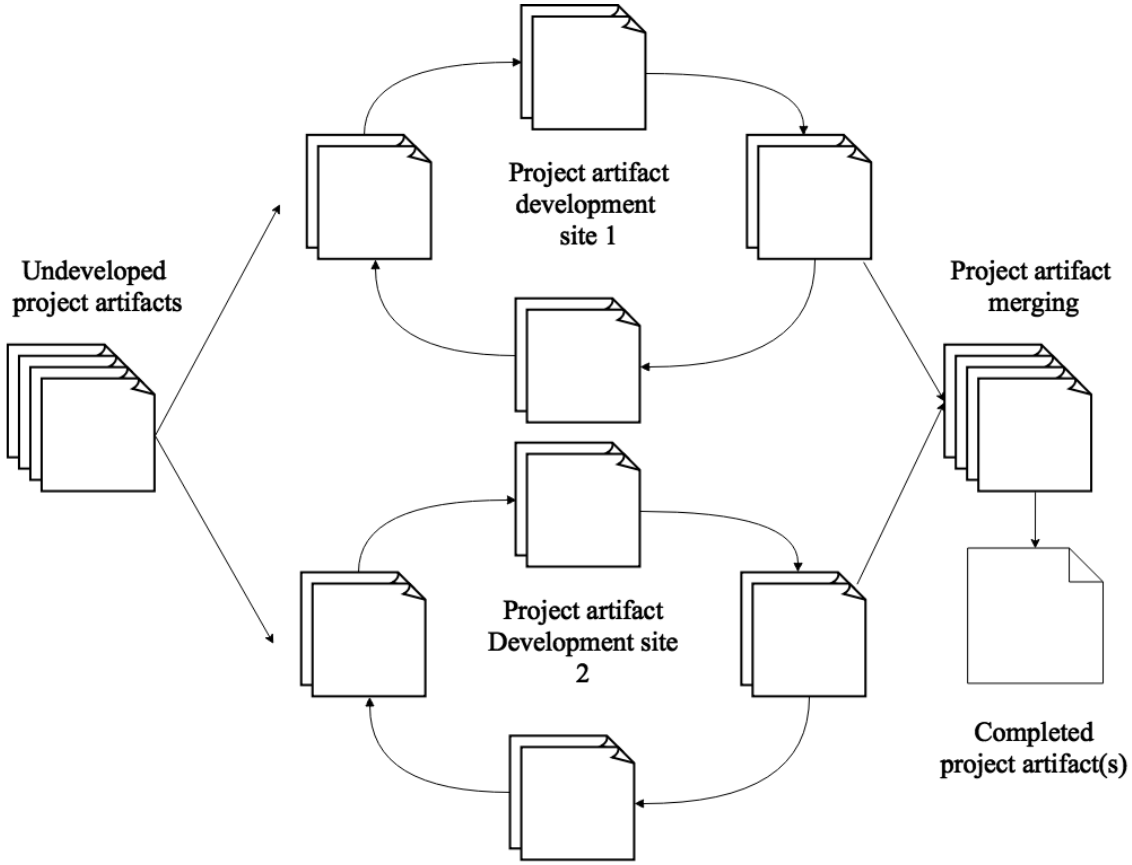

Figure 2: Modularization of work in distributed development

3.3 Socio-Cultural Benefits

Innovation and shared best practices (Communication). Organization may have gain in innovation by exposing itself and its methods to new employees with different nationalistic, culturist and organizational backgrounds. Resulting in improved or entirely new and better practices in all forms of communication, coordination and control. (Ålgerfalk et al., 2008.)

Asynchronous communication preferred by non-native speakers (Communication). As previously mentioned, in distributed development and in situations where the shifts do hand-offs, due to difference in time zones or in work shifts. The next team does not have any informal communication, which is usually performed by using the organization's primary language. While within the development team's internal communication, can be done in language done of by majority. the Asynchronous communication allows communication in (Ålgerfalk et al., 2005.)

Formal records of communication (Communication). As the team arrives to the site, the information can be shared by written text. Emails and other forms of text-based communication allows non-native speakers to construe their thoughts in a better form, which will improve the communication and decrease the chance of mistakes. (Ålgerfalk et al., 2005)

Improved documentation (Communication). As already implied in geographical distance and on formal records of communication, communication becomes standardized, since communication can only be made asynchronously between teams. Meaning that all forms of communication will leave a trace. Which will increase traceability and accountability (Ålgerfalk et al., 2008). Gumm (2006) stated in her research that organizations will face challenges in distribution of information within the organization. Furthermore, according to Gumm (2006) "Organizational distribution emerged as bigger challenge than physical distribution." (p. 50). Organization which have already solved their challenges in physical distance has an advantage when tackling the problem of organizational distribution of information, since first challenges of distance-based communication is already solved. Whether solving of communication has been done intentionally or not, the lack of face-to-face communication will require more formal methods of communication e.g. e-mail. Delone, Espinosa, Lee and Carmel (2005) also stated in their research that access to knowledge sharing between development teams allowed further improvements in communication.

Mix of skills and experiences (Coordination). As already mentioned in geographical distance, socio-cultural distance's members of a different culture can also come for nearby or from far. Whether the case these people with different cultural background can bring new policies, skills and experiences to the organization and to the development team. Furthermore, when the organization needs to learn a new culture, due to finding a new market location from a different culture. It is beneficial to have employees of the same culture as the new market. Which again, as previously mentioned, can increase and improve communication with customers and introduce new possible partners. (Ålgerfalk et al., 2005.)

Language and cultural training (Coordination). In addition to benefits to the organization, the new mix of culture and their people, can also teach and improve already employed personnel, giving them a chance to learn new learn new languages, learn of new cultures, integrate and study new techniques, share new experiences and improve communications between each other. (Ålgerfalk et al., 2005.)

*Standardization in work practices (Coordination). G*eographical distance's standardizations in work practices is about creating an even playing field for all teams that par take in the project's development. When standardization of work practices is compared from socio-cultural distance, distributed team might be able to implement their local knowledge base and culture to lessen the amount of overhead. (Ålgerfalk et al., 2005.)

Improved team autonomy (Control) are improved due to the organizational and geographical limitations that limit the micro-management within the distributed developer team. This will leave more autonomy and responsibilities with the team. Increase trust in the team will increase the morale of the work-force and give the development team-lead more freedom to use his and his team's expertise to provide a better product (Ålgergalk et al., 2008). This also includes use of the expertise of the

individuals, which can be transferred between sites, according to previously mentioned resource allocation.

Clearly defined processes (Control). According to Ålgerfalk et al. (2008) organization of distributed development takes process definitions into consideration more than their non-distributed organization counterpart. This careful approach to processes makes them better defined and thus easier to modularize if necessary.

3.4 Conclusions of Benefits

Benefits can arguably be separated to two different groups. Effortless benefits and effortful benefits. Effortless benefits are a byproduct of implementation of distributed development, whether intended or not, whereas effortful benefits can be only gained when they are properly planned, executed and followed up. Depending on the organization and its' distributed teams, benefits can change between being effortful or effortless. For example, distributed team that has employees from the same socio-cultural distance flown to them, do not acquire the effortless benefit of *mixed skills and experiences* and *innovations and shared practices*, rather the organization must go through the effort of hiring local people to acquire the mentioned benefits for their organization.

Organization and its' management has to weigh the amount of different benefits the organization can gain in changing its procedures. As some of the effortful benefits might require sacrifices in behalf of other benefits. As such it can be argued that thorough analysis of effortful and effortless benefits is paramount for the organization's success.

4. Challenges of Distributed Development

Due to phenomenon of globalization of information technology many forms of distributed development have increased in geographical distance. Furthermore, there has been an increase of outsourcing of development in recent years. Akbar and Safdar (2015.). Arguably a common theme has emerged from the negative aspects of distributed development; communication problems.

Drawbacks or *challenges* will be a hindrance to the overall performance of the distributed software development. These challenges can manifest in multitude of ways: some can be a mild inconvenience where as some might have critical consequences. Challenges are divided in temporal dimension followed by geographical dimension and socio-cultural dimension. Dimensions are further divided by the processes that are used to indicate how the process is handled within the organization or in the distributed team. Some of the challenges are mentioned multiple times in different sections of the table, and thus will be analyzed multiple times in their respective dimensions and processes.

Challenges of distributed software development are given in Table 2. Frame and section titles for the table are from Ågerfalk et al. (2005), copy of the table can be found in Appendix. Contents of the table are a collection of the effects mentioned in sources: Ågerfalk et al. (2005); Ågerfalk et al. (2008); Carmel and Agarwal (2001); Herbsleb, Mockus, Finholt and Grinter (2001) and Herbsleb and Moitra (2001).

Table 2. Challenges in distributed development.

Process	Dimension		
	Temporal Distance	**Geographical Distance**	**Socio-Cultural Distance**
Communication	- Delayed communication - Delayed feedback	- Lack of informal communication - Dependency on ICT - Increased effort to initiate contact - Providing technical infrastructure - Cost of travel	- Language differences and misunderstandings - Managing frames of reference
Coordination	- Reduced hours of collaboration - Synchronized team meetings difficult - Availability of technical infrastructure - Coordination complexity - Lack of mechanisms for creating shared understanding - Management of project artefacts	- Reduced trust - Lack of awareness/team spirit - Lack of mechanisms for creating shared understanding - Coordination complexity	- Language and cultural training - Lack of domain knowledge - Doubtful of others' capabilities - Lack of mechanisms for creating shared understanding - Coordination complexity - Lack of awareness/team spirit
Control	- Management of project artefacts	- Lack of concurrent engineering principles	- Perceived threat from low-cost alternatives - Adapting to local formalized norm structures - Different perceptions of authority/hierarchy

4.1 Temporal Challenges

Delayed communication (Communication). One of the most apparent issue of distributed development, is the increased communication times. This is due to the asynchronous nature of the distributed development. Face-to-face communication is limited only to the co-location of the workspace of the distributed team. All other communications are limited to how the distributed teams are available when interaction is necessary. Distributed communication methods include, but is not limited to, video conferences, e-mail, phone calls and instant messaging. When dealing with temporally asynchronous communication, the teams are limited to forms of communication where information must wait until receiver can read it. Asynchronous communication brings another problem, since if the communication is done by e-mail for example. The distributed teams are not synchronously working at the respective sites. This form of written communication is

prone to mistakes and misunderstandings, which may result in further delays in communication. (Ålgerfalk et al., 2005; Herbsleb et al., 2001; Hersleb & Moitra, 2001.)

Delayed feedback (Communication) is an extension of a challenge derived from *delayed communication.* When members of different distributed teams communicate in temporally asynchronous manner the quality of communication becomes important, as all mistakes and misunderstanding cannot be corrected until the next time the asynchronous cycle rotate. This challenge might manifest for example when first shift gives misinformation to the second shift. The second shift is unable to know the answer until the start of their next shift, assuming first shift is informed of the mistake and it is corrected in time for the shift switch. If further mistakes are made or subject of a problem is hard to convey in asynchronous manner. It might require multiple back and forth discussions between distributed teams. This can result in delays of multiple days in productions as information is not conveyed in timely fashion. (Ålgerfalk et al., 2005.)

Reduced hours of communication (Coordination). Temporally distributed teams, whether working in shifts or separated by time-zones, will have trouble finding time to coordinate discussions even when the teams are separated by small temporal distance. According to Ålgerfalk et al. (2005), if two distributed teams are differentiated by a single time-zone difference. Their hours of communication can be cut up to four hours per day. This is due to having two-hour loss from start to end of the workday and further two-hour loss from two separate one-hour lunch breaks. This reduction in communication is further increased, as the temporal distance increases. (Ålgerfalk et al., 2005.) Distributed teams that are working asynchronously have restricted time between their shifts to communicate and coordinate between each other. For example, in "follow-the-sun" approach, this limited time frame is designated for the teams to hand-off their work and bring the next team up to speed. This overlap between shifts can be anything from seconds up to hours and in some cases, it might not exist at all. Problem is that limited overlap generates delays on communication, but increasing overlap decreases worktime done by the distributed team. (Herbsleb et al.,2001.) According to Conchúir, Ågerfalk, Olsson and Fitzgerald (2009), teams that followed follow-the-sun approach started to overlap more hours thus lessening the original strength of follow-the-sun's benefit of around-the-clock work times. Reportedly this was, because some parts of the development are more inclined to need collaboration between teams. According to Conchúir et al. (2009), this resulted in distributed teams moving their worktimes to overlap between each other.

Synchronized team meetings difficult (Coordination). As time of communication and coordination decreases between distributed teams do to temporal distance, so does the time to plan, synchronize and coordinate distributed team meetings. This can result in situations where distributed teams might be forced to meet off hours, especially when there are teams two or more distributed teams that have great differences of temporal distance. (Ålgerfalk et al., 2005.)

Availability of technical infrastructure (Coordination). According to Ebert and De Neve (2001), the lack of meaningful technical infrastructure, such as distributed version control or database synchronization meant that teams could not communicate efficiently between each other round-the-clock, each day of the week. Lack of availability of technical infrastructure, has been somewhat alleviated by new generation of programs such as GitHub (GitHub, Inc., 2008). GitHub is specifically designed to tackle the problem of distributed version control. Nevertheless, distributed software development infrastructure is hindered by the fact that teams are likely to be forced use these programs to be able to manage their technical version control in asynchronous way. (Ålgerfalk et al., 2005.)

Coordination complexity (Coordination) is a general reference to a situation where a group of people or in this case distributed teams are tasked with a single complex task. According to Ålgerfalk et al. (2005) it is necessary for the team not to rely on a single individual for inter-team coordination, since the absence of said person would be troublesome to the distributed team. Furthermore, delays and costs of communication, generate even more issues for coordination that must be corrected, which will generate more complexity to the coordination effort. According to Conchúir et al. (2009), distributed teams develop software slower than single site team. New distributed team members took more time to get started with the development and more management roles were reported.

Lack of mechanisms for creating shared understanding (Coordination). According to Ålgerfalk et al. (2005), Lack of mechanisms for creating shared understanding can manifest as faulty, delayed or even missing information to some of the intended recipients. It is necessary for the information to be distributed between teams in a correct, full and timely manner. Shared understanding is hindered, because asynchronous communication and temporal distance creates gap between given and received information. This challenge can be alleviated with a proper planning and architecture for the organization.

Management of project artifacts (Coordination) According to Ålgerfalk et al. (2005), traditional way to manage artifacts is creation of centralized hub or a repository, where artifacts are saved and synchronized. Challenge comes when an artifact is updated synchronously and no common standard or procedure is established between distributed teams. Distributed teams may be unaware of individual artefact statuses, if there is a possibility of multiple distributed teams to operate the same artefact.

Management of project artifacts (Control). Follows the same principles as coordination challenge of management of project artifacts. Control can be established with proper enforcing of procedures and standards. Neglection of artifact management can lead to poor interoperability and consistency between the project artifacts. (Ålgerfalk et al., 2005.)

4.2 Geographical Challenges

Lack of informal communication (Communication). Distributed team can gain the benefits of informal face-to-face communication within their team, which can improve the artefact management, establish new contacts between co-workers and improve information distribution within the team. Inter-team communication is a challenge, because teams are geographically separated from each other. Hindrances coming from the lack of informal communication can be alleviated, if information sharing, communication technology and communication architecture has been created to combat the lack of informal communication. (Ålgerfalk et al., 2005 and Conchúir et al., 2009.)

Dependency on information and communication technologies. Due to distance between distributed teams, communication has almost complete dependency on information and communication technologies. Meaning that as the distance grows between the distributed teams, so does the difficulty of travelling to the other development sites. This dependency is relatively common in software development *(Communication)*. According to Ålgerfalk et al. (2005), communication infrastructure is more important for distributed

development, as nearly all communication is done that way. Quality of communication tools, training for the communication tools and the corresponding infrastructure to support these tools do not only alleviate the hindrances that the communication distance poses, but also is a necessity for distributed development. Ålgerfalk et al. (2005.)

Increased effort to initiate contact (Communication). According to Herbsleb et al. (2001), member of a distributed team, who needs help from another distributed team, has increased effort to initiate contact. This increased effort is twofold, it is harder to communicate asynchronously if the problem is relatively complex, but also, it is hard to know whom to contact. For example, if a person in a distributed team needs a clarification to a problem. He or she needs to first figure out who is capable to answer to his or her problem. After correct person is located, then initial contact needs to be made. Only after of which can there be communication. In asynchronous communication, this would be done in written communication, but in complex cases, it would require a scheduled meeting, which in temporal advert is mentioned as hard to do. Thus, more time is removed from the development of the software. Furthermore, according to Ålgerfalk et al. (2005) increased effort to initiate contact can result to members creating their own solutions to the problem, instead of trying to contact a person that might know about the problem. These fixes according to Ålgerfalk et al. (2005), can slow down the production of the software, if the fixes proved to be erroneous.

Provision of technical infrastructure (Communication). According to Ålgerfalk et al. (2005), one problem of the geographical distance is the *provision of technical infrastructure*, it is not directly the fault of the organization, but the problem of geographical distance between vendor's retail sites across the globe. Organization cannot rely on the fact that 3rd party vendors, redistributors and retailers can provide the same tools and same version of the tools to every single distributed team across the globe. This lack of tool's version control can lead to problems when interacting between distributed teams. Furthermore, according Ålgerfalk et al. (2005), different regulations and prohibitions may differ between countries, which can result to excessive amount of planning to gain similar tools to all distributed teams or worse yet, it might result in having to compromise between teams to have difference in the equipment they use.

Cost of travel (Communication). Cost of travel is a natural expense that can be assumed to increase as geographical distance increases. This increase is roughly a linear one, which means, that in globally distributed software development they are as high as they can get. According to Ålgerfalk et al. (2005), face-to-face communication is important in the early phases of development. These trips are not only costly, but they are also time consuming.

Reduced trust (Coordination). Lack of informal communication can lead to *reduced trust*. Members of different distributed teams will have trouble generating contacts and relationships between each other in a natural way. When knowledge of the other person's capabilities is unknown, it can lead to unwillingness to ask for help or unwillingness to communicate changes or other ad-hoc bug-fixes. (Ålgerfalk et al., 2005.) Geographical distance relationships not only decrease the level of trust between members, but it also increases. (Ålgerfalk et al., 2005 & Conchúir et al., 2009.)

Lack of awareness and team spirit (Coordination). According to Ålgerfalk et al. (2005), Distributed team lack the inter-team relations, since informal communication is not possible. In addition to being unaware of what others are doing, it also creates conflict among teams, when something is done wrong or something is done without another team's consent. For example, one team improves code, but is not aware that one team is already doing it. Lack of team spirit is also a socio-cultural problem if teams or team

members divide in to cultural groups. Lack of trust and team spirit can be combatted by increased amount of travelling between sites by different team members. This swapping of distributed team members can create relationships between other distributed team members that continue when team members are reunited with their original distributed team. (Ålgerfalk et al., 2005 & Conchúir et al., 2009.)

Lack of mechanisms for creating shared understanding (Coordination). Similar to lack of mechanisms for creating shared understanding under temporal distance. The geographical distance derives distributed teams from discussing face-to-face about the project and its' artifacts. Preventing teams to understand the more minute details of the project. (Ålgerfalk et al., 2005.)

Coordination complexity (Coordination). Similar to Coordination complexity in temporal distance. When face-to-face communication is hard due to geographical distance, teams must rely in the more limited forms of communication, which will make it harder to coordinate efficiently. (Ålgerfalk et al., 2005 & Conchúir et al., 2009.)

Lack of concurrent engineering principles (Control). As stated in Ålgerfalk et al. (2005), standards, protocol and synchronization are important when dealing with the hand-off of processes between distributed teams. Both teams need to be aware of the state the process is in and in what state the process must be in that the next team is able to receive the project as smoothly as possible. In distributed development, the hand-off must be done in a synchrony between the distributed teams, while still being restricted with the asynchronous nature of the distributed development. To alleviate the problem, the teams need to have good communication and support tools to ensure the seamless transition of work. This is not only a challenge for infrastructure, architecture and technology, but also a challenge for the team members to communicating between each other without the aid of the informal communication. (Ålgerfalk et al., 2005.)

4.3 Socio-Cultural Challenge

Language differences and misunderstandings (Communication) arise when members of distributed teams are recruited from different socio-cultural backgrounds. Employees are unlikely to have their mother tongue same as the primary language of the organizations, primary language of most global organizations is English. Ålgerfalk et al. (2005.) Misunderstandings, according to Casey and Richardson. (2004), can occur even when both teams have the same primary language. Differences in dialects and resulting misnomers can lead to unnecessary conflicts and reduced trust between the teams. Similarly, team members of the minority language can face problems to bond.

Managing frames of reference (Communication). Socio-cultural distance not only effects the language, but also the minute or significant differences in culture. As stated in Casey and Richardson (2004), even the cultural difference between groups from US and Ireland, resulted in conflict as teams' perception of authority differed enough to create conflict. Furthermore, according to Ålgerfalk et al. (2005), inability to create understanding between the teams, will result in decrease of successful communication. In addition, national or regional socio-cultural differences, can result in disproportionate amount of work for the other distributed team, as some cultures see overwork in a differently. *Language and cultural training (Coordination).* According to Ålgerfalk et al. (2005), when a team member is transferred to a different distributed team. It is a necessity to train the transferee to the social, cultural and lingual aspects of the recipient distributed team.

This training, is not only costly, but also time consuming, since the time is not spent working. Furthermore, it might be necessary to rotate different people between sites to create relationships and new contacts, thus socio-cultural training, might have to be done to multiple people.

Lack of domain knowledge (Coordination). If work is outsourced from the most involved participants, who understand the project domain, there is a chance that the lack of familiarity to the domain from the outsourced team might cause problems, such as excessive learning and relearning of the project's domain. Ålgerfalk et al. (2005), gave an example where Norwegian tax system was outsourced to Russian developer team who were not familiar with the Norwegian tax system. Ålgerfalk et al. (2005.)

Doubtful of others' capabilities (Coordination). Where geographical distance can create reduced trust due distance, can socio-cultural distance create doubt in others' capabilities. For example, teams that come from cultures that are perceived as highly technological, might not trust in the capabilities of other distributed team that comes from a culture that is of a lower standing in technological achievements. As with reduced trust, best to overcome this challenge is by continuous communication and creation of relationships between distributed teams. Ålgerfalk et al. (2005.)

Lack of mechanisms for creating shared understanding (Coordination). Inquinate information sharing mechanism can result to missing or delayed information on temporal and geographical distance. On socio-cultural distance challenges emerge as distributed teams are unable to decipher correctly the given information. Distributed teams might create a different vision from a same set of information give. (Ålgerfalk et al., 2005.) This challenge be alleviated via having proper mechanisms and standardizations in the project's early phases.

Coordination complexity (Coordination). Similar to temporal and geographical distance. Coordination complexity increases as the socio-cultural distance grows. Teams need to be more and more careful during inter-team communications to avoid misunderstandings. (Ålgerfalk et al., 2005)

Lack of awareness/team spirit (Coordination). According to Herbsleb and Moitra (2001), distributed teams and their members can have different interpretations of urgency due to the socio-cultural distance between distributed teams. Furthermore, according to Herbsleb and Moitra (2001) inter-team communication might be left unread if the sender-receiver's socio-cultural distance is great enough and the sender's subject is written in a way that receiver fails to understand the message in a clear way.

Perceived threat from low-cost alternatives (Control). Common problem with all forms of off-sourcing whether industrial or distributed software. Workers in the higher-cost economies perceive software distribution to low-cost economies as a threat to their jobs. According to Ålgerfalk et al. (2005), this perceived threat can lead to unwanted competitive behaviour and in versus mentality that hinders teamwork, rises the threshold for asking help from other distributed team and makes handoffs between distributed teams more difficult.

Adapting to local formalized norm structures (Control), is partially geographical challenge, but in priority it is a socio-cultural challenge. Organization that is expanding to new socio-cultural sites, must take into the fact that the laws, tradition and regulation are bound to change when national borders or socio-cultural boundaries are being crossed. These differences in local formalized norm structures need to be studied and

organizational deployment in the area must be changed accordingly to the challenges faces. Ålgerfalk et al. (2005.)

Different perceptions of authority/hierarchy (Control). Socio-cultural distance increases differences in perception and in relationship to authority. These differences can manifest via difference of trust to the authority. For example, U.S developers are more trustworthy of authority than Irish, which believe trust must be earned. Ålgerfalk et al. (2005.)

4.4 Conclusions of Challenges

Challenges are an interweaving web of contradictions and exceptions. Not only they are highly dependable on processes, dimensional distances, organization and distributed teams, but they are also dependable on management and the surrounding elements like politics and culture. Distributed development challenges and their alleviating fixes could be described as a set of linear sliders that are interlinked together. When alleviating one challenge, one or more challenges must be un-alleviated. For example, to lessen the *cost of travel*, *reduced trust*, *lack of awareness/team spirit* and *doubtful of others' capabilities* are going to be un-alleviated or made worse, because transporting distributed team members between sites is important to alleviate the challenges. Furthermore, it is necessary to expect all the challenges, but it is also useful to realize that not all the challenges might manifest during the implementation of distributed development. Meaning, for example, *lack of mechanism to create shared understanding* might not manifest, such as a good implementation of shared understanding is already on place and works immediately for the distributed teams. These kinds of happy accidents are not to be relied upon but are to be utilized if they do happen.

5. Suggested Practices for Globally Distributed Development

With the beneficial and challenging effects being analyzed it is also important to see how these interact with each other. Due to the scope of this literature review, not all possible combinations of beneficial and challenging factors can be reviewed. Instead, this thesis examines what could be considered as high value benefits and challenges in GSD success during. As such a few major points are taken into consideration.

As generalization, the most common and sought-after benefit of globally distributed development is *access to large labour pool* and *time zone effectiveness*. Other benefits that are regarded highly are *reduced time to market*, *innovation and shared best practices* and *mix of skills and experiences*. (Herbsleb & Moitra, 2001; Carmel & Agarwal, 2001 and Casey & Richardson, 2004.) With these priorities in mind the organization makes multiple choices when considering their benefits. With the assumption that the organization is interested in time zone effectiveness, it can be argued that the organization will have long distances in geographical dimension and as geographical distance increases so does the temporal distance. What it leaves for organization, is choice of a benefit in socio-cultural distance. Choosing for example the balance between *innovation and shared best practices*, *mix of skills and experiences*, *reduced trust* and *language differences and misunderstandings*. Not to mention the balancing difficulties in effort and alleviating of benefits and challenges.

Whereas the benefits for an organization to push their product with a GSD plan are relatively simple to pinpoint, it is much harder to establish a clear challenge that would hinder GSD the most. No single challenge is critical enough on its' own to stop GSD, when comparing the different subsections of processes and their respective dimensional distances, as mentioned in chapter 4.4. The Issues from the challenges are intermixed, since resolving a challenge may require the leverage of negative effect of another challenging factor. As a conclusion, these arguably critical benefits and challenges should be leveraged or alleviated in the GSDs favor by making a group of suggested practices that when followed can increase the possibility of success for the organization in GSD.

More modern developments in GSD, distributed development and in technology have made changes the way that is GSD is done. Improved communication and virtual technology has made agile methods more feasible. Agile methods have gained popularity because the product can be observed by the customer during its lifespan, customer's wishes and demands can be included to the software even after the start of the development. Especially small to medium organizations value the employee-client contact, because reputation and customer service imago is valuable for them. (Akbar & Safdar, 2015.)

5.1 Suggested Practices for Communication

Providing a platform for communication within distributed team and among distributed teams (Ebert & De Neve, 2001). Making sure project's details, project's progression, project's up-coming plans and distributed teams' information is found from a single site that is easily accessible by all distributed teams and distributed team members (Ebert & De Neve, 2001; Herbsleb, Paulish & Bass, 2005). According to Herbsleb, Paulish and

Bass (2005), distributed team members need to have understanding on how and whom to contact when it is necessary. Furthermore, it is necessary for distributed team members to understand the premise that communication is to be answered and answered in a timely manner. Creation a communication environment where bottlenecks are to be avoided and establish communication infrastructure were asynchronous multi-party conversations are possible, in addition to normal asynchronous and synchronous communication (Herbsleb, Paulish & Bass, 2005).

5.2 Suggested Practices for Coordination

Organization needs to establish measurable goals and tracking metrics for management, management infrastructure, project and project artifacts (Ebert & De Neve, 2001; Herbsleb, Paulish & Bass, 2005). Furthermore, distributed teams need to have *clearly defined processes*: requirements, goals and tasks within the project. Organization requires a continuous risk assessment where highest prioritized risks are constantly monitored. All committed plans should be documented and distributed to right people as fast as possible. Organization needs to plan a way to mix distributed teams, this is to leverage the benefit of *shared best practices and innovations*. (Ebert & De Neve, 2001.) Herbsleb, Paulish and Bass (2005) states that travelling is most important at early stages of development to establish trust and overcome cultural distance.

5.3 Suggested Practices for Control

Creation of a platform where process model can be changed to fit the current needs of project development, project artifact development or a specific distributed team. Organization's project has a singular project leader who is responsible for the project and its progress. Project leader also requires a management team with members from the cultures that are participating in the project. Distributed teams are responsible for the product assigned to them. (Ebert & De Neve, 2001.)

5.4 Conclusions on Suggested Practices

Suggestion for the best way to go about GSD is impossible as there are no two situations alike. Instead it is more feasible to have a group of abstract suggestions on how to plan, start up and follow up on the process of GSD.

Planning is arguably the most important part of GSD, it is necessary for the organization to get this right, and more importantly after the planning to choose whether to execute the GSD or not. Planning phase should include estimations on gains and costs for the benefits and challenges of Globally Distributed Software Development. This would include assessment of values and risks of choosing a specific location on the globe as the next development site. Necessary travelling costs need to be estimated and plans need to be made for continuous back and forth travelling, especially early days of the distributed development life. Responsibility is shifted to individual distributed teams and to their respective managers, choosing good managers make or break the individual teams and their coordinated efforts for good communication. This also means that a more modularized development style is preferable. Still not forgetting that constant

communication between teams is important. As such an easy method of communication is necessary. Modern day instant messaging communication forums, repositories and alike provide a base for mandatory inter-distributed team meetings and discussions.

At the beginning of the development of the distributed software development. Inter-distributed team meetings should be mandatory daily and further communication is encouraged as much as possible. Inter-distributed team cross exchange of members is to be done regularly to further create relationships between members and create key inter-team personnel when asking for help. Travelling is to be emphasized during the early days of the distributed development. Upon contact, it is necessary for both parties to communicate both ways, even in hard or hard to understand questions. Cultural and lingual basics are to be learned on both sides and preferably it is started before the first contact is made. It is necessary to emphasize that misunderstandings and requests for repeats will be common and to be expected. Each team should have minimum of one person who is capable to handle miscommunications in discreet and responsible manner. Furthermore, distributed team members should to acquire basic understanding of each other's work and what they are working on currently, this is to help the spread of information, if a question is sent from the other team.

Following up with the distributed teams and the development of the software. Create environment where making and adapting to changes is possible. If things don't work, change them. If they do work, then improve upon it. Arguably, generating a culture where distributed teams are responsible for their work, allows higher quality and making sure that the distributed team leaders or managers are responsible for not only their specific product, but also for the product overall, improves not only interface merging quality, but also the product overall. Progress and set deadlines should be required to help with declaring responsibility and allowing better oversight. Require and follow data on all necessary processes, so that the progress of the development of the globally distributed software development can be tracked and acted upon, if needed.

6. Utilization of Outsourcing in Global Game Software Development

Games, arguably in its' simplest terms are software that is part art and part computer science. Games have some corresponding similarities to other software development, such as a need for designers, programmers, management and marketing. According to Bethke (2003), games may for example, further require writing, music, voice-overs, motion capture and special effects.

Arguably, one of the differences between regular software development and game software development is that the success of the game product development relies on its' sales, whereas a general software is usually contracted and purchased beforehand. This could have the consequence that the organization responsible for the development of the game is pushed to minimize *the time to market*, as every moment not spent selling is going to cost for whomever pays the salary. If a publishing company sponsors a development of a game. the game development organizations in question could be considered as an outsourcing of a game development from its' publishing partner, which in turn could be considered as a move for the benefit of *proximity to market*.

6.1 Utilization of Outsourcing

In the scope of the research of this thesis, it could be generalized that the outsourced game development team, functions similar to autonomous *distributed team*. This is because, they are following similar suggested principles as distributed teams, such as, teams need to have a priority contact between outsourced and development teams. The outsourced team is required to communicate in daily basis with the development team. (Chandler, 2014.) Outsourced team can generate modularized products, such as voice-overs. One major difference between the two teams is that the contract usually lasts only for the time of the product development.

When an organization is planning to develop a game, the number of people required to make a single site development team is financially very challenging. This, according to Bethke (2003), is for few reasons: first, a big game needs a lot of personnel, with very specific talents. Secondly the talents necessary for quality product are hard to find or expensive to hire, because they are working for someone else. Lastly, those specific talents that would be hired may only be needed for very short amount of time. (Bethke, 2003.) Furthermore, as mentioned in 3.2, benefit of *allocation of roles and team structure* is the improved stability of employees when laying out and rehiring is being avoided, but for example music artist cannot be expected to do programming, when his original job is completed. According to Bethke (2003), most game developer organizations have at least some level of out-sourcing. For example, music, voice-overs and marketing are a way to outsource. Bethke (2003) states that core-programming and art should be done within organization, but very modularized aspects of programming can be outsourced. The challenge of out-sourcing art, according to Bethke (2003), is that the constant fixes and tweaks in design and differences in designer's and artist's vision make the development of the art product cumbersome. To generalize, if your organization is unable to do a quality work on a specific area of development, then it is necessary to outsource it. (Bethke, 2003)

According to Chandler (2014), Game developer loses part of the ability to reassign resources and tasks within the organization, if some of the key-elements are being outsourced. Meaning that the development team can be contractually bound to deliver development assets to outsourced teams on time, even if there are more critical tasks in the development that the development team should be completing. Furthermore, if the outsourced team fails to meet the deadlines, game development organization is not able to relocate personnel to improve the phase of the development. (Chandler, 2014.) One of the benefits in distributed software development is *improved resource allocation*. Arguably in outsourcing this inability to manage personnel and resources within organization limits the benefit gained in globally distributed game development.

6.2 Conclusions on Outsourcing in Globally Distributed Game Development

When comparing GSD and global game software development, game development seems to rely more heavily on outsourcing. Some of the building blocks of game development are too far apart for one skill set to complete. Thus, game development teams are arguably more willing to add outsourcing to the to the development plan. Voice-lines, story and music are examples of game development building blocks where coding experience is not needed for a good product, so either game developer must hire personnel with broad skill set, multiple sets of personnel for each skill required or contract a 3rd party who possesses the skillsets required to complete the task. All in all, global game software development could be described in reference to distributed development as Globally distributed game development, with single site software development and autonomous multi-site subsidiary task development.

7. Conclusions

This thesis began with a literary review of the benefits and challenges of Distributed Development. It was preceded by an introduction to the basic terminology within Distributed development and a summary of historical cases, where Globally Distributed Software Development has been in use.

Plan for this thesis, was to study the differences of global gaming software development to other forms of global software development. A historical summary of previous attempts of Globally Distributed Development was researched. Different benefits and challenges of Distributed Development were analyzed. Analysis results created a list of suggested practices which could help to understand the impact they might have to GSD and thus to Global Gaming Software Development.

Analysis of the benefits and challenges within DD was performed. It involved a table where these effects were divided by six different factors. Then the three by three table was opened and analyzed piece by piece. Resulting in a conclusion were beneficial factors can be divided to effortful and effortless. This divide is individual on case by case basis. Use of effortless benefits should be used and an emphasis on the most beneficial, effortful benefits should be prioritized. With challenges the conclusion was little less easy, as challenges are hardly an individual problem and their mitigation is usually done with a cost to other challenges. Result was the emphasis of careful study, analysis and prioritization of challenges and their mitigation within the organization, before a DD is started.

After the conclusion of benefits and challenges, a generalization of suggested practices was made. High prioritization for early days of the GSD was emphasized, as establishment of teams' individual and inter-team communication are arguably one of the paramount reasons for success in GSD. Furthermore, emphasizing travelling and skill of management level personnel is useful. Distributed site's team leaders and managers need to be able to see problems before they arise and deal with them accordingly.

Finally, it was analyzed that Global Gaming Software Development could be treated similar to a specific from of GSD, thus expanding to a study of the differences in benefits and challenges between GSD and Global Game Software Development. Results of this analysis was that the gaming software development could be argued to follow similar practices to an autonomous, globally distributed software development, with high emphasis on modularization and a drive to find highly skilled labor around the world, whether hired or outsourced.

As a conclusion, it could be argued that GSD is a massive undertaking, which requires not only lot of pre-work before the establishment of an off-site, but also a high emphasis is needed for the initial months of the development, to allow a solidified communication structure, where each employee knows their place, knows their immediate inter-team colleagues, knows the linchpins of other development sites and knows the personnel above and below in the overall structure. During development, emphasis turn more into the management staff, who are responsible for finding and reacting to challenges within the organization.

Further studies, that could spawn interesting results, could be the study and comparison of multiple different GSD strategies, such as how does a large gaming company's development differ to a smaller gaming company that outsources more of their work.

Another interesting study would be a side by side comparison of large corporations' GSD strategies and their differences in prioritization between each other. Third option for an interesting study would be the comparison between one large globally distributed software development project and one large globally distributed gaming software project. These comparisons are arguably a massive undertaking and would require lot of cooperation from all parties and furthermore would require quite a lot time as the development life cycle could be long. Final and more enclosed study could be the analysis of communication methods and their pros and cons between the distributed teams. Especially in cases, where their teamwork is considered significant.

References

Ågerfalk, P. J., Fitzgerald, B., Holmstrom Olsson, H., Lings, B., Lundell, B., & Ó Conchúir, E. (2005). A framework for considering opportunities and threats in distributed software development.

Ågerfalk, P. J., Fitzgerald, B., Holmström Olsson, H., & Ó Conchúir, E. (2008). *Benefits of global software development: The known and unknown* doi:10.1007/978-3-540-79588-9_1

Akbar, R., & Safdar, S. (2015). A short review of global software development (GSD) and latest software development trends. *Computer, Communications, and Control Technology (I4CT), 2015 International Conference on,* 314-317.

Bethke, E. (2003). Game Development and Production
Wordware Publishing 2003 (412pages)

Casey, V., & Richardson, I. (2004). Practical experience of virtual team software development.

Carmel, E., Espinosa, J. A., & Dubinsky, Y. (2010). " Follow the sun" workflow in global software development. *Journal of Management Information Systems, 27*(1), 17-38.

Carmel, E., & Agarwal, R. (2001). Tactical approaches for alleviating distance in global software development. *IEEE Software, 18*(2), 22-29. doi:10.1109/52.914734

Chandler H. M. (2014). The Game Production Handbook, Third Edition
Jones and Bartlett Learning 2014 (482 pages)

Conchúir, E. Ó., Ågerfalk, P. J., Olsson, H. H., & Fitzgerald, B. (2009). Global software development: Where are the benefits? *Communications of the ACM, 52*(8), 127-131.

Conchúir, E. Ó., Holmstrom, H., Agerfalk, J., & Fitzgerald, B. (2006). Exploring the assumed benefits of global software development. *2006 IEEE International Conference on Global Software Engineering (ICGSE'06),* 159-168.

Ebert, C., & De Neve, P. (2001). Surviving global software development. *IEEE Software, 18*(2), 62-69. doi:10.1109/52.914748

Farrell, D., Baily M. N., Agrawal V., Bansal V., Beacom T., Kaka N., Keiriwal M., Kumar A., Palmade V., Remes J., & Heinzl T. (2003). Offshoring: Is it a win-win game? *McKinsey Global Institute,*

Gumm, D. C. (2006). Distribution dimensions in software development projects: A taxonomy. *IEEE Software, 23*(5), 45-51. doi:10.1109/MS.2006.122

Guzmán, J. G., Ramos, J. S., Seco, A. A., & Esteban, A. S. (2012). Success factors for the management of global virtual teams for software development. *Enhancing the Modern Organization through Information Technology Professionals: Research, Studies, and Techniques: Research, Studies, and Techniques*, 239.

Herbsleb, J. D., Mockus, A., Finholt, T. A., & Grinter, R. E. (2001). An empirical study of global software development: Distance and speed. *Proceedings of the 23rd International Conference on Software Engineering,* 81-90.

Herbsleb, J. D., & Moitra, D. (2001). Global software development. *IEEE Software, 18*(2), 16-20.

Herbsleb, J. D., Paulish, D. J., & Bass, M. (2005). Global software development at Siemens: Experience from nine projects. *Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005.* 524-533.

Herbsleb, J. D., & Grinter, R. E. (1999). Splitting the organization and integrating the code: Conway's law revisited. *Proceedings - International Conference on Software Engineering*, 85-96.

Jalote, P., & Jain, G. (2006). Assigning tasks in a 24-h software development model. *Journal of Systems and Software, 79*(7), 904-911.

Jan, S. R., Dad, F., Amin, N., Hameed, A., & Shah, S. S. A. (2016). Issues in global software development (communication, coordination and trust) A critical review. *Training, 6*(7), 8.

Paasivaara, M., & Lassenius, C. (2006). Could global software development benefit from agile methods? *2006 IEEE International Conference on Global Software Engineering (ICGSE'06),* 109-113.

Sangwan, R., Bass M., Mullick N., Paulish D. J., & Kazmeier J., (2007). *Global software development handbook*. Boca Raton, FL: Auerbach Publications.

W. DeLone, J. A. Espinosa, Gwanhoo Lee, & E. Carmel. (2005). Bridging global boundaries for IS project success. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences,* 48b-48b. doi:10.1109/HICSS.2005.126

Appendix

Table 1: Copy Framework of distributed development issues

Table 2: Framework of distributed development issues.

Process	*Dimension*		
	Temporal Distance	**Geographical Distance**	**Socio-Cultural Distance**
Communication	☺ Time zone effectiveness ☹ Delayed communication ☹ Delayed feedback	☺ Proximity to market/customer ☹ Lack of informal communication ☹ Dependency on ICT ☹ Increased effort to initiate contact ☹ Providing technical infrastructure ☹ Cost of travel	☺ Innovation and shared best practices 😐 Asynchronous communication preferred by non-native speakers ☹ Language differences and misunderstandings ☹ Managing frames of reference
Coordination	☺ Time zone efficiency ☹ Reduced hours of collaboration ☹ Synchronised team meetings difficult ☹ Availability of technical infrastructure ☹ Coordination complexity ☺ *Modularisation of work* ☹ *Lack of mechanisms for creating shared understanding* ☹ *Management of project artefacts*	☺ Access to large labour pool 😐 Standardisation in work practices ☺ Allocation of roles and team structure ☹ Reduced trust ☹ Lack of awareness/team spirit ☺ Modularisation of work ☹ Lack of mechanisms for creating shared understanding ☹ *Coordination complexity*	☺ Mix of skills and experiences 😐 Language and cultural training ☹ Lack of domain knowledge ☹ Doubtful of others' capabilities ☹ *Lack of mechanisms for creating shared understanding* 😐 *Standardisation in work practices* ☹ *Coordination complexity* ☹ *Lack of awareness/team spirit*
Control	☹ Management of project artefacts ☺ *Time zone effectiveness*	☹ Lack of concurrent engineering principles ☺ *Allocation of roles and team structure*	☹ Perceived threat from low-cost alternatives ☹ Adapting to local formalized norm structures ☹ Different perceptions of authority/hierarchy

(Ålgerfalk et al. 2005, p.12)