



2020_TR_06 Lero Technical Report

A Global Teaming Model for Medical Device Regulated Software Development

Adapted from MSc in Technology Management dissertation at University of
Limerick, 2014

Adeline Kinehan, supervised by Prof Ita Richardson.

Abstract

Global software development has become a common choice for developing software due to the benefits it promises. Such benefits include reduced costs, increased skilled labour pool, innovation, leveraging existing resources, proximity to customer and standardising systems. Companies enter into global software development without considering the problems that it introduces. Software development is a complex enough process when teams are located in the same office. Global software development (GSD) involves remote software development across geographically distributed countries. This introduces communication, control, coordination and cultural issues to name a few that impacts how the team work together and the success of the product produced. This thesis looks at global software development in the context of a regulated medical device manufacturer. Developing software for medical device manufacturers requires certain regulatory criteria to be met to gain approval to sell products in a country. This criterion includes strict process and controlled documentation that provide evidence to testify to the safety and quality levels of the medical devices. This thesis considers if the regulation that medical device manufacturers are subject to contributes to the problems in global distribution of software development. A case study organization is used for gathering data through interviews and company documentation. A literature review is conducted to identify aspects of GSD and regulation for consideration in a regulated GSD scenario. These findings are combined with a global teaming model that was developed to support management of global software development from existing research to provide a new updated model for global software development that incorporates the regulated medical device industry. The objective of this model is to improve the operation and outcome of these regulated software development projects.

Contents

Chapter 1: Introduction.....	1
1.1 Background to the Research	1
1.2 Research Objective and Questions	2
1.3 The Significance of the Research	3
Chapter 2: Literature Review.....	4
2.1 Introduction.....	4
2.2 Global Software Development	4
2.3 Global Software Development Configurations	5
2.4 Global Software Development Drivers	6
2.5 Global Software Development Challenges.....	7
2.5.1 Communication.....	7
2.5.2 Control	8
2.5.3 Coordination	8
2.5.4 Culture	9
2.5.5 Trust.....	9
2.6 Global Software Development Solutions	10
2.6.2 Reduce Cultural Distance	11
2.6.3 Reduce Temporal Distance.....	11
2.6.4 Trust and Personal Considerations	11
2.6.5 Face to Face Meetings	12
2.6.6 Process Maturity	12
2.6.7 Technology Tools	12
2.6.8 Agile Methods	13
2.6.9 Global Teaming Model.....	15
2.7 Regulation in Medical Device Manufacturing	16
2.8 Regulation in Software Development for Medical Devices	16
2.9 Regulation in Global Software Development.....	20
2.10 Conceptual Framework.....	23
2.11 Conclusion	23
Chapter 3: Research Methodology	25
3.1 Introduction.....	25
3.2 Research Questions.....	25
3.3 Research Philosophy.....	25
3.4 Research Methodologies.....	26
3.5 Research Strategy	27

3.6 Research Design	28
3.6.1 Interview Protocol	30
3.6.2 Content Analysis.....	33
3.7 Limitations of the Research	34
3.8 Conclusion	35
Chapter 4: Findings.....	36
4.1 Introduction.....	36
4.2 Global Teaming Model.....	36
4.2.1 Define Global Project Management	36
4.2.2 Define Management between Locations	41
4.3 Regulation in GSD.....	45
4.3.1 Software Development Lifecycle	45
4.3.2 Risk Management	46
4.3.3 Quality Management System.....	46
4.3.4 Non Regulated Systems	48
4.3.5 Customs Regulation.....	48
4.4 Other Findings	49
4.4.1 Tools	49
4.4.2 Team Dynamics	50
4.5 Conclusion	51
Chapter 5: Discussion	52
5.1 Introduction.....	52
5.2 Local Priorities.....	52
5.3 Decision Making.....	53
5.4 Global Standards.....	54
5.5 Change Management for Transitioning to GSD.....	55
5.6 Organizational Culture	55
5.7 Project Management and Lessons Learned	56
5.8 Medical Device Software Regulation in GSD.....	56
5.9 A Global Teaming Model for Medical Device Regulated Software Development.....	58
5.10 Conclusion	62
Chapter 6: Conclusions and Further Research.....	63
6.1 Introduction.....	63
6.2 GSD in Regulated Medical Device Manufacturers	63
6.3 A Global Teaming Model for Medical Device Regulated Software Development.....	64
6.4 Limitations of the study	65

6.5 Further Research	66
6.6 Overall Conclusion	67
Bibliography	68
Appendices	77
Appendix A: Interview Protocol.....	77
Appendix B: Codes for Content Analysis	80

List of Tables

Table 1 Global Software Engineering Factors (Richardson et al., 2012)	155
Table 2 Interview Participant Profiles	31
Table 3 Proposed updates to SG 1: Define Global Project Management.....	59
Table 4 Proposed updates to SG 2: Define Management between Locations..	59
Table 5 Proposed SG 3: Define Global Regulation Management.....	61

List of Figures

Figure 1 Global Teaming Model (Richardson et al., 2012).....	144
Figure 2 GAMP V-Model.....	20
Figure 3 Conceptual Framework for GSD and Regulation	23
Figure 4 Research Design.....	30
Figure 5 Medi Inc Development Verification Process using a V Model	45
Figure 6 Proposed SG3: Define Global Regulation Management.....	60

Chapter 1: Introduction

This chapter provides an introduction to the topic under investigation, global software development and regulation of the medical device industry. Its purpose is to define the area of research and the reason for pursuing it. The context of the research, a medical device manufacturer is introduced along with its motivation for adopting GSD. From this the research question is formed which also leads to a research objective that this study serves to answer. An outline for the following chapters concludes this first chapter.

1.1 Background to the Research

Global Software Development (GSD) is described by Herbsleb and Moitra (2001) as software development that is multisite, multicultural and distributed globally. GSD is attractive as it offers access to a wider range of skills and expertise, potential for reduced costs, round the clock working and proximity to local customers (Ågerfalk et al., 2005). Quality, flexibility, increased productivity and risk dilution are also motivators for GSD (Prikladnicki et al., 2006). Where a company is already operating globally it may drive the use of these global resources for global software development (Karolak, 1999, Damian et al., 2003). Also the globalisation of products and markets contributes to this distribution of projects (Herbsleb, 2007).

Medi Inc is an example of such a company. It is a large multinational medical device manufacturer headquartered in the US with a global presence in many other countries. It has organised its manufacturing plants at strategic locations across the world to meet local market demands, evenly disperse their distribution channels and benefit from lower cost base economies and accessibility to skilled workforces. Several of these sites have software development capabilities in Ireland, Australia, United Kingdom, France and US that were set up to support the operation of their local business. Since the core business, the manufacture of medical devices for the global market is the same across these sites there is a commonality present that allows the sharing

of expertise and benefit of collaboration. In an effort to take advantage of this, standardise their systems, leverage existing resources and eliminate duplication of effort they have adopted a global software development configuration. Through this a global software solution is agreed and developed using global resources. This centralizes planning and solutions which offers standardization, integration and economies of scale (DeSanctis and Jackson, 1994). This development is completely internal as part of the company's strategy to align its business and IT strategies, a widely accepted arrangement for adding value and competitive advantage (Henderson and Venkatraman, 1993, Kearns and Lederer, 2000). Its business process planning already falls under a global function in the form of Enterprise Resource Planning (ERP) (Holland and Light, 1999) with IT now encompassed for alignment. Its vision is a global one that views the company as one with all sites working towards one global solution.

The premise is admirable but software development is a complex process only exacerbated by the distance introduced by taking it global (Herbsleb and Moitra, 2001). Further to this the medical device industry in which this research is situated is subject to regulation and high quality standards. This is due to the safety aspect of implantable medical devices making their production critical. Software involved in the manufacture of medical devices is subject to scrutiny and requires stringent systems for traceability and quality. Global software development is also subject to this regulation introducing another aspect of complexity to the task. In this research we examine what effect this regulation might have on the global software development process.

1.2 Research Objective and Questions

In light of global software development being adopted in regulated environments and the lack of research in this area this research poses the question:

RQ: *How does regulation in medical device manufacturer's impact on global software development?*

This question leads to the research objective:

RO: *Provide a model for global software development applicable to a regulated medical device manufacturer engaged in internal software development.*

This model would identify areas that need attention toward achieving successful GSD projects in this context. In particular the effect of regulation on GSD is explored. Existing research carried out by Richardson et al. (2012) creates a Global Teaming Model (GTM) that is reviewed against the research data for validity in this context and potential adjustments where appropriate.

1.3 The Significance of the Research

There has been extensive research carried out in the area of global software development (Carmel and Agarwal, 2001, Herbsleb, 2007, Prikladnicki et al., 2004, Richardson et al., 2012). However the context for these studies does not consider regulated environments and there is very little research in this area. Regulation has a big influence over the software development process used in medical device manufacturers. The impact of failing to adhere to regulation is detrimental to the business. Therefore it is important that regulation is recognised and managed in a GSD setting and that management are supported in this endeavour.

Chapter 2: Literature Review

2.1 Introduction

As a means to understanding global software development (GSD) existing research in the area is reviewed in this chapter. A definition of GSD is provided to clarify the focus of this research. How and why GSD occurs is covered under configurations and drivers. The benefits of GSD drive its implementation however it also creates many challenges that threaten its potential for reward. Identifying these challenges means that they can be planned for and alleviated if not avoided. Recommended solutions are included for this purpose. Regulation of medical device manufacturing is investigated and how this affects the creation and use of software by medical device manufacturers. Finally regulation and GSD are tied together to determine how they influence each other.

2.2 Global Software Development

Global Software Development (GSD) is distributed software development where team members are distributed worldwide (Prikladnicki et al., 2006). It is multisite and multicultural (Herbsleb and Moitra, 2001). Distributed software development can be within the same country making GSD a distinct scenario. It is common to discuss virtual teams when exploring GSD as this is often how they operate (Noll et al., 2010, Jalali et al., 2010, Carmel and Agarwal, 2001). However it is also possible that some members of the GSD team are collocated thus not making them a fully virtual team (Oshri et al., 2007). A virtual team can be defined as a team that must consist of individual team members with interdependent tasks who are geographically dispersed and rely on technology for communication in the absence of face to face meetings (Gibson and Cohen, 2003, Noll et al., 2010). Being geographically dispersed does not necessarily mean across countries so again virtual teams are different to GSD teams. The literature often distinguishes global virtual teams when they are globally distributed and rarely meet face to face (Maznevski and Chudoba, 2000,

Edwards and Sridhar, 2003). A global virtual team has been defined to “connect people across organizational units whose policies, systems, and structures may not mesh together easily.” and distinct to virtual teams in this regard (Pinjani and Palvia, 2013, p. 144). Maznevski and Chudoba (2000) add that their task is a global one with a global strategy.

2.3 Global Software Development Configurations

Global software development configurations or business models are strategies used by companies for competitive advantage (Szymanski and Prikladnicki, 2007). The first choice for configuration is whether to outsource or insource software development. To outsource is to turn “over all or part of an organizational activity to an outside vendor”. (Barthelemy, 2003, p.87) compared to insourcing by keeping IT entirely inside the organisation and having total ownership of it (Dibbern et al., 2004).

Next is the location. Offshore is a term commonly used in the GSD context to refer to development in other countries (Carmel and Agarwal, 2001, Prikladnicki et al., 2006). It can be split in to two types. Farshore refers to countries that are geographically further away thus having long travel times to reach, greater differences in time zones, culture and language. In contrast nearshore are closer in distance with similar time zones, culture and language (Carmel and Abbott, 2006). Onshore, from the same country further reduces this distance and is the closest to the customer (Galvina and Smite, 2012).

Both external and internal sources of IT have their pros and cons. In an effort to get the best of both worlds companies have set up development centres offshore hiring locals as if outsourcing but keeping the centre under the parent organisation (Rao, 2004). Others refer to these as captive centres (Oshri et al., 2009) or the configuration as “offshore-insourcing” or “global insourcing” (Chakrabarty, 2006).

This thesis focuses on offshore insourcing of GSD where the organization develops its own software with both the client and customer internal but

globally distributed. Also known as internal offshoring (Prikladnicki et al., 2007).

2.4 Global Software Development Drivers

Organisations engage in GSD for the potential benefits it offers but also as it may arise due to a merger or acquisition (Ågerfalk et al., 2008). The top benefits cited for pursuing GSD are reduced costs, round the clock working, access to a larger skilled labour pool, closer proximity to market and customer (Ågerfalk et al., 2005, Ebert and De Neve, 2001, Herbsleb and Moitra, 2001, Damian et al., 2003).

Reduced costs are achieved through cheaper salaries and low cost economies found in countries such as India (Kobitzsch et al., 2001), Brazil, Russia and Ireland (Prikladnicki et al., 2004) to name a few. Leveraging time zones enables ‘follow the sun’ development to extend the working day to 24 hours (Holmström et al., 2006a). Teams at one site hand over to the next at the end of their working day and the start of the others so that work is continuous. The promised benefit is reduced time to completion (Rao, 2004). In some instances there is a local shortage of skills so going global provides necessary resources (Ramesh and Dennis, 2002). Closeness to customers enables better communication and reduces the misinterpretation of user requirements that can occur when distance hinders communication (Damian and Zowghi, 2003). Interestingly it is the same distance that causes issues for GSD teams (Carmel and Agarwal, 2001). Having a local presence close to the customer or market also allows for localisation where systems need to be customised to meet local needs (Cherbonneau, 2005) and creates good will from the local investment (Holmström et al., 2006b).

Conchúir et al. (2009) add innovation and shared best practise, and cross site modularisation of work to the list of benefits. Indirect organisational benefits can also be gained but are not often noticed. These are improved resource allocation, team benefits, such as reduced coordination cost and improved team

autonomy, and process benefits, such as improved documentation and clearly defined processes (Ågerfalk et al., 2008). Reduced coordination cost is realised when team members are not working at the same time and therefore do not need direct coordination (Espinosa and Carmel, 2003a). Again this is a benefit that is also an issue for GSD (Ågerfalk et al., 2008).

When GSD is kept in house the potential benefits are the same with the advantage of retaining control at all sites (Prikladnicki et al., 2007).

These rewards, while attractive are only perceived and are not always achieved unless the challenges that GSD creates are adequately addressed (Conchúir et al., 2009).

2.5 Global Software Development Challenges

The benefits of GSD come at a price. The distributed nature of GSD introduces difficulties for communication, control and coordination caused by distance (Carmel and Agarwal, 2001). Contributing to these difficulties is culture, both national and organisational that cause separation through different practises, language barriers and social norms (Carmel and Agarwal, 2001). Issues caused by distance can be categorised as geographical, temporal and socio-cultural (Ågerfalk et al., 2005).

2.5.1 Communication

Temporal distance reduces or eliminates real time communication and understanding, increasing the risk of misinterpretation and delays in completing work (Holmström et al., 2006a). As little as 30 m distance has been shown to cause a significant drop in the frequency of communication (Kraut et al., 1988). Informal or ad hoc communication is also hindered by distance which is a problem as it is credited for cross site coordination (Grinter et al., 1999), building relationships amongst developers (Herbsleb and Mockus, 2003), shared knowledge and understanding (Jalali et al., 2010), awareness of

other sites (Damian and Zowghi, 2003) and improving the speed of information exchange between sites (Nissen, 2004). Willingness to communicate between sites is low where trust is low which can be for several reasons such as job security (Mockus and Herbsleb, 2001). Lack of trust also inhibits knowledge sharing (Pinjani and Palvia, 2013) which in turn impacts collaboration which is essential in globally distributed teams (Kotlarsky and Oshri, 2005). Communication impacts both coordination and control (Carmel and Agarwal, 2001).

2.5.2 Control

Carmel and Agarwal (2001, p. 23) define control as “the process of adhering to goals, policies, standards, or quality levels”. Control over source code and documentation across sites is a risk to quality, with so many people involved and the number of changes that are occurring (Karolak, 1999). Misra and Fernández-Sanz (2011) find that it is difficult to have control across sites and that control is impacted by lack of communication. Control is tightly linked with coordination (Carmel and Agarwal, 2001) and often discussed as part of coordination in the literature (Fenema, 2002). It appears that coordination is required to achieve control.

Control requires reporting and management processes to ensure projects are progressing therefore relates to project management (Holmström et al., 2006b, Ralyté et al., 2008). Ralyté et al. (2008) recognise that distributed projects suffer from poor “visibility of project progress” and that issues at remote sites are not always reported or are ignored in the hope that they will be resolved without intervention. It is clear that project management is significant to maintaining control in distributed environments but must take the distributed nature into account (Richardson et al., 2012).

2.5.3 Coordination

Espinosa and Carmel (2003b) define coordination as “the management of dependencies” in a task” and find that coordination increases cost in GSD. This is the cost of communication tools and also in time due to delays caused by

miscommunications. Coordination is more difficult and costly where there is less overlap in time across sites. Reduced frequency of communication and familiarity with remote sites also impact coordination (Espinosa et al., 2001). Software products that are developed across sites need to be integrated which is dependent on coordination to be successful (Espinosa et al., 2007). Therefore the more dependency there is the more coordination is necessary. Coordination can also be strained where companies move from local to global development but use the same processes and practices without adapting to the new environment (Bosch and Bosch-Sijtsema, 2010).

2.5.4 Culture

Culture is recognised as a complex factor with several forms impacting GSD. These are “organisational culture, national culture and language, politics, and individual motivations and work ethics” (Holmström et al., 2006a). Differences in any of these cultures can lead to misinterpretations and conflicts (Kotlarsky and Oshri, 2005). For example in some Far Eastern countries women are not seen as equal to men which causes problems for female managers (Casey, 2009) and subsequently effective projects. Organisational differences include software development processes and project management methodologies (Carmel and Agarwal, 2001). These cultures need to be identified and compared so that plans to address them are put in place.

2.5.5 Trust

There are many studies on the implications of trust in virtual teams. It is one of the main issues that determine success in virtual teams (Jalali et al., 2010, Martins et al., 2004) and consequently GSD as it is the basis of everything else working. Lack of communication and coordination impacts trust and vice versa but implementing too many processes around these also diminishes trust (Moe and Šmite, 2007). Trust is positively associated with knowledge sharing (Jalali et al., 2010) so is imperative to GSD. Trust is fostered in collocated environments where face to face meetings, socialisation and active communication are possible and therefore is threatened by the distance introduced by GSD (Moe and Šmite, 2007). Studies of successful collocated

teams have proven this showing that when these teams become distributed trust is lost despite its previous existence (Boland and Fitzgerald, 2004, Casey, 2010). Fear is an inhibitor to developing trust and cooperation as well as motivation (Casey, 2010).

Other challenges in GSD stem from the differences between sites that are trying to work together such as infrastructure, tools and processes (Mockus and Herbsleb, 2001). These issues in GSD slow down software development delaying the completion of work (Herbsleb and Mockus, 2003). In projects involving complex tasks it has been found that distance has a detrimental effect on time, cost and quality (Bianchi et al., 2003). However these are well known issues that can be mitigated through coordination and planned processes (Richardson et al., 2012, Deshpande, 2012)

2.6 Global Software Development Solutions

2.6.1.1 Reduce Intensive Collaboration

Carmel and Agarwal (2001) suggest reducing intensive collaboration by giving sites ownership and making tasks more independent. This in turn reduces the need for continuous communication and therefore the issues that can arise from that. Tasks can be made more independent through modularisation and assigning these independent modules by team as per Conway's law. Which adds structure that supports the coordination of development work (Herbsleb and Grinter, 1999). It has also been found that feature based development teams reduce field defects (Ebert et al., 2001). Ebert and De Neve (2001) advise that when work is split by feature each feature is assigned to a colocated team that is dedicated to that task but that the overall project can be distributed globally. They outright disagree with Karolak (1999) solution of using virtual teams and go so far as to suggest that these colocated teams are within the same room.

2.6.2 Reduce Cultural Distance

The use of a bridgehead that resides onshore with the customer and translates to offshore development sites to reduce miscommunication and reassure customers can be used to reduce cultural distance (Carmel and Agarwal, 2001). Carmel and Agarwal (2001) also propose language training, the use of a cultural liaison and internal development to reduce organisational cultural differences as processes and company data are the same. Casey and Richardson (2006b) also found that a common organisational culture contributed to the success of offshore software development within the same organisation.

2.6.3 Reduce Temporal Distance

Carmel and Agarwal (2001) credit synchronous communication with reduced miscommunication and issues and therefore advocate working within similar time zones to allow it. They also note that this rules out the ‘follow the sun’ work model. This may actually be another advantage depending on the work involved. Intel who are well experienced in software development find that a ‘follow the sun’ approach is not suitable for software development but is useful for defect resolution (Holmström et al., 2006a).

2.6.4 Trust and Personal Considerations

Coordination is also improved through work familiarity, meaning that team members have shared knowledge of the same tasks and therefore a common ground for communication (Espinosa et al., 2001). Keeping with familiarity coordination has been found to be easier where social ties existed across teams (Hinds and McGrath, 2006). How people interact and team dynamics comes through in the literature as the cause and solution to many of the GSD issues.

In addressing trust issues Pyysiäinen (2003) proposes meetings between the distributed teams that outline roles and responsibilities and enable familiarity. Knowing roles and responsibilities makes it clear who to contact and encourages communication. Also important is the visibility of the progress of

the project and feedback on team member's contribution. In their research visibility of other sites and shared experiences encouraged the teams to help each other and boosted working morale and motivation. Clear expectations should be set from the outset as failing to do so can damage trust which impacts the effectiveness of the team (Jalali et al., 2010). Continuous learning is said to increase motivation and reduce employee attrition (Ebert and De Neve, 2001). Motivation is also better when the remote sites have responsibility over their own work (Bosch 2010).

2.6.5 Face to Face Meetings

Face to face meetings are most useful at the start of a project for establishing relationships, trust, work patterns and improving communication (Sudershana et al., 2007). They are highly recommended when collaboration is high such as during the design phase of development (Herbsleb and Mockus, 2003). Physical meetings alleviate geographical distance and create trust and a sense of belonging and “teamness” that help teams work better together (Holmström et al., 2006b). They also provide the opportunity for one-to-one communication so people can get to know each other (Casey and Richardson, 2006a).

2.6.6 Process Maturity

Process maturity levels impact project management in GSD (Casey and Richardson, 2006a). Increased maturity levels contribute to effective project management (Ebert and De Neve, 2001). Prikladnicki et al. (2007) recommend starting with smaller less complex projects where it is more practical to refine processes and standards and deal with challenges before scaling up to bigger projects.

2.6.7 Technology Tools

Several tools have been suggested to aid issues in communication, collaboration and coordination. Instant messaging is useful for synchronous communication and has the added benefit of showing a person's availability as do shared calendars which can be used to time communication (Herbsleb and

Mockus, 2003). Asynchronous communication is over email and while not the most efficient means of communication or always effective toward understanding does leave a record, is an opportunity to provide detailed explanations and can help understanding if there are language differences (Damian and Zowghi, 2002). Video conferencing enables the teams to see each other creating awareness of the other team (Damian and Zowghi, 2002). A software configuration management tool is recommended for maintaining control (Prikladnicki et al., 2007) and coordination and should be common across sites (Lings et al., 2007). These tools are only useful if they are maintained with accurate and current information to ensure user engagement (Layman et al., 2006).

2.6.8 Agile Methods

Despite the problem caused by distance to informal communication (Grinter et al., 1999) it is possible that methodologies such as XP, dependent on informal communication can be implemented in GSD as discovered by Layman et al. (2006) if certain critical success factors are addressed. They identified a customer role that had autonomy to make decisions, was available to developers and a stakeholder in the project, a bridgehead to relay communication and act as a technical and cultural liaison, distributed mailing lists and the use of a global project management tool to monitor daily project status and improve control and planning. Holmström et al. (2006b) also looked at agile methods for alleviating distance in GSD finding that agile methods could be implemented in part where they suited the circumstances of GSD. Not every agile method was suitable so was simply excluded, for example a 40 hour work week was not feasible where work was conducted across time zones. The most effective methods to address communication, coordination and control were found to be XP and Scrum.

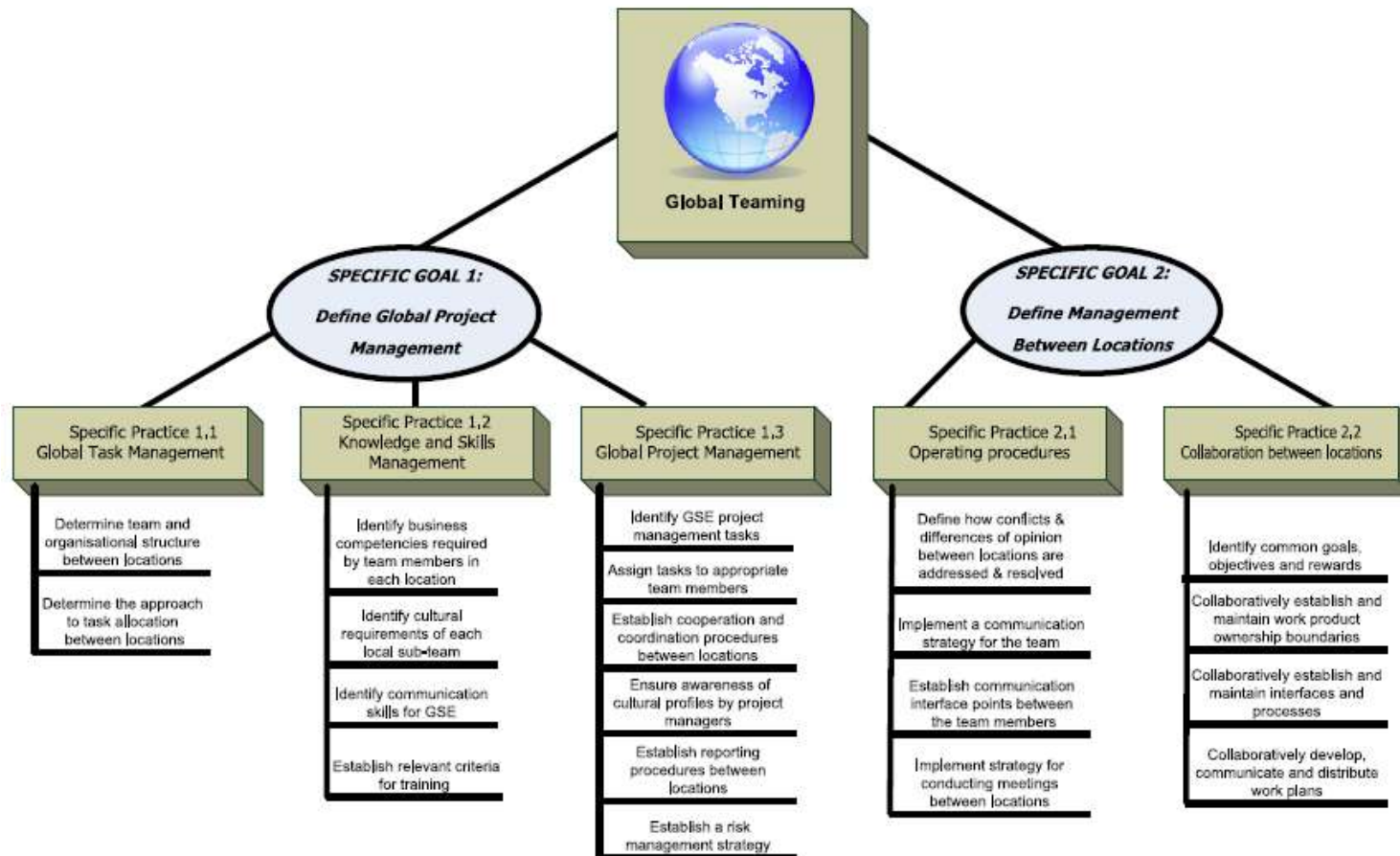


Figure 1 Global Teaming Model (Richardson et al., 2012)

2.6.9 Global Teaming Model

Of particular interest is the Global Teaming Model (GTM) proposed by Richardson et al. (2012) as it covers a comprehensive range of challenges imposed by GSD and specifically addresses the prevailing team issues in this scenario. Figure 1 illustrates this model clearly grouping areas of relevance and breaking down the required actions for establishing an effective GSD team.

The objective of this model is to provide a check list for managers to support the GSD initiative to ultimately make it a success. 25 factors that affect GSD as listed in

Table 1 were identified in this study through literature, case studies and comparisons to the Capability Maturity Model Integration (CMMI®).

Table 1 Global Software Engineering Factors (Richardson et al., 2012)

Communication	Skills Management	Language	Tools	Fear
Communication Tools	Knowledge Transfer	Motivation	Culture	Trust
Temporal Issues	Define Roles and Responsibilities	Technical Support	Teamness	True Cost
Effective Partitioning	Team Selection	Coordination	Visibility	Reporting Requirement
Project Management	Risk Management	Cooperation	Information Management	Process Management

Deshpande (2012) offers an expansive model for coordinating GSD named the GSD-COORD Model. It is referred to as an approach to project management including processes and strategies that can be tailored to meet global and organisational settings. This model also includes management of the client vendor relationship where GSD involves outsourcing.

Due to the breakdown of goals into action lists provided by the Global Teaming Model and its project management orientation it has been chosen over the GSD-COORD model for this research.

2.7 Regulation in Medical Device Manufacturing

Due to the safety critical nature of medical devices with their potential to harm a patient, including loss of life, they are highly regulated and tightly controlled (McAllister and Jeswiet, 2003). The Medical Device Directive (MDD) in Europe provides this regulation under Directive 93/42/EEC. EU member states are responsible for ensuring compliance to this directive in order to put devices on the market in their country (European Council, 1993). Each state has its own authority e.g. Irish Medicines Board (IMB) and the Ministry of Health (MoH), Italy. In the United States the FDA (U.S. Food and Drug Administration) manages this responsibility (Burton et al., 2006). Notified bodies are designated by these country specific governing bodies to execute audits and provide certification of compliance to medical device manufacturers to allow the marketing and sale of a device (Kaplan et al., 2004). While the market for medical devices is global the regulations are not. Each country has its own variation. This adds extra device development time and cost to ensure that each and all standards are complied with to market and sell a product globally (McAllister and Jeswiet, 2003).

The International Standards Organisation (ISO) provides for the manufacture of medical devices under ISO13485. This standard calls for a quality management system in the manufacture of medical devices. The NSAI (2009) refer to this as “a system for minimizing risk and maximizing opportunity”. The emphasis is on designing device and manufacturing processes that reduce or eliminate risk to the patient (European Council, 1993). Traceability of requirements (Regan et al., 2013) and of design history records form part of this (Mc Caffery and Dorling, 2009).

2.8 Regulation in Software Development for Medical Devices

Software used by medical device manufacturers is subject to the same regulation requirements as its manufacturing is when it is used in the manufacture of a device, implements the manufacturer’s quality system or it

forms part of the device itself (FDA, 2002). When the software forms part of the device or is classified as a medical device itself it is subject to more regulation, for example requiring CE Marking as per the MDD requirements (Klümper and Vollebregt, 2009). This software as a medical device is the subject of most medical device software papers rather than the other types (Cawley et al., 2010, Casey and Mc Caffery, 2011) but the conclusions are still applicable.

Whether developed by the medical device manufacturer itself or a third party software provider it is subject to the same rules of regulation and compliance is the responsibility of the medical device manufacturer (Klümper and Vollebregt, 2009, FDA, 2002).

Little research pre 2010 exists around software regulation in the medical device industry. However Lero, The Irish Software Engineering Research Centre, have established the Regulated Software Research Group (RSRG) and the Medical Device Software Engineering Group to address this (Lero). This is evident in the volume of recent papers that exist in this area attributable to researchers from these groups (Cawley and Richardson, 2010, Cawley et al., 2010, Casey and Mc Caffery, 2011, Cawley et al., 2011, Mc Caffery et al., 2012, Mc Hugh et al., 2012). Many studies of medical device software development are related to implementing agile methods (Mc Hugh et al., 2012, Cawley et al., 2010, Vogel, 2006). This may be due to the fact that agile favours working software over documentation (Fowler and Highsmith, 2001) seemingly contradicting regulation's requirement for documentation. Also the fact that traditional waterfall models of software development are inflexible to changing requirements (Mehrfard et al., 2010) and cumbersome for software developers to work with (Spence, 2005).

Software in the medical device industry is not alone in regulation. Other industries where regulation also covers software are Aviation, Automobile, Railway, and Nuclear due to their safety critical nature (Cawley et al., 2011) . The financial industry is also regulated for integrity to prevent fraud and theft (Coates, 2007). In the US publicly traded companies must adhere to the

“Sarbanes-Oxley Act of 2002 (SOX) which governs the processes for financial reporting, and therefore the systems and applications which contain and could affect the financial data.” (Cawley and Richardson, 2010, p.2).

Ingolfo et al. (2013) look at software requirements for establishing compliance from the outset. The functionality of the software must comply with regulation as well as the methods and processes used to produce it. One such requirement is that imposed by the FDA (1997) under CFR 21 Part 11 to provide for electronic records and electronic signatures. This requirement relates to traceability. As with medical devices, traceability and quality are paramount for software development in this industry and necessary for regulatory approval (Mc Caffery et al., 2012). Both Mc Caffery et al. (2012) and Regan et al. (2013) emphasise traceability in their creation and implementation of Med-Trace, a medical device traceability software process assessment method. Their aim is to develop a method that complies with the ambiguous plethora of guidelines and standards around software development in the industry that require differing levels of traceability at different points in the process.

The difference between guidelines and standards is that the standard must be met to achieve regulatory compliance whereas the guideline only suggests how this might be achieved and is open to interpretation (Regan et al., 2013). The FDA provide guidance for software validation to meet the Quality System regulation under their “General Principles for Software Validation” which are “applicable to the validation of medical device software or the validation of software used to design, develop, or manufacture medical devices” (FDA, 2002, p. 1). However the standard adopted by the industry is ANSI/AAMI/IEC 62304:2006 for medical device software lifecycle processes (Regan et al., 2013). The detail in both is general and while they require a software development lifecycle to be adopted and followed they are not specific on which one (Mc Hugh et al., 2012).

In summary the FDA’s General Principles for Software Validation (FDA, 2002) covers “planning , verification, testing, traceability, configuration

management, and many other aspects of good software engineering” (FDA, 2002, p. 1). It requires the manufacturer to:

- Implement a software development lifecycle (SDLC).
- Document plans and procedures for validation.
- Provide evidence that the software is validated through planned efforts throughout the SDLC.
- Assess changes for impact on the existing system and validate accordingly to maintain a validated system.

Validation is achieved through rigorous verification in the form of testing and analyses and confirms that the system is fit for purpose and fulfils its intended user needs (FDA, 2002). The level and type of validation corresponds to the level of risk involved (Rakitin, 2006). Burton et al. (2006) highlight the importance of managing risk throughout the software development lifecycle to meet regulatory approval. They propose a Risk Management Capability Model (RMCM) to extend the Risk Management (RM) process area of the well established process framework the CMMI® to address another area in regulation where there are many and conflicting requirements for compliance.

The FDA (2011) call for the implementation of Good Manufacturing Practices (GMP) in the manufacture of regulated products. The Good Automated Manufacturing Practice (GAMP) covers GMP for automated systems validation and compliance (ISPE, 2001). GAMP covers software involved in manufacturing and document management systems amongst many others. It presents a V-Model (SearchSoftwareQuality, 2014) that matches documented verification referred to as qualifications to specifications along the SDLC. (See Figure 2) Specifications include a user requirements specification, functional specification and design specifications which match to the Performance Qualification (PQ), Operational Qualification (OQ) and Installation Qualification (IQ) respectively. The IQ as its name suggests covers correct installation of the software. The OQ and PQ cover acceptance testing (ISPE, 2001). The FDA (2002) also mention IQ/OQ/PQ as a useful way to validate software but since it is not a familiar system to many software professionals

does not use this terminology to describe its validation requirements. In any case, the objective to validate the system is the same.

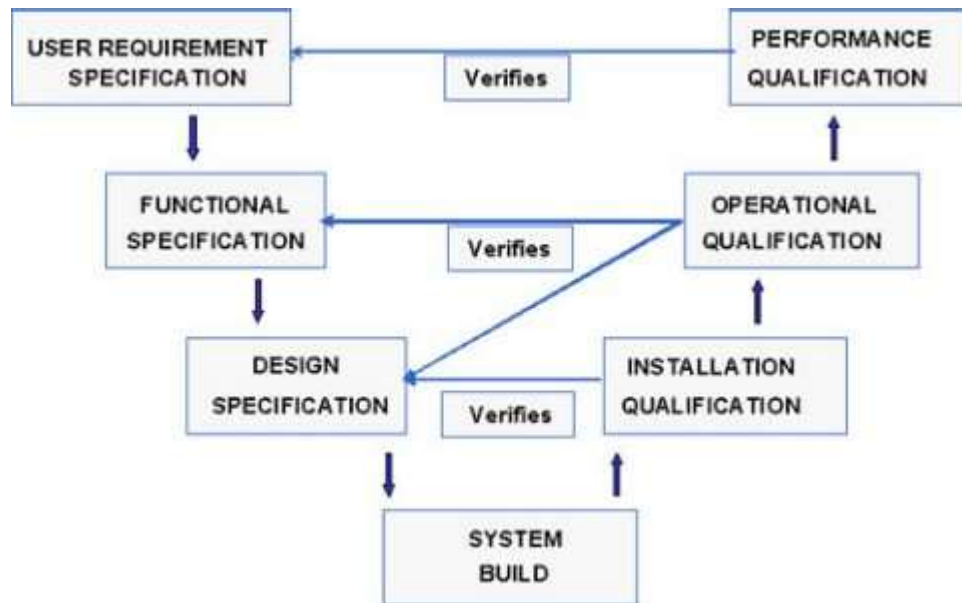


Figure 2 GAMP V-Model

As this model tells us, the software process is designed to meet the requirements of regulation. This is a key area in developing regulatory compliant medical device software evident in the research around software process improvement (SPI) in this industry (Mc Caffery and Dorling, 2009).

2.9 Regulation in Global Software Development

Regulation is not commonly a specific area of study in the context of global software development. As mentioned in the last section neither was regulation in software in general until recently. There are few sources to draw from on this specific subject. However the attributes of regulation such as quality, traceability, testing and risk are more prevalent therefore have also been considered here in an effort to gain more insight into the potential effect of regulation on GSD.

Cawley and Richardson (2010) found that regulation can be problematic within GSD and recommend setting a minimised common global process to address it. The regulation in this case is related to financial reporting and the SOX act as

mentioned earlier. Applying this research to other industries would be beneficial.

Sudershana et al. (2007) suggest that regulation should be used as an opportunity to be more efficient and systematic in global software development to avoid the complexity it introduces. This complexity comes from the FDA requirements for continuous risk evaluation and mitigation, validation and design process compliance.

In GSD the quality of the software produced has a higher risk of defects as the number of sites and temporal distance increases and where there is an imbalance in the distribution of developers across locations (Cataldo and Nambiar, 2009). This suggests that it is possible to plan GSD appropriately to reduce the risk of defects.

To maintain traceability a configuration management (CM) system is necessary to record and track all changes. Mc Caffery and Coleman (2007) have developed a CM process area for CMMI[®] that can be applied to global software development in a regulated environment. Synchronizing configurations in GSD adds complexity and difficulty (Bird et al., 2009) which could be an issue for regulation if it is not properly organized.

Bird et al. (2009) found that quality is more strongly influenced by organisational differences rather than geography. This is interesting and worth noting since other studies on quality on GSD only look at the issues that distance creates such as communication, collaboration (Gotel et al., 2012) and coordination (Misra and Fernández-Sanz, 2011) difficulties rather than other underlying issues.

Since regulation demands more thorough documented validation of software (FDA, 2002) and distributed teams are less efficient at validation than collocated teams (Ebert et al., 2001) there is an extra time cost for developing regulated software in a global configuration. Ebert et al. (2001) calculated a financial cost to a project of > 10% where validation activities were left until

later testing phases in GSD rather than collocated inspections early on in the project. Through this study on validation in GSD they identified eight lessons learned:

- Agree and communicate at project start the respective project targets, such as quality, milestones, content or resource allocation. Similarly, at phase or increment start team targets are adjusted and communicated to facilitate effective internal team management.
- Make teams responsible for their results
- While having one project leader who is fully responsible to achieve project targets, assign her a project management team that represents the major cultures within the project.
- Define at the beginning of projects which teams are involved and what they are going to do in which location. This includes a focus on allocation rules, such as scattering or collocation.
- Set up a project homepage for each project that summarizes project content, progress metrics, planning information and team-specific information.
- Collocate as much as possible teams to facilitate effective teamwork.
- Provide the necessary coaching on the job and free of friction by mixing different levels of expertise.
- Provide the necessary tools and technology to manage workflow and workspaces around the world (e.g. CM, problem management, test environments) (Ebert et al., 2001, p. 91).

These lessons reiterate findings from other studies in GSD to alleviate issues.

It is clear from the research presented here that global software development increases the risk involved in developing software. Ramasubbu and Balan (2007) propose that “high software quality process” be put in place to mitigate this risk.

2.10 Conceptual Framework

The conceptual framework presents the structure of the research. It is based on the information uncovered through the literature review and forms the focal point for this research. The research is set in the context of a regulated environment namely the medical device industry. Both industry regulation and the challenges that distance in GSD introduces are influencing factors on the team and processes that are necessary for successful GSD. The solutions suggested in the literature to address these pressures are included with a view to considering them in the final model for global teaming in a regulated environment. The drivers of GSD are discussed in the literature review to support the background to GSD but do not form part of the investigation so are excluded from the conceptual framework.

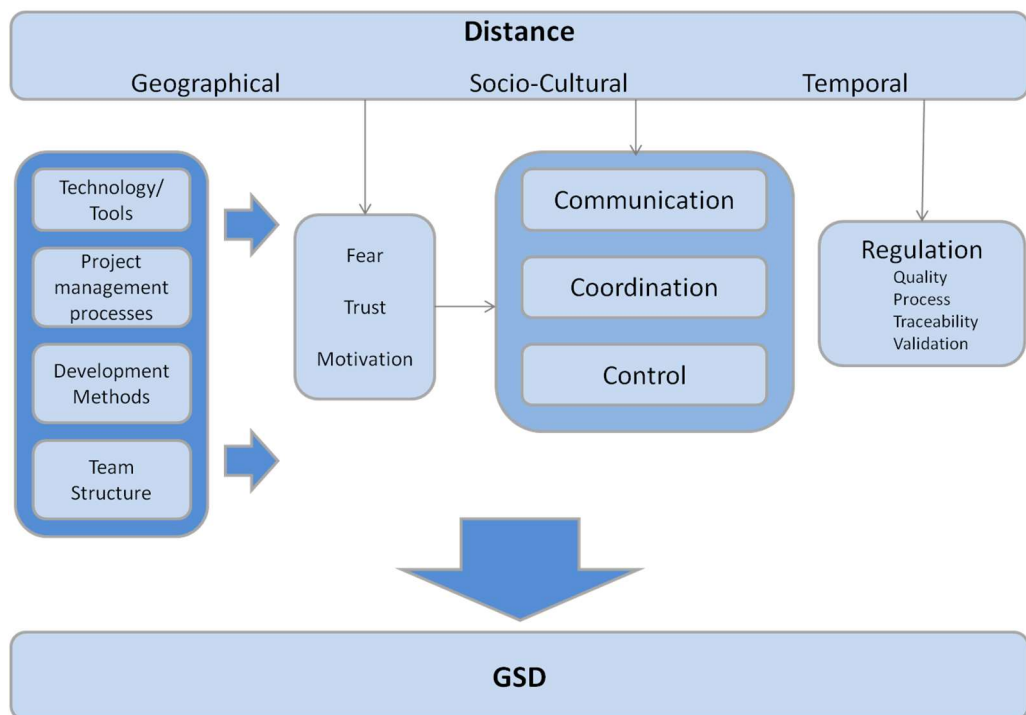


Figure 3 Conceptual Framework for GSD and Regulation

2.11 Conclusion

The overarching issues that affect GSD were presented in this chapter. It is evident from the literature that many of the challenges introduced by GSD are dependent on each other. For example lack of trust impacts communication

which in turn impacts knowledge sharing and collaboration compromising the success of GSD projects. It seems that it should be possible to resolve or at least alleviate some of these by addressing the overarching issues.

The gap in the literature on GSD in relation to regulated environments led to the research question; how does regulation in medical device manufacturer's impact on global software development?

Chapter 3: Research Methodology

3.1 Introduction

This chapter looks at the theory of research. Research philosophies are considered to underpin the research approach to the area of study. Based on this a research strategy is selected and research methods are identified and assessed for suitability against the research objective. It is important to choose the right research method to achieve the research objective. The rationale behind the chosen research method is provided, followed by a detailed description of the research design and the steps taken toward ensuring its reliability. Finally recognised limitations of the method are acknowledged.

3.2 Research Questions

The aim of this research is to explore global software development in the context of a regulated environment, in particular within the medical device industry to answer the research question

How does regulation in medical device manufacturer's impact on global software development?

A case study situated at Medi Inc, a medical device manufacturer provides such a regulated environment in which to base this research. Specifically, the objective is to determine if the advice of the current literature is applicable in the context of the case study and if there are any other considerations that may have been overlooked that could add to the body of knowledge.

Upon answering the research question the research objective is to compile a global teaming model for software development in the regulated environment of a medical device manufacturer.

3.3 Research Philosophy

Choosing an appropriate research method depends on the research objective and the research philosophy. Saunders et al. (2012) refer to a research

philosophy as the “development of knowledge and the nature of that knowledge”. As the subject of this study is concerned with how teams work in a distributed configuration it involves studying human interaction and behaviour. It is therefore a social enquiry. The main research philosophies are positivism and interpretivism. Positivism proposes scientific research methods for studying human action (Schwandt, 2001). It can be used for social enquiry if human behaviour is seen as “governed by law like regularities” (Ritchie and Lewis, 2003). Interpretivism disagrees with the positivist view and sees human behaviour as more complex and undefined by rules. It calls for an exploratory approach where the researcher is more involved and seeks to understand meaning in context (Rubin and Rubin, 1995). Hence this research follows the interpretivist philosophy as we seek to find meaning in human experience.

3.4 Research Methodologies

Research methodologies are influenced by the research philosophy and can be categorised as qualitative, quantitative and multiple methods (Saunders et al., 2012).

Qualitative research follows the interpretivist philosophy dealing with “soft” data that is not easily quantifiable (Fitzgerald and Howcroft, 1998). Or if it is quantifiable may not be useful in that form. It is concerned with process and meaning and is open to interpretation (Sale et al., 2002). On the other hand quantitative research methods deal with facts, hypothesis testing and statistics (Kaplan and Duchon, 1988). Multiple methods refer to a combination of qualitative and quantitative research approaches which they are said to complement each other (Seaman, 1999). Due to the suitability of qualitative research to social phenomena (Myers and Avison, 1997) and the importance of the social aspect in professional software development teams (Casey, 2010) it is very applicable to this research.

Research methods are the means by which data is collected and analysed to be used to achieve a research objective and answer research questions (Saunders et al., 2012). The research methods are influenced by the research methodology

(Myers and Avison, 1997). Since we are not testing a theory and the frequency of occurrence of a theme are not of concern to us rather the fact that it occurs at all and what the implications of its presence are, coupled with the social aspects previously mentioned a purely qualitative research approach was taken.

Qualitative research methods include action research, ethnography, grounded theory, narrative research and case study research (Saunders et al., 2012). Coughlan and Coughlan (2002) describe action research as an iterative process of planning, taking action and evaluating the action. The purpose of this study is not to test a theory but to explore a phenomenon so no action is taken. Ethnography can be described as participatory observation where the researcher watches interactions and events in an effort to experience the phenomena as it occurs (Ritchie and Lewis, 2003). This type of research requires the researcher to spend a significant amount of time in the field (Myers and Avison, 1997). This approach is not feasible for this study however the researcher is a past participant therefore has some limited observation that may be applicable. Grounded theory seeks to develop theory (Myers and Avison, 1997) and is driven by data rather than theory (Saunders et al., 2012) and therefore suited to this research. Due to the relevance of chronological events in the narrative inquiry method (Saunders et al., 2012) it was not suitable for this research which holds no significance to the order of events. Schreier (2012) describes qualitative research as case-oriented taking a holistic view of the situation and individuals of the study. This leads to depth and richness of data but is also time consuming. Case study research “focuses on understanding the dynamics present within single settings” (Eisenhardt, 1989).

Based on the interpretivism philosophy and qualitative research methodology and having considered the research methods it offers the case study research method was deemed most appropriate to achieve the research objective.

3.5 Research Strategy

Case studies are suited to “how” and “why” research questions and take a practical approach to studying a phenomenon within its real life context (Yin,

2009). This is relevant to the research questions of how regulation impacts global software development for a medical device manufacturer and how it could be managed. The why answers will enable the impacts if any to be addressed.

A case study can be both positivist and interpretivist (Darke et al., 1998). In this research it is interpretivist as there is no hypothesis testing or experiment. They can also be used to gather quantitative data as well as qualitative. Case study strategies can be single or multiple. A single case study is employed in this research which looks at the case of global software development within the regulated environment of a medical device manufacturer. Yin (2009) supports the use of a single case study if the case is unique or serves a revelatory purpose. Also if there is sufficient access to data. This case study meets these criteria with potential to glean rich data from experience. Case studies have been criticised for lack of rigour however this can be mitigated through a systematic approach (Yin, 2009). Flyvbjerg (2006) also argues in favour of the single case study describing it as “a contemporary approach” adding that it is supported by Harvard.

3.6 Research Design

Research began with a literature review covering global software development and regulation of medical device manufacturers specifically in relation to software development. Through this gaps were identified that led to the research questions. It also helped decide the approach to the research by highlighting the social aspects of software development as well as the demands of regulation within the environment. The guide questions that were used for the semi structured interviews were assisted by the Global Teaming Model found in the literature.

This case study is situated in a medical device manufacturer who develops software to support its business. Software development is globally distributed as is the business that uses it and it is subject to industry regulation. Thus the case is suitable for answering the research question. The unit of analysis for the

case study is the individuals working within the context of global software development teams within the regulated environment of a medical device manufacturer.

Primary data sources included in-depth semi structured interviews and company documentation. Empirical data was collected through these interviews conducted by the researcher for the sole purpose of this study. Company documentation consisted of existing documentation created in the past as part of global software development projects and quality documents that ensure regulatory compliance within the company. The literature review provided secondary data for this research. The use of multiple data sources allows for triangulation or more specifically “within-method” triangulation which is cross checking data sources for consistency and reliability (Jick, 1979). It also reduces the errors associated with individual methods when used alone (Patton, 1999).

Mixed method approaches combining quantitative and qualitative methods are often applied to get the best of both worlds. A quantitative survey may be conducted initially and followed up with qualitative interviews for further investigations. A survey follows a deductive research approach useful for generalising numbers, definitive responses and treating people as if they are alike (Rubin and Rubin, 1995). Based on that definition a survey was not seen to add any value in this instance and therefore was ruled out in favour of going straight to the interviews. The layout of the research design is illustrated in Figure 4.

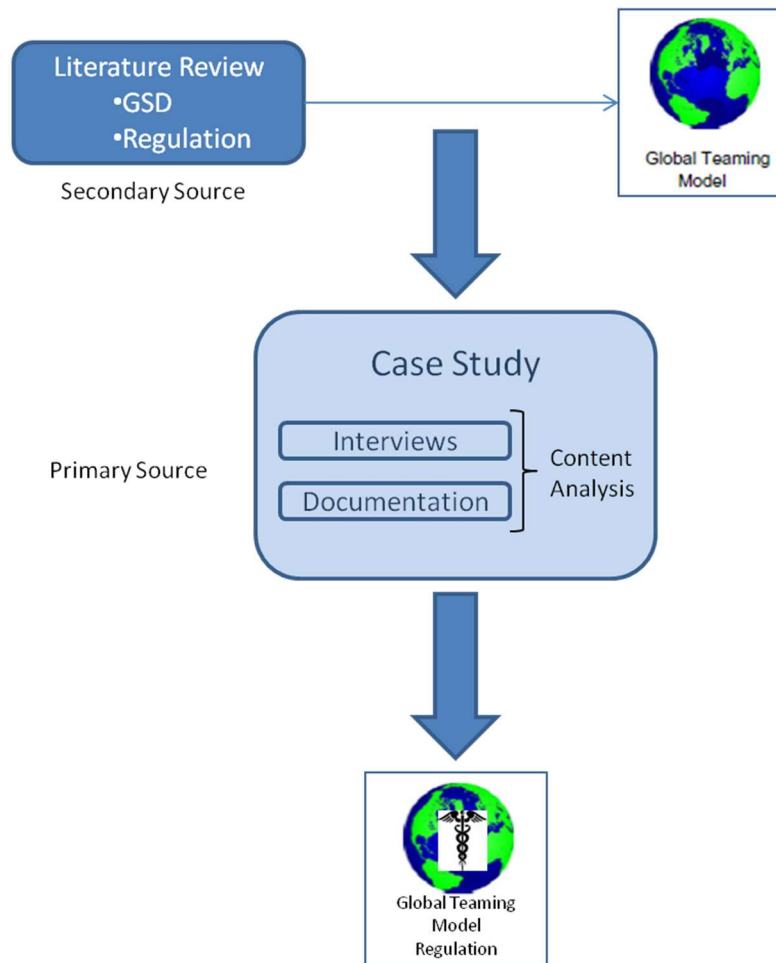


Figure 4 Research Design

3.6.1 Interview Protocol

Interviews were conducted with individuals who had participated in global software development projects. These projects were both regulated and non-regulated in order to determine if regulation made any difference to GSD. The participants for the interviews were opportunistic as the researcher had access within the case study but also selected through purposeful sampling. Purposeful sampling focuses the research on specific data rich cases in detail (Patton, 1999). Role, location and experience of varyingly regulated projects were selection criteria in order to get a broad view of the situation. This is in keeping with the premise that interviewees need not be the same and are chosen for their expertise or different perspectives (Rubin and Rubin, 1995).

Table 2 Interview Participant Profiles

Country	Role	Project Type
Ireland	IT Manager	Non regulated
Ireland	IT Manager/Developer	Regulated
Ireland	Global Project Manager	Regulated
Ireland	Software Developer	Regulated
US	Director	Regulated
US	Business Manager	Regulated
US	Software Developer	Non regulated
US	Software Developer	Regulated
Australia	Director	Both
France	Software Developer	Both

The interview participants as presented in Table 2 are involved in global projects on a daily basis but all except two, a software developer and global project manager also have local project commitments. The directors would be involved in more global projects as they represent their home sites in the global strategy.

10 interviews were estimated at the outset for sufficient data to be collected. This was based on qualitative samples only needing to record a theme once for it to be considered and collect rich detail (Ritchie and Lewis, 2003). This number matched the simple selection criteria of people involved in global software development within the case study. A more complex selection criteria or more diverse population in relation to the area of study would require a greater sample size (Ritchie and Lewis, 2003). In addition the time intensity of interviews meant that 10 interviews would be achievable within the scope of this research. Common themes emerged during the interviews with few new ones. However with such a small sample size it could not be said that a point of diminishing return could be reached at which no new information is gleaned from conducting additional interviews (Ritchie and Lewis, 2003) and therefore that saturation

could be reached (Glaser and Strauss, 2009). The researcher participated through observation and a reflective interview.

Interview candidates were contacted via email, phone and in person to request their participation. Everyone graciously agreed to be interviewed. Fortunately timing allowed for most interviews to be conducted face to face as some offshore staff were visiting around this time. Only one, with Australia was conducted over the phone. A pilot interview was conducted to begin with and from this improvements were made for the following interviews.

The interviewees were not prepared in anyway prior to doing the interview. They were simply told that the research was in relation to global software development when invited for interview. In this way it was intended that answers could not be prepared and perfected to meet some expected kind of answer. The answers should be instinctive and reactive therefore more honest revealing their true experience.

Each of the interviews followed the same general guidelines. They began with a brief description of the research area and each interviewee was asked to discuss a global software development project they had worked on. This enabled the researcher to get the context and structure of their GSD experience. They were then invited to provide three examples of things that worked on the project and subsequently three things that did not work well. These are an example of open ended questions that were designed to elicit detailed answers and to encourage interviews to freely discuss their experience. No dichotomous yes/no questions were included although they may have been used for the purpose of clarification where necessary (Legard et al., 2003). Closed questions were excluded to prevent leading and biasing answers (Legard et al., 2003). Questions were also directed towards the individuals own experience as Yin (2009) advises they should cater to the unit of analysis of the case study. An outline of the interview questions is provided in Appendix A: Interview Protocol. Included in this is a table for tracking the emergence of themes but not their frequency. As mentioned earlier the frequency of the occurrence is not relevant to this research.

Where there was not much information given in responses interviewees were asked to expand or explain on why they thought something was good or bad for example depending on the case and what could have been done to address it. Questions were also asked based on these answers and the Global Teaming Model to tease out the different aspects raised during the first half of the interview to allow the interviewee to expand and potentially mention other aspects or detail that they might not have previously thought of thus providing richer information for analysis. The interviewer having some knowledge of the systems and structures within the organisation was able to identify answers that were short on the actual situation and delve deeper into answers to discover the true experience. However the interviewer was aware of limiting the effectiveness of the interview and so endeavoured to avoid leading and giving unintended cues through note taking during the interview (Legard et al., 2003) and instead focused on listening, formulating follow up questions based on previous answers and interpreting body language as recommended by Mason (2002) (Legard et al., 2003).

The interviews lasted between 40 and 55 minutes on average. All interviewees consented to being recorded. Their anonymity was assured and these recordings were later transcribed for analysis. Notes were taken after the interview to support interpretation of the interview transcripts (DiCicco-Bloom and Crabtree, 2006). The interview transcripts were subjected to the content analysis research technique.

3.6.2 Content Analysis

Content analysis is a qualitative research technique that can be used to “describe the characteristics of content of document, make observations and provide analysis” (Alias et al., 2013, p.765). It seeks to infer meaning from content (Hsieh and Shannon, 2005). Typically it is intended for large amounts of data but can also be useful in single case studies (Slowey and Richardson, 2006) .

Codes are used to categorise themes emerging from documents. Initial codes were compiled from the literature review in line with directed content analysis that focuses research based on existing theory as a guideline (Hsieh and Shannon, 2005). These were refined and amended as new themes emerged. The codes used to identify themes are listed in Appendix B: Codes for Content Analysis.

Content analysis for this research consisted of reading the interview transcripts and documentation through initially to get an idea of the content and then reading again several times highlighting topics and assigning categories to the content. Summaries were also written along the way to keep track and gather ideas. In this way the researcher was able to breakdown the textual content into identifiable topics relevant to GSD.

3.7 Limitations of the Research

The use of a single case study may be seen as a limitation. Indeed the use of multiple case studies for comparison to support or challenge the research findings would strengthen the outcome. Using a small sample size and a single case study makes it difficult to generalise the results. However the intention is to further understand the field (Sale et al., 2002). Due to the time intensity of interviews and data analysis but also the fit and suitability of this case study to the research question the single case study was used.

Interviews are open to many issues and often questioned for their reliability. This is because they are dependent on human behaviour with the potential for poor memory and bias (Yin, 2009). Hence documentation was used as another source of data toward validating interview data. In this case trust was a consideration. There was a danger that interviewees would hold back on their answers for fear of backlash from criticising the company or the people that they work with. They were reassured of their anonymity, the confidentiality of the resulting work and the non specificity of answers. Interviews can be misleading due to their dependence on experience and interpretations but this can be mitigated through observation (Orlikowski and Baroudi, 1991). A longer study that allowed for

extended observation could yield more data however time was a limiting factor in this research.

Sample size can be an advantage as explained in section 3.6 but may also be a disadvantage. The larger the size the more potential there is to uncover more information.

Given the researchers familiarity with the case study and the research area there was potential for bias to arise. The researcher was aware of this and therefore second guessed and questioned findings to ensure that this was not the case.

3.8 Conclusion

This chapter provides detail of the investigations into finding a suitable research method to meet the research objective. An interpretivist philosophy was decided leading to an exploratory or inductive approach (Bottom up). Qualitative research based on a single case study was found to be the most compatible to answer the research objective. A literature review was conducted to identify areas of research that needed further investigation as well as guiding data collection in the case study. In depth semi-structured interviews were conducted and along with documentation related to GSD projects and regulation were subject to content analysis. This was used to uncover aspects of global software development and regulation within this context.

The limitations of the research methods applied are acknowledged along with reasoning for their validity. The researcher was conscious of the validity of the research throughout. To this end every effort was made to understand the research method being used and to identify and address as much as possible any weaknesses that it may have.

Chapter 4: Findings

4.1 Introduction

The findings from the case study at Medi Inc are presented in this chapter. The Global Teaming Model (GTM) (Richardson et al., 2012) is used to group findings as appropriate for presentation along with other findings that did not quite fit the model. The two specific goals of the Global Teaming Model are broken into their respective sub practices under which the findings are summarised. This serves to provide an understanding of the state of GSD at Medi Inc. How regulation fits in to GSD in the case study is then outlined. These findings go toward answering the research question of how regulation impacts GSD and achieving the research objective to suggest a model for global teaming applicable to a regulated medical device manufacturer engaged in internal software development.

GSD is a relatively new concept to Medi Inc. Historically the company consisted of independent sites that were effectively their own company. They had little in common apart from their parent company in the US, some initial processes and regulatory requirements. Some requirements and the main objective to manufacture and ship medical devices were the same but the implementation and processes that achieved this were left to the discretion of the local company. This is the legacy on which GSD is now based. The main software development and business sites involved in GSD at Medi Inc are located in Ireland, Denmark, France, Germany, US and Australia.

4.2 Global Teaming Model

4.2.1 Define Global Project Management

4.2.1.1 SP 1.1: Global Task Management

There are two types of GSD team or project at Medi Inc. The first is that which set out to be global having a global objective from the beginning and planned

accordingly. Requirements analysis and task allocation were given more thought and conducted collaboratively onsite with all team members present. They “*started off the right way, doing the analysis*” according to a developer. Contrary to this are the global teams that evolved to meet changing requirements as they arose and consequently lacked in planning and structure. Global projects are defined on a global list that is reviewed and prioritised by the board of directors. These projects have a global structure with a project and product manager assigned. The main purpose of the product manager is to communicate with stakeholders both business and IT at each of the sites. Their role comes into play at the beginning in gathering requirements and again towards the end at implementation time. The role of project manager is to act as one point of contact that liaises between the business and IT. They pick up from the product manager after the requirements are defined to resource the project, allocate tasks and set deadlines for deliverables.

Teams are function based, by locality each with a functional manager. The functional manager helps resource global teams. The project manager felt that a “*project orientated*” team structure would be more suitable but that it would be too much of a change for the company at present. He described the teams as “*forming and storming but not performing yet*”.

Development was divided into modules with each of the local teams developing a piece that was later integrated into the global software release. They had ownership over that piece of the software and were responsible and accountable for its delivery. Common file structures were agreed up front to enable later integration. Generally team sizes were balanced across sites except on the management side which was often more US oriented.

General roles and responsibilities are captured in the change request document that initiates a change. This includes a high level description of the change, the affected departments and their respective manager but is not comprehensive enough to allocate tasks. Roles and responsibilities have been hard to define across sites as to who actually owns a project or product. Sometimes there are too many stakeholders but no definite manager or project manager. Without a

project manager there have been occasions where people have been left out of projects only to realise they are needed later requiring a rethink on requirements and the project plan as a whole. Project deadlines were then delayed. These coupled with a lack of communication between the business and IT made for awkward situations that threatened the project. Confidence between teams was diminished as a result. A business manager in hindsight felt that *“everyone should have gotten in a room”* to apologise and move on. On the other hand it also created more communication over and back to try to figure things out and reach an agreement without a final decision maker. A business manager described it as *“a democracy”* which made it difficult and reckoned that there needed to be *“a tie breaker vote”*. This was deemed to strengthen the relationship between developers as they sought to progress the project themselves. Some developers mentioned that they shared a common desire to do a good job and deliver the software so as one mentioned they *“bonded together as developers to get it done”*. They were not interested in getting involved in what they called *“politics”*.

Task allocation in Medi Inc is based on skills and availability. Known experts in an area are assigned first but these are limited so are balanced with those of lesser experience. Resource managers globally work with those locally and the project manager to decide. They try to evenly distribute the tasks where possible.

4.2.1.2 SP 1.2: Knowledge and Skills Management

Experts are known between sites and attempts have been made to cross train others. To date this has been difficult with existing workloads. Teams close to customers are usually the preferred team for a project but these are usually local projects. In global projects the customer is often global but a local developer representing each site will be involved and look after their local customer's needs within the overall project.

Most projects would try to have face to face kick off meetings to elicit requirements. The social aspect of these meetings such as going out for dinner

and meeting informally was seen as very effective in building relationships. Personalities were also noted to help with one developer even inviting the team over to his house for dinner. This contributed to understanding the cultures within the global team. It was seen as costly but valuable at the start of the project which was usually the only time it occurred.

A communication protocol was not considered for any projects as such. Communication occurred as usual over email and phone. When time zones caused issues teams were happy to accommodate out of hours calls as these were only occasional and necessary to move the project forward. There was no formal arrangement around this type of communication.

There was no specific training for working on global projects. Training has been on an ad hoc basis at the discretion of the direct manager in relation to technical or business skills. Mostly it is “*on the job training*” (IT manager) and not formally organised. Cultural and linguistic issues had not been considered for training. There was no mention of any concern or issues in relation to the absence of training in any of these. However there were cultural differences with customers at a site in Germany that were not immediately obvious. They only emerged as they attempted to implement systems. In one case a developer was sent over to observe why the system was not working as intended. They discovered that the task the system performed was not being completed by the operator if it went past going home time. At other sites the operators would finish a task out before leaving. It was a subtle difference but caused a lot of confusion. They adapted to these differences as they transpired. They also placed a developer permanently on site for support and to liaise with the Irish for development.

4.2.1.3 SP 1.3: Global Project Management

Project managers work with the local resource managers to resource projects. They are aware of the contribution that each member makes and follow up with them on regular calls for progress updates as well as keeping in contact with the resource manager who also tracks employee progress and performance.

This contribution is not documented apart from status and progress updates by the project manager. Direct resource managers may include knowledge and skills development for annual appraisal purposes.

No cultural assessment is undertaken as part of global projects. There has never been anything “*that could be described as a cultural issue*” (IT Manager) therefore it has just never been considered. Cooperation and coordination procedures fall under the project plan incorporating reporting and where requirements need overlap between sites to be achieved.

Reporting procedures are in place in the form of weekly status meetings. These change as the project requires it. For example, when software is released these meetings become daily in the event of an issue. However there have been problems in reporting status to a remote project manager. Sometimes local demands have taken priority over global projects and the global project manager was not involved or even notified. It goes back to the old way of doing things and local reporting. This is more the case where the head of the department is located in that locality but it does mean that the global project suffers causing delays and frustration on the other global team members who are oblivious that work has stopped at another site.

Risk management comes under the quality system that medical device manufacturers are required to have. As part of the project plan risk is assessed and mitigation plans created. This does not cover all project risks. Its concern is primarily the level of impact to a patient. Other than that “*it’s an informal process*” (Director). Mostly informal checks were carried out as projects progressed. As seen in the previous example threats to time are not always managed correctly however the local executive would have informally assessed the risk before pulling resources. This does not appease the global project manager who is supposed to be responsible for the project’s delivery. As the global project manager put it “*You think they’re working on other things and they’re not*”. It is the global project manager who has to deal with the fall out later with the business customer. It also undermines the role of the project manager.

IT project managers are largely based in US headquarters with few in Europe tasked with the regulated manufacturing system. Business project managers are more common across the board. They aim to be one point of contact for control and consistency and also to reduce extra communication effort and the potential negative impact of having too many people involved.

Some projects did not have a project manager. In those cases a manager was in place but the role was different to that of an actual project manager. It meant that only status updates were required. One developer was not clear on his role “*Maybe somebody thought that I was supposed to do that (act as project manager)*”. There was no structure or proper deadlines to work to. This frustrated developers who liked to have more definition and guidelines on their work. It also impacted productivity allowing projects to drift.

4.2.2 Define Management between Locations

4.2.2.1 SP 2.1: Operating Procedures

Conflicts and differences of opinion were not formally addressed. There were no procedures to deal with this specifically for GSD. The usual local route through a direct manager was given by IT managers as the procedure to follow in the event of an incident. They also suggested Human Resources if the conflict was to escalate on a personal level. Difference of opinion was common due to the absence of global standards. Management often conflicted over the best approach to global, differing styles and agreeing business requirements. These would have been debated with each side arguing the merits of their approach and finally more senior management may have intervened to confirm global direction if need be. It did not appear to affect the working relationship after the event.

No written communication procedures existed. Formal communication for project management updates was planned. Usually weekly meetings occurred where everyone was included and then just between the project manager and

the business and then the developers to manage all stakeholders. The developers themselves also met on a regular basis. The project manager felt that the frequency of the meetings created an opportunity to keep people informed in the absence of informal meetings or if they had informal conversations locally they could update the global team before they were forgotten. These meetings were through conference calls, video conferencing or in some cases over WebEx sessions for desktop sharing. Mostly language was not an issue. There were a couple of occasions where it was not the first language of some team members so there was a little more clarification needed. Conference calls were particularly challenging for these people with several different people and accents to decipher. Minutes were sent out following the meetings but this was not consistent and depended on the project and individuals. Some communication has tended to be at individual level resulting in some members being left out and uninformed.

Delays in communication were attributed to time zones, local priorities and workloads. There are routes through managers or other local roles that were used to prompt responses. Accessing key people was a common problem as they tend to be involved in many projects so are always in demand. An IT manager highlighted the futility of emailing them:

“If I drop them a mail they won’t even get to read it because they have so much mail they’re not going to get to”

This increased in accordance with their level of seniority within the organisation resulting in a severe lack of availability of the decision makers. Even when they did get to talk to them time was limited so discussions and decisions were brief. Proposals or teasing out ideas often did not get enough attention as a result. The extent of contact can be summarised in the statement:

“You have their 10 second view and they say “no, not doing that””

(IT manager)

“You can’t just walk over to someone, you have to wait and try to figure out when they are available” is how a project manager described the problem of distance. They also mentioned that they would try to access these people through others in the same office. They might check the availability of their

counterpart or someone else they know that sits in the same office and ask them to either check if the person they need is at their desk so they can call them or get them to remind them to call the other site.

Communication became more of an issue when software changes were implemented on live systems. During this time reactive changes were made however these were not communicated to other teams.

Generally people knew who to contact when they had an issue or needed to report on project status. There were also plans in place setting milestones and points of integration where collaboration was required and with whom.

4.2.2.2 SP 2.2: Collaboration between locations

Common goals and objectives were established across sites on projects. However obtaining agreement was almost always troublesome. Each site had its own take on requirements, technologies and coding structures from its past as a standalone site. A director explained that

“If we all had the luxury of knowing we were going to be global at the beginning it would have been easier rather than allowing everyone to become engrained into a local way of thinking and then converting them into a global approach”.

On some particularly new and complex projects a common high level goal was agreed but the detail in achieving it was not and caused a lot of issues for the project such as difficulties in testing and implementation, trust and delays. This was as a result of the challenges these projects faced including changing requirements as the business was not certain of the full implications across sites, leadership changes and political agendas. It was suggested that someone would have to have the final decision to prevent a stalemate and that people would have to *“get over the fear that other areas will never understand how we do business”* (Business Product Manager).

The approach to projects also caused frustration. The culture at one site was to get started quickly once they decided they were going to do something. If it failed they would try something else. This code first approach without proper analysis of the requirements and solution conflicted with another site who favoured a more considered approach to *“do it right and do it once”* (Developer). Conversely this analysis was seen as a delay to starting a project from the other site that was keen to get moving on the project and make progress.

There were times when the definition of a *“global”* project was open to debate. This was when a US project was to be implemented at global sites or when *“the US has the last word”* (Developer). It was termed a global project but local sites did not feel like they were contributing but rather following instructions to implement a US solution. Since the head of the project was based at the US headquarters decisions were made there first, without collaboration with other sites. Changes in requirements were mentioned several times that did not filter down to other sites until they were already started on what they thought the project was supposed to be. Lack of collaboration on requirements then led to more frequent meetings and a dependence on communication to figure out what was needed. In the meantime the *“hard date”* (IT Manager) for delivering the software still held despite the late changes. This meant that meeting the deadline meant reducing functionality or changing approaches to provide the minimum to meet the overall requirement and adhere to regulation.

The company does not have a reward policy and acknowledgement of team success is not something that occurs unless informally from a direct manager. This may occur during annual employee appraisals that the local direct manager carries out.

Work was effectively and appropriately partitioned across sites. There was awareness that changes made had a global impact. Sometimes this was handled with care and other times not so much with assumptions made that other systems would be the same.

From a process perspective in line with the Global Teaming Model the ownership was placed with those closest to the process. This was the case with regulation which required a quality system to be followed. In the absence of a global quality system and hence a global software validation process a local one had to be used instead. While there was ownership it always prompted discussion and debate at the start of every global project to decide which site would own the process. Once decided then the team members from other sites would have to adopt that process for the duration of that project and use templates and controlled documentation from the chosen quality system only. Collaboration tools were recognised as necessary and the way to improve collaboration and visibility. However a global standard had yet to be agreed.

4.3 Regulation in GSD

4.3.1 Software Development Lifecycle

The software development process follows a waterfall model based on the GAMP V model as suggested by the FDA guidelines. All software development follows this path.

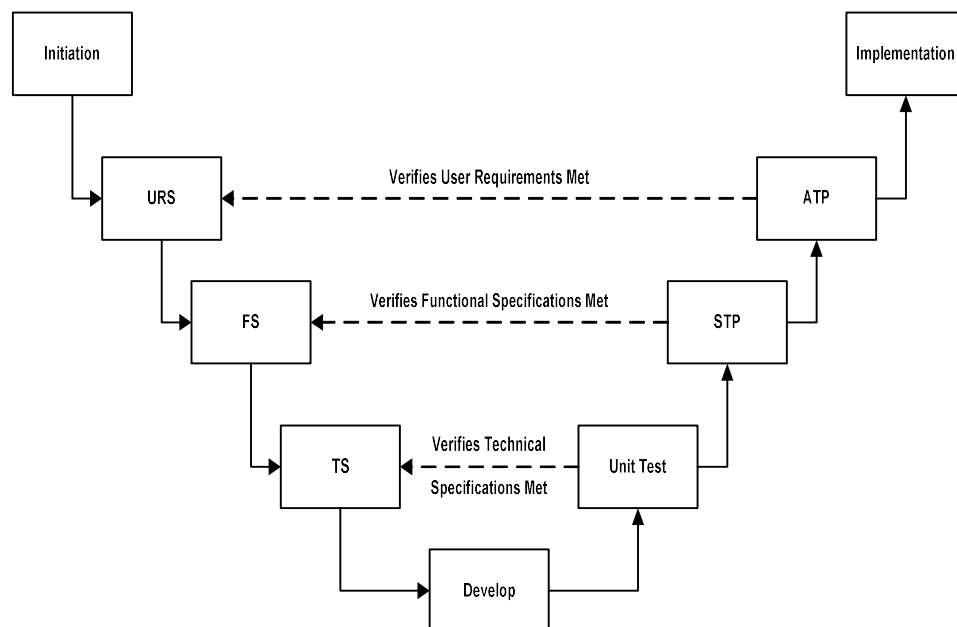


Figure 5 Medi Inc Development Verification Process using a V Model

4.3.2 Risk Management

At the outset a risk analysis is performed to determine the level of risk and determine the type of validation that is required. This is a “*formal process*” (Director) documented in a quality procedure that every project follows. The validation type indicates the level of detail and documentation that is required. The higher the risk the more extensive the validation and level of documentation required. Forms of risk are potential risk to a patient, regulatory risk and business risk. The same process is effectively followed at each site for both local and global projects but under different systems. If requirements change during a project then the risk must be re-evaluated and changes to the validation plan made as necessary.

4.3.3 Quality Management System

The differing regulatory bodies were not an issue as each manufacturer already sold into a global market. Regulatory requirements were compiled from each country into one specification that the company used to validate against. This is incorporated under the quality management systems that govern the development process. It ensured that they were compliant for every market.

The issue with regulation was the quality system that enforced regulation. Each site has its own quality system. This system sets out quality system procedures (QSPs) and standard operating procedures (SOPs) that must be followed to introduce software changes. These procedures define risk management, development and quality processes, validation activities and required documentation. Version controlled document templates for completing software validation is maintained at each site. Since there is no global validation process one of the local sites systems had to be selected for use in a global project. This was decided based on the business that initiates a change or the first site that will implement the change. A meeting is held at the start of every project, usually between the IT managers and global project manager to decide which route to follow. A global project manager described it as “*one of the big questions*” that is always asked on global projects.

Developers across sites on the project must then complete the others documentation. This can change between projects and be confusing. There are corresponding documents but they do not have equivalent names. For example at one site it is a QSP and at another it is a QSI. The content of the procedures and layout of the templates were collaboratively agreed across sites for alignment and cannot be changed unless under global consensus. This does help in following the process but the correct country's templates have to be used in order to meet the quality system demands. This process is not documented nor is the requirement for global consensus to change any of the local quality documentation in relation to software development. In fact none of the local quality documents made any reference to global projects apart from mentioning that the owner of the process as the Global IT Functional Leader along with the regional IT Director. Surprisingly there were documents with global titles but these did not refer to a global system as none existed. The US, being the headquarters adopted global into their document titles in relation to software processes and operations. The content of these "global" documents referred to local roles and systems.

There are still a couple of sites that are different from this global initiative. On those projects the resources have had a "*learning curve*" (IT Director) and some difficulty in following the process as they are quite different. An IT director explained the difference:

"At a high level it's similar but at documentation and functional level timing is a bit different on some things and the extra content of the documents are a bit different."

Differing systems has also caused concern and delay to know who has responsibility and authority to sign off on changes and processes.

One of the main differences across sites is how a change is initiated and the change control process needed to approve the change and introduce it at a site. Each site requires that this part conforms to their system. Sometimes it can be possible to leverage initial validation from the first site when implementing at other sites by initiating the local change control process, referencing the first site and justifying that the validation completed at the sister site and using their

controlled documentation and data is also appropriate to this site. This must be completed on local document templates to be acceptable. It is confusing switching between systems and referencing other systems and does spark conversation for every project. The document repository that stores this validation is also independent at each site so it requires access and navigation of the remote application.

The other issue with having to choose a local quality system is that it usually means that resources are aligned with the chosen process. This helps the smooth execution of the process but not necessarily the software development that needs to be achieved.

Software changes that were deemed high risk due to their potential to affect the patient require significant documentation and validation under regulation. Thus these projects were more difficult to manage at a global level. As a consequence changes to these systems were postponed unless critical. Another reason was the reluctance to accept accountability for such changes that if issues arose could seriously threaten the business.

4.3.4 Non Regulated Systems

Non regulated systems followed the same software validation processes at Medi Inc. However these were deemed to be low risk based on the impact to a patient if they failed so the level and detail of validation required was significantly reduced. This was supported by the light documentation available for such projects. Often these projects had no connection to the patient and were business process oriented. Routing these projects through the same system as the regulated ones was a company choice and not required by any regulatory bodies.

4.3.5 Customs Regulation

A different kind of regulation emerged through the interviews in the form of customs. This is regulation imposed by a country receiving the medical device

products and includes for example correct labelling, classifications, nomenclatures and duties. This was a bigger issue in Europe than in the US. In the US it is possible to ship to any state under the same single legislation whereas in Europe each member state is independently governed setting its own customs rules. This made projects more complicated for Europe. US counterparts failed to recognise this extra work and were primarily concerned with their own issues. This regulation did not impact processes or how the teams worked. The customs rules formed requirements for system functionality that could be included like any other requirements.

4.4 Other Findings

4.4.1 Tools

Each site favoured the use of differing tools. Some of this was historic but even in recent times in light of global initiatives alternatives between sites were common. This was due to the indecision and lengthy process of reviewing tools for global use. In the meantime another tool was picked that was suitable for the task at hand to avoid waiting. It often got adopted by other teams after this and became another system to manage. On global projects each site used their own as far as possible and had to make decisions on ones to share. This meant that people could be using several different tools for the same purpose depending on the project. Software configuration management and version control also was not common between sites but one was adopted for global projects out of necessity. For example Subversion was used with both command line and graphical user interfaces at different sites and Team Foundation Server at others. The developer would have to get to know each application so they could switch between them as they worked on different projects. On a positive note, a project manager could see an advantage stating that it meant *“you get to pick and chose the best”* from *“proven technologies”* when working across experienced sites.

Email was the typical tool for communication. Instant messaging was rarely used. Video conferencing was available but not regarded as very effective. WebEx was also used and praised by developers for the ability to share workspaces to better understand development issues and work through them in real time.

4.4.2 Team Dynamics

Team dynamics varied. One manager described their relationships *“like a family. Sometimes you get on fine other times you fight”* but they always resolve their issues.

Fear appeared when sharing code and due to lack of global direction. In some instances a site would be protective of code and would not aid others and share. This did not help collaboration or trust. At an individual level this was described as *“I don’t want anyone messing with my code”* (Director). It caused *“friction and frustration”* (IT Manager). A development manager believed that it was out of fear of finding fault with the code. The lack of global direction and definitive roles has led to a *“lockdown on sharing information”* and reduced motivation. This was also attributed to politics as people began to compete for more prestigious and seemingly powerful global roles. Fear caused resistance to change which was put down to individual personalities and mindset as to whether they were *“open minded and easier to get along with”* or set in their ways believing *“that that’s not the way I would have done it so it’s not the right approach”* (Director).

Sometimes local teams were actually individuals working remotely. This could be a problem when it was their only team leading to disconnect from local colleagues and feelings of isolation. A developer described it as *“socially weird to be separated from the people that you share the office space with”*. Others saw an advantage to being remote as avoiding getting caught up in *“office politics”* (Project manager).

4.5 Conclusion

This chapter summarises the findings from the research conducted in the case study. It appears that Medi Inc is at an early stage of GSD with many issues that need to be addressed. GSD was seen to be more time consuming but the benefits were worth it. The trade off was acceptable. Despite the inherent issues and frustrations with working like this the overall attitude was positive and in favour of GSD. There was unanimous agreement that there was need for improvement and that regulation did make things more awkward and time consuming.

The most significant impact on GSD was the organisations' GSD maturity i.e. global software projects are new to them. The move from local to global has been a big change. The change in behaviour from a local autonomous organisation to a global one has been a major challenge. The introduction of a global aspect requires a lengthy process of analysis and planning in comparison to the shorter process for local projects. Waiting for the final decisions to be made and reduced control over these decisions has been frustrating. Trust and motivation have been impacted as a result. The lack of global procedures and emphasis on decision making at the US site have also contributed to this situation.

Chapter 5: Discussion

5.1 Introduction

In this chapter the main findings from the case study as presented in the previous chapter are analysed and discussed with reference to the literature. The Global Teaming Model (GTM) is reviewed for suitability to address the main issues surrounding GSD and regulation in GSD as discovered in the findings and earlier literature review. This analysis is used to answer the research question and fulfil the research objective:

***RQ:** How does regulation in medical device manufacturer's impact on global software development?*

***RO:** Provide a model for global software development applicable to a regulated medical device manufacturer engaged in internal software development.*

5.2 Local Priorities

Balancing local and global project priorities has been a struggle at Medi Inc. The GTM's Specific Practice 1.1 Global Task Management allows for reporting to managers at more than one location to address this. More attention is needed here. From a manager's perspective they need to be mindful of each other's local projects as well as the global ones. Any occurrences that might compromise any of the projects should be diligently reported to the affected project manager so that risks can be managed and plans adjusted. This is usual project management practice. From developers perspective their commitment and motivation is tied to those that influence their prospects. At Medi Inc they report to both local and global managers but it is their local functional manager that is responsible for their annual appraisals and directly influences their work life. Therefore their commitment lies with them. The GTM refers to rewards for the global team which could also create global commitment. The onus

should be on the managers to be committed to global projects which would in turn commit their direct reports.

When global direction is lacking “local leadership will fulfil its self interests” (Karolak, 1999). This has led to conflict over local and global priorities, lack of motivation for global projects and further deviation from global objectives.

5.3 Decision Making

Local sites have lost the power to set their own direction. Collaboration and ownership of tasks also covered in the GTM should alleviate this so sites feel they have some control and contribute value. The emphasis on decision making from Medi Inc’s headquarters only exacerbates this situation at remote sites (Karolak, 1999). Receiving instruction from a remote site without involvement in the decision making led to feelings of alienation (Beecham et al., 2010) and diminished trust.

When they did collaborate to gather requirements distance compromised understanding and agreement as Prikladnicki et al. (2003) also found. Agreement was also difficult to attain on technologies and approaches to development given each sites preference for its local methods which led to conflict. As an IT manager put it the “democracy” was not working. Karolak (1999) overcome this by appointing a “design authority” tasked with resolving technical conflicts. All team members could put forward their rationale to the design authority that makes the final decision and keeps the project going. Ensuring the objectivity of the design authority from individual sites would be crucial to its success. A neutral stance that supports a transparent global vision and ensures everyone’s place in achieving it would remove the political aspect and foster trust. Further to preventing delays they also suggest “deputy design authorities” that report to the design authority and have limited capabilities. This would also distribute responsibility and prevent blame between sites in the event that the project has issues based on the direction taken. It would also reduce demands on key decision makers that previously were inaccessible from remote sites of Medi Inc meaning that their availability would no longer be such a problem.

5.4 Global Standards

Global standards would provide a common ground for remote teams to operate within and prevent local conflicts. Putting global standards in place in relation to coding, processes, technologies and tools would also contribute to better decision making. It would mean that some decisions are already made and following the standard is all that is required. It would prevent ambiguity and allow projects to get started on time and stay on track. Further time savings could be gained from removing the need to know a range of methods or tools and training on new ones.

No common development or validation processes existed across the teams at Medi Inc and none was implemented for these projects due to the pressing deadline to deliver and the effort to create them. Battin et al. (2001) in their study of one of Motorola's first GSD projects had the same experience and like Medi Inc avoided a learning curve that would delay delivery but impacted project management. The project management impact did cause its own problems of confusion and processing overhead but the developers were not responsible for this aspect so could proceed to develop while the project manager coordinated the sites to understand and explain progress. However global processes do improve project outcomes so should at least have a minimised common global process (Cawley and Richardson, 2010). Managing the introduction of new processes in terms of project impact and outcomes should be considered so that projects can move forward but also be improved over time. Prikladnicki et al. (2003) study of organisations with differing levels of experience at GSD would support the view that problems become less complicated over time as GSD matures.

Medi Inc. have managed thus far with minimal global processes but this may be due to the closeness in culture of the sites, the good working relationship of the developers, lose deadlines and size of the projects involved. Relying on this to continue is risky and not scalable. It is inevitable that global projects will

expand and become more complex as the company seeks to become more global and integrated across the board.

5.5 Change Management for Transitioning to GSD

Medi Inc has moved to GSD without fully considering the implications. While their GSD projects have some solutions in place such as face to face meetings, global project and product managers, global project lists, independent tasks and work ownership these are not sufficient to address GSD issues. Commitment to the GSD effort is lacking and local situations constrain them, particularly the differences in organisational culture. The GTM offers guidance and would be beneficial in addressing some issues however a wider change management effort is required to actually implement it and to achieve sustainable commitment to GSD. Bosch and Bosch-Sijtsema (2010) surmise that a change in business strategy such as moving to global projects requires a change in tacit knowledge which is difficult to do because it is implicit and engrained. This change in organisational culture toward GSD needs senior management support to be successful (Nidiffer and Dolan, 2005).

5.6 Organizational Culture

The guidance provided by the Global Teaming Model (GTM) (Richardson et al., 2012) is very applicable to this case study and covers a wide range of considerations for achieving successful GSD. Some of the sub practices were more relevant than others. “Ensure awareness of cultural profiles” was not an issue that needed to be addressed at Medi Inc due to the closeness in culture of most of the sites. Neither was linguistics as English is the first language of Ireland, UK, Australia and US. It is Denmark’s second language and they have a high proficiency in it as it is commonly studied from an early age in schools there. However organizational culture was an issue and threatens quality (Bird et al., 2009) which is fundamental to medical device manufacturing. Attitudes to development, politics and power distribution fall under organizational culture (Johnson et al., 2011). These could be mitigated through better processes, collaboration and transparent objectives. Organisational culture

should be closer since all members are within one company (Carmel and Agarwal, 2001) but their history as independent sites has allowed them to develop differences (Damian et al., 2003). While organisational culture is not explicitly mentioned in the GTM the goals it aims to achieve should indirectly address it. It promotes openness through planned and documented processes including communication and reporting which should remove the uncertainty that leads to mistrust and allow them to develop a common organizational culture. This would also positively impact the fear that prevented sharing code and improve motivation to achieve global projects collaboratively.

5.7 Project Management and Lessons Learned

Project management at Medi Inc is only partially implemented across the organisation. Its effectiveness is also attributable to the immaturity of their processes (Ebert and De Neve, 2001). Project management enables control over projects (Holmström et al., 2006b, Richardson et al., 2012) which was not always evident at Medi Inc. It is necessary that it is fully and consistently implemented for GSD as well as regulation in GSD.

An important part of project management is lessons learned (PMBOK®, 2008) which could be applied to the GSD experience as well as the actual project itself. Casey and Richardson (2008) refer to this as leveraging and emphasise its usefulness in learning from experience to improve existing and future projects.

5.8 Medical Device Software Regulation in GSD

In this case study regulation at a global level is challenged by the differing quality management systems (QMSs) that exist at each site to guide regulatory compliance. Contrary to expectation the differing regulation between countries was not the issue. As a global company selling into global markets Medi Inc created QMSs at manufacturers that would cover all regulatory requirements across countries so they could sell anywhere overcoming this potential issue.

Using different QMSs and thus changing documentation and processes between GSD projects was a problem. As well as the time and effort used to choose the QMS to work with for every project. Coordination became more demanding as Bosch and Bosch-Sijtsema (2010) said it would when local processes are used for global projects.

It would seem that the solution would be to have one global QMS. However in suggesting this to the global project manager it was pointed out that this is not as simple as it sounds. Each local QMS would have to agree to the global QMS. The effort to achieve this is substantial. The main obstacles are the tight control on existing QMS and their critical role in sustaining the business. Not to mention local protectiveness. A change to the organisational structure would be required to deal with a global QMS and a change in local mindsets making it a difficult undertaking. The cost benefit analysis of such a change may not make it worthwhile, for the moment at least. In the interim a shared global location that references local systems and documents the process used between sites would clarify how it works and make it more traceable. At present it is understood but not officially documented. Karolak (1999) emphasises the importance of documentation to avoid assumptions and ambiguity particularly in geographically distributed projects.

The findings have shown that regulation does impact GSD. When compared with non regulated projects at Medi Inc also engaged in GSD they had more overhead and difficulties. It can be said that this is also the case with collocated projects, regulation introduces overhead. However when the project is globally distributed there are more elements to manage and control to ensure regulatory compliance. For example development and validation processes followed must be documented for regulatory purposes and follow a QMS using predefined controlled documentation. For GSD projects coordination is more difficult to ensure that that all involved sites have completed relevant documentation and have followed the process.

Risk increases with GSD due to the problems it introduces such as communication, control and coordination further complicating adherence to

regulatory requirements that are based on risk minimisation. Medi Inc's delay in decision making and confirming requirements reduced development time which in turn reduced the capacity to deliver. Working under pressure and delivering minimal requirements on occasion reduces quality but also increases risk of failure. The development process influences quality (Karolak, 1999) so this must be looked at from a GSD point of view. Their high quality standards and rigour in dealing with regulation beyond the guidelines serves as a buffer here but should not be relied on.

5.9 A Global Teaming Model for Medical Device Regulated Software Development

The GTM advocates creating and documenting processes to guide GSD projects and improve their operation (Richardson et al., 2012). The presence of documentation and process make it attractive for meeting regulatory requirements. The objective to improve coordination and manage risk also makes it suitable for implementation in a regulated environment. While it would help to facilitate regulation it does not provide for it specifically. An optional addition to the model covering regulation for medical device manufacturers would be useful.

From the literature review and case study performed in this research the following additions are proposed to the GTM.

1. Provide for managing the move from local to global projects and balancing the demands of both.
2. Provide for lessons learned to improve GSD.
3. Include feedback to a change management initiative.
4. Provide for regulation at a global level.

Recognising and managing local and global conflicts and lessons learned can be integrated into the existing GTM model specific goals. A change management initiative is set at the executive and senior management level and

is an organizational wide change. Therefore it is beyond the scope of the GTM. It would have a supporting role in implementing the GTM. Its goal is to establish a GSD strategy within the organization and to promote that vision from the top down (Johnson et al., 2011). Feedback from experiences gathered using the GTM would highlight areas where change is needed or has been difficult to achieve. A reference to a change management initiative is made under the sub-practice “Identify GSE project management tasks”. Proposed additions to address these are presented in Table 3.

Table 3 Proposed updates to SG 1: Define Global Project Management

Specific Goal 1: Define Global Project Management	
SP1.3 Global Project Management	
1.	<p><i>“Identify GSE project management tasks”</i></p> <p>Plan for lessons learned to be recorded during the project and reviewed at the end by all project team members. Update procedures based on this experience.</p> <p>Identify issues from lessons learned that require a wider initiative such as a change in organizational culture and report to a global change management agent.</p>
6.	<p><i>“Establish a risk management strategy”</i></p> <p>Identify and address local project demands that may pose a risk to the global project.</p>

Providing a decision making authority to avoid deadlock between sites is suggested as an addition to the existing GTM in Table 4.

Table 4 Proposed updates to SG 2: Define Management between Locations

Specific Goal2: Define Management Between Locations	
SP 2.1: Operating Procedures	
1.	<p><i>“Define how conflicts & differences of opinion between locations are addressed & resolved”</i></p> <p>Appoint an objective design authority responsible for approving global requirements (Karolak, 1999).</p> <p>Establish criteria for global projects based on global strategy and vision to guide decision making.</p>

Since a provision for regulation would be an optional add-on to the GTM it will be treated as a new “Specific Goal”. Its sub practices could also be broken up and assigned under specific practices of the existing specific goals as

appropriate but may be confusing and its importance lost. Processes that are already covered in the GTM are not repeated but should cover regulatory operations. Such processes include reporting and communication protocols and global project management. A global quality management system (QMS) could also be used specify these requirements. In the presence of a QMS that covers these measures they should be acknowledged in the model and referred to the QMS for guidance. The literature review and case study highlighted the important aspects of regulation as risk, quality, process, traceability and validation which are also included in the new specific goal “Define Global Regulation Management” as outlined in Figure 6 and detailed in Table 5.

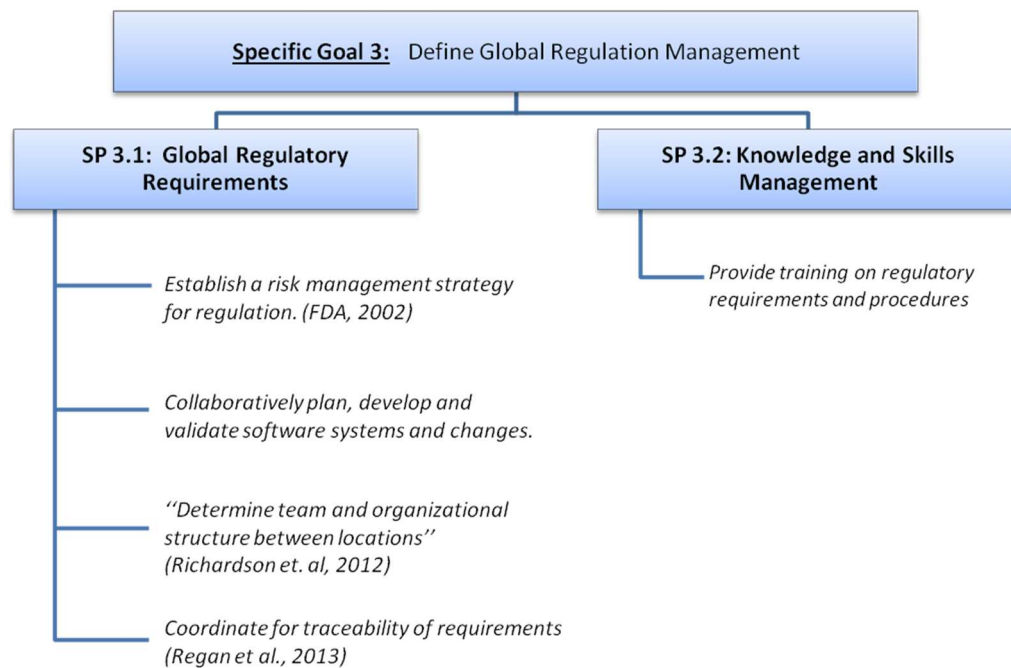


Figure 6 Proposed SG3: Define Global Regulation Management

Table 5 Proposed SG 3: Define Global Regulation Management

Specific Goal 3: Define Global Regulation Management	
Goal: To ensure that global software development meets global regulatory requirements and reduces impact to GSD.	
SP 3.1: Global Regulatory Requirements	
1.	<p><i>Establish a risk management strategy for regulation (FDA, 2002).</i></p> <p>All potential risks to meeting regulatory and quality requirements should be identified and addressed.</p> <p>Assess changes for impact on the existing system and validate accordingly to maintain a validated system (FDA, 2002).</p> <p>Include this risk management strategy in the global project management plan.</p>
2.	<p><i>Collaboratively plan, develop and validate software systems and changes.</i></p> <p>Collaboratively establish project requirements and functionality to meet regulatory requirements (Ingolfo et al., 2011).</p> <p>Collaboratively adopt a QMS as per ISO13485. Where a global QMS does not exist chose a local one. Establish rationale for choosing the QMS.</p> <p>Collaboratively adopt a global software development lifecycle (SDLC) process (FDA, 2002).. Document the SDLC process.</p> <p>Create a shared repository to include all project documentation including requirements, design and validation documents.</p> <p>To ensure traceability any documentation that is controlled outside of the shared repository must be linked to. Document the overall process so it is clear where to find relevant information. Collaboratively establish validation plan. Document plans and procedures for validation along the SDLC (FDA, 2002).</p> <p>Collaboratively agree roles and responsibilities for validation and include in global project management tasks.</p>
3.	<p><i>“Determine team and organizational structure between locations” (Richardson et al., 2012)</i></p> <p>Where possible ensure even distribution of developers and keep the number of sites involved to a minimum to reduce risk of defects (Cataldo and Nambiar, 2009). Consider collocated team for validation.</p>
4.	<p><i>Coordinate for traceability of requirements (Regan et al., 2013)</i></p> <p>Provide a shared Configuration Management (FDA, 2002) process and tool for traceability. Link between artefacts from requirements, process and validation to source code, bug reports, test cases etc. (Regan et al., 2013)</p> <p>Establish and document a change control processes.</p>
SP 3.2: Knowledge and Skills Management	
1.	<p><i>Provide training on regulatory requirements and procedures</i></p> <p>Provide training on the chosen QMS.</p> <p>Provide training on the software development lifecycle (SDLC) process.</p>

5.10 Conclusion

The history on which global software development began in this case study at Medi Inc has constrained it and been the cause of many of its issues. The transition has been further complicated by the lack of global direction and process. Local and global priorities have been conflicting which has affected commitment to global projects and stifled decision making. The absence of global standards and differences in organizational culture between sites has further complicated decision making and created conflict. A consistent and global project management approach has not been rolled out across projects. On the projects where it has they are not leveraging lessons learned to prevent repeating the same mistakes in the future. Regulation is made complicated further adding to coordination and communication demands on GSD projects as there is no global quality management system to provide for it. Based on these findings and the regulation measures identified in the literature updates to the Global Teaming model and additions were proposed to improve the situation.

Chapter 6: Conclusions and Further Research

6.1 Introduction

Taking them independently it is widely agreed that both global software development (GSD) and regulation in medical device manufacturing are complicated processes. This research has looked at the issues introduced by both and the effect of combining the two areas to perform GSD in a regulated environment. The previous chapter discussed the findings and proposed how to address them. This chapter reviews the research question and objective in light of these results. Limitations of the study are identified and further research is suggested.

6.2 GSD in Regulated Medical Device Manufacturers

***RQ:** How does regulation in medical device manufacturer's impact on global software development?*

Regulation impacts global software development as it imposes extra constraints on it that must be managed and controlled. These include defined and controlled processes of software development, risk management, traceability and validation which must be documented and easily referenced for audits and compliance.

Each country in the world provides a set of conditions that a medical device manufacturer must address in order to market and sell its product within that country. These regulations are often similar and can be combined to produce one set that will ensure products can be sold anywhere in the world as long as they are compliant. This is what our case study company Medi Inc has done to improve their process for compliance. Software development is also covered under regulation and therefore GSD is subject to this same regulation.

Regulation is closely linked to quality as regulations exist to ensure quality. Regulatory requirements include providing a quality management system (QMS) that provides for and controls regulation. As the literature and case

study have shown compliant software development must be development using a software development lifecycle, include risk management, be traceable from start to finish and validated to ensure that the requirements are fulfilled and there are no unexpected side effects. All these processes must be well defined and documented. Evidence that they have been followed must also be documented and traceable.

This study has shown how GSD introduces many challenges such as communication, control, culture and cooperation. These challenges are more complicated and tend to impact each other. Issues such as trust between remote sites can create fear or vice versa which then affects communication and knowledge sharing which can threaten the success of a project. Team dynamics are strained in GSD due to the lack of physical presence and the loss of informal communication. Human behaviour and interaction is complex. In the case study the move from local to global projects without clear direction and processes caused trust issues as control and politics became issues.

The requirements of regulation when combined with the issues of GSD make it difficult to control and therefore ensure regulatory compliance. This is a significant issue for a medical device manufacturer whose business depends on being compliant to sell its product and protect the patient. In particular the evolution from independent sites with independent QMSs in the case study impacted its GSD efforts. With one global project but no global QMS one of the local QMS had to be adopted per project. This was a time consuming effort that added to the communication and control demands of GSD.

6.3 A Global Teaming Model for Medical Device Regulated Software Development

***RO:** Provide a model for global software development applicable to a medical device manufacturer engaged in internal software development.*

Based on the research question answer, that medical device regulation does impact global software development the research objective could be achieved.

Due to the complexity of both GSD and regulation a structured approach is required to prevent failure. Existing research proposed a Global Teaming Model (GTM) (Richardson et al., 2012) that was reviewed against the case study, literature and regulatory requirements. The GTM addressed a lot of the issues presented by both the case study and literature and would be suitable and very useful to implement. The case study had specific issues in relation to local and global objectives and the conflict that introduced. Such conflict included balancing priorities and commitments and difficulties in collaborative global decision making. These issues related to existing areas covered by the GTM of “Global Project management” and “Management between Locations”. Proposed additions are provided in Table 3 and Table 4 of Chapter 5:.

In line with the structure of the GTM and with a view to implementing the GTM in a regulated environment a new “Specific Goal” labelled “Define Global Regulation Management” was added to the GTM as outlined in Figure 6 and Table 5 of Chapter 5: to provide a model for global software development applicable to a medical device manufacturer engaged in internal software development.

6.4 Limitations of the study

This study was limited to a specific case study making the results difficult to generalise within the medical device industry. While there will be commonalities with other medical device manufacturers the applicability of the results to others is limited without further research.

The limited time provided to complete the thesis and the complexity of the area of research and the many issues within it made it difficult to cover all aspects in detail and therefore limited the results.

A larger sample size for interviews may have produced more results. It would have been interesting and beneficial to expand the pool of resources outside the IT department to include the Quality departments across sites. Gaining access to these people would also be more difficult.

The common problems associated with interviews are a potential limitation. The honesty and accuracy of the participants' responses cannot be guaranteed. Every effort was taken to through the interview process to make them feel comfortable in providing information making it confidential and anonymous and to validate their responses against each other, company documentation and observations.

Finally the opportunity to validate the proposed additions to the global teaming model was not possible in this research.

6.5 Further Research

This research looked at the impact of regulation on global software development of by a medical device manufacturer. Particularly this software is used to manufacture the devices and support the business. To fully assess the affect of regulation on global software development further research would look at software as a medical device. Regulation is far more rigorous where the operation of the medical device is dependent on software since the software plays a more critical role in the patient's outcome.

This research is specific to the case at hand. It would be interesting to see if other companies in the same industry have had the same issues with moving to global projects and how they have managed it. These companies should be assessed for comparison to see if the results hold and to further develop the model from new findings. Following this a more robust model may be produced that could be implemented to test its validity. A longer study would be required to implement, monitor and adjust the model toward developing a standard.

Other industries may also be assessed to see if regulation can be generalised across the board or if there are any nuances specific to an industry that need to be addressed separately.

A suggestion to further this particular study would be to implement this GTM in the case study and after a period of time repeat the research to determine if there were improvements and if it made any difference to GSD in the regulated medical device environment.

6.6 Overall Conclusion

This research has established that regulation of the medical device industry impacts global software development. Both are complicated areas that require a lot of time and effort in planning and execution to succeed. The global teaming model is comprehensive in the aspects of global software development it covers and would be useful to implement at any company that operates this way. There are aspects of it that may not be fully applicable but it is worthwhile acknowledging they exist and ensuring every area for potential issues is covered. Given the importance of regulation in medical device manufacturing and the complexity of it a structured and committed approach needs to be applied to address it in the context of global software development. The decision to pursue GSD should consider if the benefits outweigh the effort and ability of the organisation to implement the necessary process and a model such as the GTM making it achievable and worthwhile. Given the limitations of this research it could be considered a starting point for further research in this area.

Bibliography

- ÅGERFALK, P. J., FITZGERALD, B., HOLMSTRÖM, H., LINGS, B., LUNDELL, B. & CONCHÚIR, E. O. A framework for considering opportunities and threats in distributed software development. International Workshop on Distributed Software Development, 2005. Citeseer, 47-61.
- ÅGERFALK, P. J., FITZGERALD, B., OLSSON, H. H. & CONCHÚIR, E. Ó. 2008. Benefits of global software development: the known and unknown. *Making Globally Distributed Software Development a Success Story*. Springer.
- ALIAS, N., RAHMAN, M. N. A., UJANG, A., GELAMDIN, R. B. & SAID, A. M. 2013. Research and Trends in the Studies of Webquest from 2005 to 2012: A Content Analysis of Publications in Selected Journals. *Procedia-Social and Behavioral Sciences*, 103, 763-772.
- BARTHELEMY, J. 2003. The seven deadly sins of outsourcing. *The Academy of Management Executive*, 17, 87-98.
- BATTIN, R. D., CROCKER, R., KREIDLER, J. & SUBRAMANIAN, K. 2001. Leveraging resources in global software development. *Software, IEEE*, 18, 70-77.
- BEECHAM, S., NOLL, J., RICHARDSON, I. & ALI, N. Crafting a global teaming model for architectural knowledge. Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on, 2010. IEEE, 55-63.
- BIANCHI, A., CAIVANO, D., LANUBILE, F. & VISAGGIO, G. Defect detection in a distributed software maintenance project. Proc. of the International Workshop on Global Software Development (GSD 2003), Portland, Oregon, USA, 2003. 48-52.
- BIRD, C., NAGAPPAN, N., DEVANBU, P., GALL, H. & MURPHY, B. 2009. Does distributed development affect software quality?: an empirical case study of Windows Vista. *Communications of the ACM*, 52, 85-93.
- BOLAND, D. & FITZGERALD, B. 2004. Transitioning from a co-located to a globally-distributed software development team: A case study at Analog Devices Inc.
- BOSCH, J. & BOSCH-SIJTSEMA, P. 2010. From integration to composition: On the impact of software product lines, global development and ecosystems. *Journal of Systems and Software*, 83, 67-76.
- BURTON, J., MCCAFFERY, F. & RICHARDSON, I. A risk management capability model for use in medical device companies. Proceedings of the 2006 international workshop on Software quality, 2006. ACM, 3-8.
- CARMEL, E. & ABBOTT, P. Configurations of global software development: offshore versus nearshore. Proceedings of the 2006 international workshop on Global software development for the practitioner, 2006. ACM, 3-7.
- CARMEL, E. & AGARWAL, R. 2001. Tactical approaches for alleviating distance in global software development. *Software, IEEE*, 18, 22-29.

- CASEY, V. Leveraging or exploiting cultural difference? Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on, 2009. IEEE, 8-17.
- CASEY, V. 2010. Developing trust in virtual software development teams. *Journal of theoretical and applied electronic commerce research*, 5, 41-58.
- CASEY, V. & MC CAFFERY, F. 2011. Med-Trace: Traceability Assessment Method for Medical Device Software Development.
- CASEY, V. & RICHARDSON, I. Project management within virtual software teams. Global Software Engineering, 2006. ICGSE'06. International Conference on, 2006a. IEEE, 33-42.
- CASEY, V. & RICHARDSON, I. Uncovering the reality within virtual software teams. Proceedings of the 2006 international workshop on Global software development for the practitioner, 2006b. ACM, 66-72.
- CASEY, V. & RICHARDSON, I. 2008. A structured approach to global software development. *Proceedings of European Systems and Software Process Improvement and Innovation, Dublin, Ireland*.
- CATALDO, M. & NAMBIAR, S. Quality in global software development projects: A closer look at the role of distribution. Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on, 2009. IEEE, 163-172.
- CAWLEY, O. & RICHARDSON, I. 2010. Lessons in global software development—local to global transition within a regulated environment. *European Systems & Software Process Improvement and Innovation, Grenoble, France. Springer*.
- CAWLEY, O., RICHARDSON, I. & WANG, X. 2011. Medical Device Software Development-A Perspective from a Lean Manufacturing Plant. *Software Process Improvement and Capability Determination. Springer*.
- CAWLEY, O., WANG, X. & RICHARDSON, I. 2010. Lean/agile software development methodologies in regulated environments—state of the art. *Lean Enterprise Software and Systems. Springer*.
- CHAKRABARTY, S. 2006. Making sense of the sourcing and shoring maze: various outsourcing and offshoring alternatives. *Outsourcing and Offshoring in the 21st Century: A socio economic perspective*, 1, 18-53.
- CHERBONNEAU, G. 2005. *A framework for improving the effectiveness of distributed project teams*. Massachusetts Institute of Technology.
- COATES, J. C. 2007. The goals and promise of the Sarbanes-Oxley Act. *The Journal of Economic Perspectives*, 91-116.
- CONCHÚIR, E. Ó., ÅGERFALK, P. J., OLSSON, H. H. & FITZGERALD, B. 2009. Global software development: where are the benefits? *Communications of the ACM*, 52, 127-131.
- COUGHLAN, P. & COUGHLAN, D. 2002. Action research for operations management. *International journal of operations & production management*, 22, 220-240.
- DAMIAN, D., LANUBILE, F. & OPPENHEIMER, H. L. Addressing the challenges of software industry globalization: the workshop on global software development. Proceedings of the 25th International Conference on Software Engineering, 2003. IEEE Computer Society, 793-794.

- DAMIAN, D. E. & ZOWGHI, D. The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. *Requirements Engineering*, 2002. Proceedings. IEEE Joint International Conference on, 2002. IEEE, 319-328.
- DAMIAN, D. E. & ZOWGHI, D. 2003. RE challenges in multi-site software development organisations. *Requirements engineering*, 8, 149-160.
- DARKE, P., SHANKS, G. & BROADBENT, M. 1998. Successfully completing case study research: combining rigour, relevance and pragmatism. *Information systems journal*, 8, 273-289.
- DESANCTIS, G. & JACKSON, B. M. 1994. Coordination of information technology management: Team-based structures and computer-based communication systems. *Journal of Management Information Systems*, 10, 85-110.
- DESHPANDE, S. 2012. *A coordination model for global software development teams*. Doctor of Philosophy Unpublished doctoral thesis, University of Limerick.
- DIBBERN, J., GOLES, T., HIRSCHHEIM, R. & JAYATILAKA, B. 2004. Information systems outsourcing: a survey and analysis of the literature. *ACM SIGMIS Database*, 35, 6-102.
- DICICCO-BLOOM, B. & CRABTREE, B. F. 2006. The qualitative research interview. *Medical education*, 40, 314-321.
- EBERT, C. & DE NEVE, P. 2001. Surviving global software development. *Software, IEEE*, 18, 62-69.
- EBERT, C., PARRO, C. H., SUTTELS, R. & KOLARCZYK, H. Improving validation activities in a global software development. Proceedings of the 23rd international Conference on Software Engineering, 2001. IEEE Computer Society, 545-554.
- EDWARDS, H. K. & SRIDHAR, V. Analysis of the effectiveness of global virtual teams in software engineering projects. *System Sciences*, 2003. Proceedings of the 36th Annual Hawaii International Conference on, 2003. IEEE, 9 pp.
- EISENHARDT, K. M. 1989. Building theories from case study research. *Academy of management review*, 14, 532-550.
- ESPINOSA, J. A. & CARMEL, E. 2003a. The impact of time separation on coordination in global software teams: a conceptual foundation. *Software Process: Improvement and Practice*, 8, 249-266.
- ESPINOSA, J. A. & CARMEL, E. Modeling coordination costs due to time separation in global software teams. *International Workshop on Global Software Development at ICSE*, 2003b. 64-69.
- ESPINOSA, J. A., KRAUT, R. E., SLAUGHTER, S. A., LERCH, J. F., HERBSLEB, J. D. & MOCKUS, A. Shared Mental Models, Familiarity and Coordination: A Mult-Method Study of Distributed Software Teams. 2001. AIS.
- ESPINOSA, J. A., SLAUGHTER, S. A., KRAUT, R. E. & HERBSLEB, J. D. 2007. Team knowledge and coordination in geographically distributed software development. *Journal of Management Information Systems*, 24, 135-169.
- EUROPEAN COUNCIL 1993. Council Directive 93/42/EEC of 14 June 1993 concerning medical devices. Luxembourg: Official Journal of The European Communities.

- FDA 1997. PART 11 ELECTRONIC RECORDS; ELECTRONIC SIGNATURES. In: ADMINISTRATION, U. S. F. A. D. (ed.) *CFR - Code of Federal Regulations Title 21*. Silver Spring.
- FDA 2002. General Principles of Software Validation; Final Guidance for Industry and FDA Staff. *Rockville (MD): FDA*. U.S. Department Of Health and Human Services Food and Drug Administration.
- FDA. 2011. *Quality System (QS) Regulation/Medical Device Good Manufacturing Practices* [Online]. Available: <http://www.fda.gov/medicaldevices/deviceregulationandguidance/postmarketrequirements/qualitysystemsregulations/default.htm> [Accessed June 15 2014].
- FENEMA, P. C. 2002. *Coordination and control of globally distributed software projects*, Erasmus Research Institute of Management (ERIM).
- FITZGERALD, B. & HOWCROFT, D. 1998. Towards dissolution of the IS research debate: from polarization to polarity. *Journal of Information Technology*, 13, 313-326.
- FLYVBJERG, B. 2006. Five misunderstandings about case-study research. *Qualitative inquiry*, 12, 219-245.
- FOWLER, M. & HIGHSMITH, J. 2001. The agile manifesto. *Software Development*, 9, 28-35.
- GALVINA, Z. & SMITE, D. Low Degree of Separation Does Not Guarantee Easy Coordination. *Software Engineering and Advanced Applications (SEAA)*, 2012 38th EUROMICRO Conference on, 2012. IEEE, 345-348.
- GIBSON, C. B. & COHEN, S. G. 2003. *Virtual teams that work: Creating conditions for virtual team effectiveness*, John Wiley & Sons.
- GLASER, B. G. & STRAUSS, A. L. 2009. *The discovery of grounded theory: Strategies for qualitative research*, Transaction Publishers.
- GOTEL, O., KULKARNI, V., SAY, M., SCHARFF, C. & SUNETNANTA, T. 2012. Quality indicators on global software development projects: does 'getting to know you' really matter? *Journal of Software: Evolution and Process*, 24, 169-184.
- GRINTER, R. E., HERBSLEB, J. D. & PERRY, D. E. The geography of coordination: dealing with distance in R&D work. *Proceedings of the international ACM SIGGROUP conference on Supporting group work*, 1999. ACM, 306-315.
- HENDERSON, J. C. & VENKATRAMAN, N. 1993. Strategic alignment: Leveraging information technology for transforming organizations. *IBM systems journal*, 32, 4-16.
- HERBSLEB, J. D. Global software engineering: The future of socio-technical coordination. *2007 Future of Software Engineering*, 2007. IEEE Computer Society, 188-198.
- HERBSLEB, J. D. & GRINTER, R. E. 1999. Architectures, coordination, and distance: Conway's law and beyond. *IEEE software*, 16, 63-70.
- HERBSLEB, J. D. & MOCKUS, A. 2003. An empirical study of speed and communication in globally distributed software development. *Software Engineering, IEEE Transactions on*, 29, 481-494.
- HERBSLEB, J. D. & MOITRA, D. 2001. Global software development. *Software, IEEE*, 18, 16-20.

- HINDS, P. & MCGRATH, C. Structures that work: social structure, work structure and coordination ease in geographically distributed teams. Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work, 2006. ACM, 343-352.
- HOLLAND, C. P. & LIGHT, B. Global enterprise resource planning implementation. Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on, 1999. IEEE, 10 pp.
- HOLMSTRÖM, H., CONCHÚIR, E. Ó., ÅGERFALK, P. J. & FITZGERALD, B. Global software development challenges: A case study on temporal, geographical and socio-cultural distance. Global Software Engineering, 2006. ICGSE'06. International Conference on, 2006a. IEEE, 3-11.
- HOLMSTRÖM, H., FITZGERALD, B., ÅGERFALK, P. J. & CONCHÚIR, E. Ó. 2006b. Agile practices reduce distance in global software development. *Information Systems Management*, 23, 7-18.
- HSIEH, H.-F. & SHANNON, S. E. 2005. Three approaches to qualitative content analysis. *Qualitative health research*, 15, 1277-1288.
- INGOLFO, S., SIENA, A., MYLOPOULOS, J., SUSI, A. & PERINI, A. 2013. Arguing regulatory compliance of software requirements. *Data & Knowledge Engineering*, 87, 279-296.
- ISPE, G. 2001. Guide for Validation of Automated Systems. GAMP.
- JALALI, S., GENCEL, C. & ŠMITE, D. Trust dynamics in global software engineering. Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 2010. ACM, 23.
- JICK, T. D. 1979. Mixing qualitative and quantitative methods: Triangulation in action. *Administrative science quarterly*, 602-611.
- JOHNSON, G., WHITTINGTON, R., SCHOLLES, K. & PYLE, S. 2011. *Exploring strategy: text & cases*, Financial Times Prentice Hall Harlow.
- KAPLAN, A. V., BAIM, D. S., SMITH, J. J., FEIGAL, D. A., SIMONS, M., JEFFERYS, D., FOGARTY, T. J., KUNTZ, R. E. & LEON, M. B. 2004. Medical device development from prototype to regulatory approval. *Circulation*, 109, 3068-3072.
- KAPLAN, B. & DUCHON, D. 1988. Combining qualitative and quantitative methods in information systems research: a case study. *MIS quarterly*, 571-586.
- KAROLAK, D. W. 1999. *Global software development: managing virtual teams and environments*, IEEE Computer society press.
- KEARNS, G. & LEDERER, A. L. 2000. The effect of strategic alignment on the use of IS-based resources for competitive advantage. *The Journal of Strategic Information Systems*, 9, 265-293.
- KLÜMPER, M. & VOLLEBREGT, E. 2009. Navigating the New EU Rules for Medical Devices Software. *RAJ Devices*, 17, 1-8.
- KOBITZSCH, W., ROMBACH, D. & FELDMANN, R. L. 2001. Outsourcing in India [software development]. *Software, IEEE*, 18, 78-86.
- KOTLARSKY, J. & OSHRI, I. 2005. Social ties, knowledge sharing and successful collaboration in globally distributed system development projects. *European Journal of Information Systems*, 14, 37-48.

- KRAUT, R., EGIDO, C. & GALEGHER, J. Patterns of contact and communication in scientific research collaboration. Proceedings of the 1988 ACM conference on Computer-supported cooperative work, 1988. ACM, 1-12.
- LAYMAN, L., WILLIAMS, L., DAMIAN, D. & BURES, H. 2006. Essential communication practices for Extreme Programming in a global software development team. *Information and software technology*, 48, 781-794.
- LEGARD, R., KEEGAN, J. & WARD, K. 2003. In-depth interviews. *Qualitative research practice: A guide for social science students and researchers*, 138-169.
- LERO. *Medical Device Software Engineering* [Online]. Available: <http://www.lero.ie/research/competency/medicaldevicesoftwareengineering> [Accessed June 14th 2014].
- LINGS, B., LUNDELL, B., AGERFALK, P. J. & FITZGERALD, B. A reference model for successful Distributed Development of Software Systems. Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, 2007. IEEE, 130-139.
- MARTINS, L. L., GILSON, L. L. & MAYNARD, M. T. 2004. Virtual teams: What do we know and where do we go from here? *Journal of management*, 30, 805-835.
- MASON, J. 2002. *Qualitative researching*, Sage.
- MAZNEVSKI, M. L. & CHUDоба, K. M. 2000. Bridging space over time: Global virtual team dynamics and effectiveness. *Organization science*, 11, 473-492.
- MC CAFFERY, F., CASEY, V., SIVAKUMAR, M., COLEMAN, G., DONNELLY, P. & BURTON, J. 2012. Medical Device Software Traceability. *Software and Systems Traceability*. Springer.
- MC CAFFERY, F. & COLEMAN, G. 2007. Developing a configuration management capability model for the medical device industry. *International Journal of Information Systems and Change Management*, 2, 139-154.
- MC CAFFERY, F. & DORLING, A. 2009. Medi SPICE: an overview.
- MC HUGH, M., MC CAFFERY, F., CASEY, V. & PIKKARAINEN, M. 2012. Integrating Agile Practices with a Medical Device Software Development Lifecycle.
- MCALLISTER, P. & JESWIET, J. 2003. Medical device regulation for manufacturers. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 217, 459-467.
- MEHRFARD, H., PIRZADEH, H. & HAMOU-LHADJ, A. 2010. Investigating the capability of agile processes to support life-science regulations: The case of XP and FDA regulations with a focus on human factor requirements. *Software Engineering Research, Management and Applications 2010*. Springer.
- MISRA, S. & FERNÁNDEZ-SANZ, L. Quality Issues in Global Software Development. ICSEA 2011, The Sixth International Conference on Software Engineering Advances, 2011. 325-330.
- MOCKUS, A. & HERBSLEB, J. Challenges of global software development. Software Metrics Symposium, 2001. METRICS 2001. Proceedings. Seventh International, 2001. IEEE, 182-184.

- MOE, N. B. & ŠMITE, D. 2007. Understanding lacking trust in global software teams: A multi-case study. *Product-Focused Software Process Improvement*. Springer.
- MYERS, M. D. & AVISON, D. 1997. Qualitative research in information systems. *Management Information Systems Quarterly*, 21, 241-242.
- NIDIFFER, K. E. & DOLAN, D. 2005. Evolving distributed project management. *Software, IEEE*, 22, 63-72.
- NISSSEN, H. W. 2004. Designing the Inter-Organization a] Software Engineering Cooperation: An Experience Report.
- NOLL, J., BEECHAM, S. & RICHARDSON, I. 2010. Global software development and collaboration: barriers and solutions. *ACM Inroads*, 1, 66-78.
- NSAI. 2009. *ISO 9001 - Quality Management Systems* [Online]. Dublin: NSAI. Available: <http://www.nsai.ie/our-services/certification/management-systems/iso-9001.aspx> [Accessed June 8 2014].
- ORLIKOWSKI, W. J. & BAROUDI, J. J. 1991. Studying information technology in organizations: Research approaches and assumptions. *Information systems research*, 2, 1-28.
- OSHRI, I., KOTLARSKY, J., ROTTMAN, J. W. & WILLCOCKS, L. L. 2009. Global sourcing: recent trends and issues. *Information Technology & People*, 22, 192-200.
- OSHRI, I., KOTLARSKY, J. & WILLCOCKS, L. P. 2007. Global software development: Exploring socialization and face-to-face meetings in distributed strategic projects. *The Journal of Strategic Information Systems*, 16, 25-49.
- PATTON, M. Q. 1999. Enhancing the quality and credibility of qualitative analysis. *Health services research*, 34, 1189.
- PINJANI, P. & PALVIA, P. 2013. Trust and knowledge sharing in diverse global virtual teams. *Information & Management*, 50, 144-153.
- PMBOK® 2008. *A Guide to the Project Management Body of Knowledge: (PMBOK® Guide)*, Project Management Institute.
- PRIKLADNICKI, R., AUDY, J. L. & EVARISTO, R. 2004. An empirical study on global software development: Offshore insourcing of IT projects.
- PRIKLADNICKI, R., AUDY, J. L. N., DAMIAN, D. & DE OLIVEIRA, T. C. Distributed Software Development: Practices and challenges in different business strategies of offshoring and onshoring. Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, 2007. IEEE, 262-274.
- PRIKLADNICKI, R., AUDY, J. L. N. & EVARISTO, R. A reference model for global software development: findings from a case study. Global Software Engineering, 2006. ICGSE'06. International Conference on, 2006. IEEE, 18-28.
- PRIKLADNICKI, R., NICOLAS AUDY, J. L. & EVARISTO, R. 2003. Global software development in practice lessons learned. *Software Process: Improvement and Practice*, 8, 267-281.
- PYYSIÄINEN, J. Building trust in global inter-organizational software development projects: problems and practices. International Workshop on Global Software Development, 2003. 69-74.

- RAKITIN, R. 2006. Coping with defective software in medical devices. *Computer*, 39, 40-45.
- RALYTÉ, J., LAMIELLE, X., ARNI-BLOCH, N. & LÉONARD, M. A framework for supporting management in distributed information systems development. *Research Challenges in Information Science*, 2008. RCIS 2008. Second International Conference on, 2008. IEEE, 381-392.
- RAMASUBBU, N. & BALAN, R. K. Globally distributed software development project performance: an empirical analysis. *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, 2007. ACM, 125-134.
- RAMESH, V. & DENNIS, A. R. The object-oriented team: lessons for virtual teams from global software development. *System Sciences*, 2002. HICSS. *Proceedings of the 35th Annual Hawaii International Conference on*, 2002. IEEE, 212-221.
- RAO, M. T. 2004. Key issues for global IT sourcing: country and individual factors. *Information Systems Management*, 21, 16-21.
- REGAN, G., MC CAFFERY, F., MC DAID, K. & FLOOD, D. 2013. Medical device standards' requirements for traceability during the software development lifecycle and implementation of a traceability assessment model. *Computer Standards & Interfaces*, 36, 3-9.
- RICHARDSON, I., CASEY, V., MCCAFFERY, F., BURTON, J. & BEECHAM, S. 2012. A process framework for global software engineering teams. *Information and Software Technology*, 54, 1175-1191.
- RITCHIE, J. & LEWIS, J. 2003. *Qualitative research practice: A guide for social science students and researchers*, Sage.
- RUBIN, H. J. & RUBIN, I. S. 1995. Y Rubin, IS (1995): *Qualitative interviewing. The art of hearing data*. London: Sage cap.
- SALE, J. E., LOHFELD, L. H. & BRAZIL, K. 2002. Revisiting the quantitative-qualitative debate: Implications for mixed-methods research. *Quality and quantity*, 36, 43-53.
- SAUNDERS, M., LEWIS, P. & THORNHILL, A. 2012. *Research methods for business students*.
- SCHREIER, M. 2012. *Qualitative content analysis in practice*, Sage Publications.
- SCHWANDT, T. A. 2001. *Dictionary of qualitative inquiry*, Sage.
- SEAMAN, C. B. 1999. Qualitative methods in empirical studies of software engineering. *Software Engineering, IEEE Transactions on*, 25, 557-572.
- SEARCHSOFTWAREQUALITY. 2014. *Definition of V-Model* [Online]. TechTarget. Available: <http://searchsoftwarequality.techtarget.com/definition/V-Model> [Accessed June 20 2014].
- SLOWEY, K. & RICHARDSON, I. 2006. A Practical Application of Content Analysis Techincal Report No: UL-CSIS-06-03. University of Limerick, Ireland.
- SPENCE, J. W. There has to be a better way![software development]. *Agile Conference*, 2005. *Proceedings*, 2005. IEEE, 272-278.

- SUDERSHANA, S., VILLCA-ROQUE, A. & BALDANZA, J. Successful Collaborative Software Projects for Medical Devices in an FDA Regulated Environment: Myth or Reality? Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, 2007. IEEE, 217-224.
- SZYMANSKI, C. H. & PRIKLADNICKI, R. The Evolution of the Internal Offshore Software Development Model at Dell Inc. Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, 2007. IEEE, 40-50.
- VOGEL, D. 2006. Agile Methods: Most are not ready for prime time in medical device software design and development. *DesignFax Online*, 1-6.
- YIN, R. K. 2009. *Case study research: Design and methods*, sage.

Appendices

Appendix A: Interview Protocol

Semi Structure Interview Guideline Questions

1. Can you give me an example of a **global software development** project that you worked on where other team members were **located at different sites**?
2. Name three things that worked well on this global project?
3. Name three things that DID NOT work well on this global project?
4. Why were these issues?
5. Were there any solutions put in place to address these?
6. What would you suggest could have been done?
7. How involved was everyone?
8. Was there overall responsibility?
9. What **processes** were put in place to facilitate this way of working?
 - a. Communication/Understanding (meetings, tools, formal and informal processes.)
 - b. Requirements gathering and understanding.
 - c. Collaboration. (Actually work together? Working relationship? Any trust issues?)
 - d. Quality
 - e. Roles and responsibilities (Control)
10. Were there any extra demands on you to work like this?

Specific Goal 1: Define Global Project Management

SP 1.1: Global Task Management

- How was the team structured?
- How were tasks allocated within the team?

SP 1.2: Knowledge and Skills Management

- Was training assessed and provided before undertaking this project? (Had they worked like this before?)

- How was the team selected? Skills?
- Were there any cultural differences between the sites?
- What communication requirements were needed to work like this?

SP 1.3: Global Project Management

What project management tasks were carried out?

- Cooperation and coordination procedures between locations
- Ensure awareness of cultural profiles by project managers
- Establish reporting procedures between locations
- Establish a risk management strategy

Specific Goal 2: Define Management between Locations

SP 2.1: Operating Procedures

- Define how conflicts & differences of opinion between locations are addressed & resolved
- Communication procedures
- Meetings

SP 2.2: Collaboration between locations

Extent of collaboration:

1. in GSD process
 - Identify common goals, objectives and rewards
 - Collaboratively establish and maintain work product ownership boundaries
 - Collaboratively establish and maintain interfaces and processes
 - Collaboratively develop, communicate and distribute work plans
2. In work completion e.g. handover – ‘follow the sun’, modular, phased etc.

11. How was regulation dealt with on this project?

12. Why was the project set up like this?

13. Do you think the objectives were achieved?

14. Could they have been achieved in a better alternative way? How and why? (Asking this again in light of extra info. provided since)

15. Is there anything else you would like to add in light of this discussion and your experience?

Checklist for identifying existing themes:

	A	B	C	D	E	F	G	H	I	J
1. Communication										
2. Time difference issues										
3. Project Mgmt										
4. Risk Mgmt										
5. Roles & Responsibilities										
6. Motivation										
7. Coordination (integration)										
8. Cooperation										
9. Control										
10. Culture										
11. Quality/Regulation										
12. Trust										

Appendix B: Codes for Content Analysis

Initial set of codes

Title	Code
Time zone	TZO
Communication	COM
Control	CON
Culture	CUL
Trust	TRU
Motivation	MOT
Fear	FEA
Team	TEA
Procedures	PROC
Quality	QA
Regulation	REG
Collaboration	COL
Training	TRA
Language	LAN
Tools	TOOL

Final set of codes

Title	Code
Time zone	TZO
Communication	COM
Control	CON
Culture	CUL
Trust	TRU
Motivation	MOT
Fear	FEA
Team	TEA
Procedures	PROC
Quality	QA
Regulation	REG
Collaboration	COL
Training	TRA
Language	LAN
Tools	TOOL
Modular	MOD
Sharing	SHAR
Requirements	REQ
Dependencies	DEP
Power	POW
Standards	STAN
Planning	PLAN
Onsite	ONS

Team Structure	TSTRUC
Task Allocation	TALLOC
Validation	VAL
Visibility	VIS
Priority	PRIO
Roles and responsibilities	R&R
Informal communication	INF
Politics	POL
Perspectives	PER
Follow the sun	FWS
Face to face meeting	F2F
Relationship	REL
Leadership	LEAD
Quality	QUAL