Software Quality Across Borders: Three Case Studies on Company Internal Alignment

Sebastian Barney[a,b], Varun Mohankumar[b], Panagiota Chatzipetrou[c],
Aybuke Aurum[b], Claes Wohlin[a], Lefteris Angelis[c]

[a] *Blekinge Institute of Technology, Sweden*
sebastian@unswalumni.com, claes.wohlin@bth.se
[b] *School of Information Systems, Technology and Management,*
University of New South Wales, Australia
aybuke@unsw.edu.au, z3207004@unswalumni.com
[c] *Department of Informatics,*
Aristotle University of Thessaloniki, Greece
pchatzip@csd.auth.gr, lef@csd.auth.gr

Abstract

Context: Software quality issues are commonly reported when offshoring software development. Value-based software engineering addresses this by ensuring key stakeholders have a common understanding of quality.

Objective: This work seeks to understand the levels of alignment between key stakeholder groups within a company on the priority given to aspects of software quality developed as part of an offshoring relationship. Furthermore, the study aims to identify factors impacting the levels of alignment identified.

Method: Three case studies were conducted, with representatives of key stakeholder groups ranking aspects of software quality in a hierarchical cumulative exercise. The results are analysed using Spearman rank correlation coefficients and inertia. The results were discussed with the groups to gain a deeper understanding of the issues impacting alignment.

Results: Various levels of alignment were found between the various groups. The reasons for misalignment were found to include cultural factors, control of quality in the development process, short-term versus long-term orientations, understanding of cost-benefits of quality improvements, communication and coordination.

Conclusions: The factors that negatively affect alignment can vary greatly between different cases. The work emphasises the need for greater support to align company internal success-critical stakeholder groups in their understanding of quality when offshoring software development.

Keywords:
software, quality, alignment, global software development, insourcing, outsourcing, onshore, offshore

1. Introduction

The rapid rise of global software development (GSD) (Dibbern et al., 2008) has brought with it new benefits and challenges. The main drivers for this practice are cost reduction, proximity to markets and making use of different competencies (Šmite et al., 2010), but there are also many challenges. The most commonly cited challenges in GSD contexts concern communication and coordination (Mockus and Herbsleb, 2001; Herbsleb, 2007; Carmel and Abbott, 2007), both essential elements in creating alignment between the growing number of stakeholders involved in software development.

There is a wide body of evidence showing that organisations that can create alignment through "convergent intentions, shared understanding and coordinated procedures" will outperform organisations that cannot create this alignment (Chan, 2002; Chan and Reich, 2007). Although Chan (2002), and Chan and Reich (2007) refer to Business-Information Systems and IT alignment in Information Systems departments, the alignment related issues that they addressed in their articles are very relevant to software development in a global context too. Alignment of stakeholders allows them to collaborate more effectively and produce results that support long-term business strategies, while highly misaligned teams can cause conflict and eventually lead to project failure.

Studies of alignment between groups, stakeholders and so forth are not new. It is a well-known challenge. However, GSD creates a number of challenges in relation to alignment. A challenge identified when working with one of our industrial partners was the alignment in terms of the understanding of the importance of different software quality attributes. The company was interested in understanding both whether different roles have the same understanding of the importance of different software quality attributes, and whether or not people at different sites of the company shared this understanding of the importance of different software quality attributes. Based on the identified industrial challenge in relation to alignment of the understanding of the importance of different software quality attributes, three case studies are presented seeking to understand alignment in environments employing two different GSD strategies. Thus, the objective is that the case studies should help in (1) understanding the levels of alignment between stakeholder groups in terms of software quality attributes, and (2) identification of reasons for the level of alignment, or potential misalignment. These two items form the basis for two of the research questions formulated.

This paper is an extension of Barney et al. (2011b). The method applied to evaluate the alignment in relation to software quality attributes was originally proposed in Barney and Wohlin (2009). The paper that forms the basis for this work, Barney et al. (2011b), is the first to apply this method to study alignment in a GSD setting, and extends the statistical analysis originally proposed for the method. This paper extends Barney et al. (2011b) to include an additional case study from the financial services industry with a US-based company undertaking offshore development in Australia and India. The objective was to include an additional company coming from a different application domain and where the GSD context was different (including the companies having different GSD strategies). Furthermore, this article adds an additional scope by introducing a new research question as it applies the Stakeholder Alignment Assessment Method for Software Quality (SAAM-SQ) for a new company in a GSD context. This gives an opportunity to study the usefulness of SAAM-SQ in new context.

The remainder of this paper is structured as follows. Key literature introducing the topic is presented in Section 2. The research questions and methodology are presented in Section 3. Information about the case studies is presented in Section 4. The levels of alignment found in the case studies are presented in Section 5 and the reasons identified for these levels of alignment are presented in Section 6. A reflection on the use of the method is provided in Section 7. A discussion of the results is made in Section 8, with the conclusions in Section 9.

2. Background

This section introduces key concepts and related work.

2.1. Software Quality

There have long been many definitions of software quality (Kitchenham and Pfleeger, 1996), with the most common in software engineering being ‘conformance to specification’ and ‘fit for purpose’. Here software quality is used to denote the general perception of how good the software is according to the definitions of software quality, while software quality attributes are used to acknowledge that software quality constitutes of many different attributes, for example performance, reliability, usability and security to name a few. The growing body of value-based software engineering (VBSE) literature recognizes that perceptions of software quality are individual and shaped by experiences (Johansson et al., 2001). VBSE suggests that the most successful way to move forward with software development is for the success-critical stakeholder groups to reach mutual consensus. The success-critical stakeholder groups are the groups upon whom the success of the product depends (Boehm and Jain, 2006).

Software quality does not need to be perfect (Yourdon, 1995). The tough question to answer is, ‘how much less than perfect is sufficient?’ There is no single answer to this question, as any answer must consider the context in which it is being asked. The Quper model helps answer this question by defining a relationship between the level of quality, the benefits and the costs (Regnell et al., 2008). The model defines a series of quality/benefit levels, in which a software product can be categorised as *useless*, *useful*, *competitive* or *excessive*. It recognizes that under-investment leads to an unusable product, but over-investment costs more than the benefits gained.

2.2. Models of Software Quality

There are many models that describe software quality. The most common representations of software quality present a hierarchy of quality attributes. Examples of such models include the quality models by McCall and his colleagues—often referred to as McCall’s model (McCall et al., 1977), the quality model by Boehm and his colleagues—often referred to as Boehm’s model (Boehm et al., 1976, 1978), and ISO9126 (2001). All of these models are criticized for various deficiencies. The major complaints about ISO 9126 are missing or insufficiently detailed aspects of quality, and insufficient information about measuring the aspect defined.

An alternative approach to defining quality models was undertaken in Dromey’s quality model (Dromey, 1996). This work proposes the definition of the actions required to achieve the desired level of quality rather than describing the quality itself. As such this approach provides developers with concrete actions that will achieve the desired quality.

2.3. Key Stakeholder Alignment

Stakeholder alignment has been defined as "convergent intentions, shared understanding and coordinated procedures" (Chan, 2002), inline with the goals of VBSE. Individuals from different areas in a product development organisation can understand the same aspect of the product differently (Brown and Eisenhardt, 1995). This problem is worse in environments where work is handed sequentially from team to team, but it is better in environments with cross-functional teams. Forming cross-functional teams to work on concrete problems has been found an effective way to increase the information flow.

There is overwhelming evidence showing aligned groups outperform those who are not aligned (Chan and Reich, 2007). A shared understanding is a precondition for stakeholders to collaborate more effectively (Barczak and McDonough, 2003), and hence being more aligned will help producing systems that support the long-term business strategies. It is important that all stakeholders are included in the process of goals, planning, defining roles and defining responsibilities (Cramton, 2001). Highly misaligned teams can cause conflict and eventually lead to the failure of a project.

Brown and Eisenhardt (1995) present primary and secondary evidence of the importance of communication both within and between groups during product development to achieve alignment. From this work they conclude that effective communication supports successful product development and requires a communication strategy. Communication strategies that frequently employ multiple communication channels with the various groups involved in product development are associated with greater levels of success. There is research that shows that teams that engage in more thorough internal communication have a superior performance as they are more successful at a range of activities, including goal definition. Cramton (2001), after examining 45 geographically dispersed teams of students who work on an assignment project for 7 weeks, suggests exploring the advance potential differences in situations for dispersed teams. For example, ideally every member in the team should receive the same information; however the information overload can be a big problem in practice. Hence, it is important that team members communicate the information that makes sense within the parameters of the collaboration and actively seek for information rather than making assumptions.

Shared knowledge helps a team communicate more effectively as they have shared expectations, experiences and vocabulary (Espinosa et al., 2007). There are many different types of shared knowledge that support product development, including knowledge of strategies, goals, tasks and processes. Cramton (2001) recommends exploring incentives, and goals that will affect team members and points out that communication is not the only problem. The complex distributed dynamic nature of teams will affect the process and outcomes if not managed well.

Stakeholder alignment requires an ongoing effort to ensure that the group remain aligned (Chan, 2002). Communication, coordination, control, supervision, creating social bonds and building trust, (Carmel and Abbott, 2007), are key to creating alignment between stakeholders. A lack of awareness and belief in alignment has been identified as one of the main contributors to misalignment (Chan and Reich, 2007).

Several studies highlight the challenges related to alignment when it comes to the view and priorities of different software quality attributes. Based on a large survey in the US of 13 different software quality attributes, it is concluded that different roles may affect the priorities, but also that a strong shared culture could help mitigate the problems (Haigh, 2010). Similar findings are reported from a survey on quality requirements, where the researchers report on the importance of different quality requirements and how the role of people affects their priorities (de la Vara et al., 2011). Issues related to software quality attributes have also been surveyed within the area of software architecture (Ameller et al., 2012). The authors conclude that the software architects in their study have some differences in their vocabulary and terminology. In conclusion, the surveys of industry practice in relation to software quality attributes show that alignment is a challenge. Thus, there is a need to have better support for alignment in relation to software quality attributes, which becomes even more challenging if working in a distributed development organisation.

At the same time, the increase in amount of software development being sent offshore from Australia is typified by the Australian Federal Police preparing to send the software development work of its core policing system to an offshore vendor (Colley, 2008). The security and privacy issues have traditionally meant that organisations such as Australian Federal Police and Australian Customs Service kept their software development internal. However, the fact that even these organisations are outsourcing software development and sending it offshore typifies the popularity of the option.

Although a wealth of academic literature exists on examining alignment issues among teams and team members, these studies mainly focus on alignment between stakeholders in dispersed teams—for example, Cramton (2001);

or the alignment between stakeholders on quality attributes on co-located groups—for example Phongpaibul and Boehm (2005); or the impact of global software development on quality in general—for example (Bird et al., 2009).

To the best of our knowledge there is a gap in the literature about the understanding of alignment between success-critical stakeholders when it comes to the prioritisation of different software quality attributes in the context of outsourced and offshored software development. Hence, the objective of this research is to investigate the alignment between success-critical stakeholders between roles (e.g., business users, business analysts, developers and testers) and between multiple project sites (e.g., Sweden, Australia, India and US) on software quality requirements prioritisation. The study also aims to examine the alignment between the globally distributed project sites.

2.4. Global Software Development

GSD is defined as software development work undertaken across national boundaries at geographically separated locations. While teams are not colocated, they are still working towards a common goal with a commercially viable product. Work is commonly classified against two criteria:

1. *Insourcing/Outsourcing* defines whether-or-not the work is undertaken by employees of the organisation.
2. *Onshore/Offshore* defines whether-or-not the work is undertaken in the home country of the organisation.

GSD has become a common practice (Conchúir et al., 2009), with a number of benefits and risks. It allows companies to call upon a global talent pool to supplement a locally scarce resource pool (Wang et al., 2008). These specialised skills can have a positive impact on productivity and quality (Conchúir et al., 2009; Mockus and Herbsleb, 2001). Furthermore, many studies cite the ability to reduce development costs (Conchúir et al., 2009; Morgan, 2004; Sahay et al., 2003), and to focus on strategic business functions with day-to-day operations off-loaded (Wang et al., 2008).

The most commonly cited challenges of GSD relate to communication (Mockus and Herbsleb, 2001) and coordination (Herbsleb, 2007). Co-located teams have a greater opportunity to share formal and informal discussions, which have been found to be an effective way of creating a shared understanding in relation to what is expected in software quality. Geographic distance reduces a team's ability to communicate and collaborate (Noll et al., 2010). Less communication occurs and it uses less rich media (Herbsleb and Mockus, 2003). However, creating an environment that supports a team's ability to have shared knowledge of tasks will result in improved coordination (Espinosa et al., 2007). While new technologies have helped to reduce the barriers created by distance, it is emphasised that distance still matters (Carmel and Abbott, 2007).

When dealing with teams of people across the world, it is important to recognise that while each team may work well individually, it does not necessarily follow that they will be able to work well together (Loch and Kavadias, 2008).

Intercultural factors, such as power distance, individualism, and uncertainty avoidance (Hofstede, 1980) can be problematic in GSD settings. Studies on the impact of national and cross-cultural issues on systems development emphasise the need to take organisational and national culture seriously when working in these environments (Hofstede, 1980; Liang et al., 2010; MacGregor et al., 2005; Nicholson and Sahay, 2001) Furthermore, cultural issues can exacerbate exiting communication and coordination problems (Herbsleb and Moitra, 2001).

While much is made of the success of open-source communities developing software in distributed settings, Loch and Kavadias (2008) recognise that most successful open source projects use one of two approaches that help deal with the challenges of developing software in a GSD environment—either having a small close-knit management group or a very modular architecture. The benefits of a modular architecture have also been recognised as beneficial in the development of non-open-source software in GSD environments.

The challenges described in this section result in development work undertaking in GSD environments facing lower levels of productivity (Herbsleb and Mockus, 2003), reduced understanding of requirements (Espinosa et al., 2007) and increased numbers of bugs (Cataldo and Nambiar, 2009).

While the value of stakeholder alignment is clear, there is a complex myriad of factors that can stand in the way of software development organisations achieving this goal. Understanding the level of alignment and reasons for the level of alignment is of value to organisations, as this information can be used to support improvement initiatives.

2.5. Stakeholder Alignment Assessment Method for Software Quality

The Stakeholder Alignment Assessment Method for Software Quality (SAAM-SQ) provides a way to determine the level of alignment within and between key stakeholder groups when it comes to the priorities placed on different aspects of software quality (Barney and Wohlin, 2009). The method has successfully been used in relation to alignment between success-critical stakeholder groups in one location for one company. This paper contributes by applying the method in a GSD context at the previously studied company, and also for a new company that has not been previously studied as part of this work. This helps ensure that the method is not only suitable for the company where it was originally developed and applied. Using the output from the method, it is possible to determine where individuals or groups are not aligned and do not share a common set of priorities. This allows a more focused study to be undertaken to understand the factors that are not supporting the alignment of priorities.

3. Methodology

This paper aims to study alignment with respect to software quality within and between groups involved in the development of software products in GSD environments. Software quality and stakeholder alignment are both key to the successful development of products in GSD settings. Using the SAAM-SQ method, it is possible to quantify the levels of alignment within and between success-critical stakeholder groups. This paper aims to use this method to help identify benchmark figures that can support future work by researchers and practitioners working in this area.

With an understanding of the levels of alignment, this paper aims to understand the challenges seen in practice that limit the ability of organisations to achieve alignment within and between groups. An understanding of the challenges faced by practitioners will help other practitioners and researchers focus on likely problem areas.

Furthermore, this paper presents a growing body of experience applying SAAM-SQ in industry settings. Thus the authors believe it is timely to reflect upon the factors that support and limit the successful application of this method.

3.1. Research Questions

The aims of this paper have been translated into three research questions. The first question relates to the prerequisites to make the Stakeholder Alignment Assessment Method for Software Quality useful in industrial settings. The answer to this question is driven by the two questions identified from industry and put forward in Section 1. Thus, the following research questions are addressed:

- **RQ 1.** What factors have to be in place to succeed in using the Stakeholder Alignment Assessment Method for Software Quality (SAAM-SQ) in industrial settings?
- **RQ 2.** To what degree are success-critical stakeholder groups aligned between roles in offshore software product development on the priorities given to aspects of software product quality?
- **RQ 3.** What are the reasons for the level of alignment identified between these groups?

3.2. Method

SAAM-SQ was employed to answer RQ2 and RQ3 and its repeated application is reflected upon in answering RQ1. SAAM-SQ was previously developed and used by the authors to determine the level of alignment between key stakeholder groups in Barney and Wohlin (2009, 2010). Given the success of this method to meet the aims of this research it is reused.

The method draws on Theory-W (Boehm and Jain, 2006), which aims to create win-win scenarios by getting success-critical stakeholder groups to agree on how product development should proceed. The first steps of Theory-W are to identify the success-critical stakeholders, identify how they want to win, negotiate plans to achieve win-win scenarios and control the process to achieve these scenarios.

The seven main steps of the SAAM-SQ method used to answer the research question are described below.

3.2.1. Select a company and product

As each product can have different quality requirements and quality expectations by success-critical stakeholders, it is essential to ensure the study is sufficiently focused.

3.2.2. Identify success-critical stakeholder groups

Success-critical stakeholder groups, are groups of people upon whom the success of the product depends—for example product managers and developers. Identifying these groups ensure that critical perspectives are not lost, while less important perspectives cannot dominate.

3.2.3. Develop a quality model

Literature addressing software quality recognizes that quality depends on both the perspective of the observer and the actual software product in question. That is quality will be defined differently by different people, and quality will be defined differently for different products. As such, using any model of software quality as it appears in the literature risks not adequately defining quality in the context being studied. To use one of the quality models mentioned in Section 2.2 is a good starting point, but company and product specific needs must be taken into account (Barney and Wohlin, 2009).

3.2.4. Develop a questionnaire

This method proposes to use the hierarchical cumulative voting technique (HCV) (Berander and Jönsson, 2006) to elicit the priorities given the various aspects of software product quality. This method allows respondents to state the relative importance of the aspects being studied. Past research has shown that respondents have trouble comparing some aspects of software product quality at a low level directly (Barney and Wohlin, 2009); and HCV provides a method for breaking the problem into a series of smaller direct comparison exercises, with a method to join these results back together.

In this research participants are asked to complete the HCV exercise twice. First they are asked to describe the priorities as they understand them today, then they are asked to describe their perceived *ideal* set of priorities for the situation today.

3.2.5. Conduct the questionnaire

The questionnaire should be completed by representatives of each of the identified success-critical stakeholder groups. Doing this in a one-on-one structured interview allows richer information to be collected from the participants, providing greater understanding of the results obtained. This also helps ensure that participants have a common understandings of the questionnaire with the interviewer being able to assist with questions or problems faced by the respondents.

3.2.6. Analyse the results

It is possible to transform the results of the HCV exercise into cumulative voting (CV) results (Berander and Jönsson, 2006). From here it is possible to group the results by success-critical stakeholder group, and calculate the mean number of points awarded to each aspect of software product quality. These results can be used to calculate Spearman rank correlation coefficients and corresponding p-values (McDonald, 2009), determining the level of alignment between the success-critical stakeholder groups. Spearman rank correlation coefficients can range from one (1) to minus one (-1). The closer a result for two groups is to one, the higher the level of agreement between the two groups in the ranking given to aspects studied. The corresponding p-values for the Spearman rank correlation coefficients indicate the presence of a significant relationship between the priorities of the two groups.

To determine the degree to which members of a group were aligned, the inertia and the significance of the chi-square test (p-value) of each group was calculated. The closer an the inertia value is to zero, the more aligned the member of the group in their priorities. The corresponding p-values for the inertia can indicate a significant level of alignment within a group. The process is described in Section 3.3.

3.2.7. Workshop the results

Finally the results should be presented to participants from each of the success-critical stakeholder groups, asking them for their response to the following questions:

- Do these results look reasonable?
- Do the differences make sense?
- Why do these differences exist?

3.3. Determining alignment within a group

Inertia measures the variation in a dataset (Greenacre, 1984). It uses contingency tables, where two categorical variables are cross-tabulated and the chi-square statistic is computed to test the hypothesis of their independence. The inertia is computed by dividing the chi-square statistic by the total number of cases that are distributed in the cells of the contingency table.

For this method each group of stakeholders is considered separately. Given a group of S stakeholders and a set of I aspects, it is possible to denote n_{ij} as the amount allocated to each aspect ($j = 1, ..., I$) by the stakeholder (i=1,...,S) in the cumulative voting results. Thus, the mean value of the amounts allocated by each stakeholders within a group to issue j can be defined as:

$$\overline{n}_j = \frac{1}{S}\sum_{i=1}^{S} n_{ij} \tag{1}$$

Inertia checks the level of variation between the responses of the stakeholders in a group. When there is little variation between members of the group, all of the members prioritize the aspects in more or less the same way, which is very close to the mean values calculated by Equation 1. The next step to calculating the inertia is to calculate the chi-square statistic for the group being studied:

$$\chi^2 = \sum_{i=1}^{S}\sum_{j=1}^{I} \frac{(n_{ij} - \overline{n}_j)^2}{\overline{n}_j} \tag{2}$$

The chi-square statistic tests the significance of the variation within the group. The test used is the common chi-square test, which compares the value computed in Equation 2 with the critical value of the theoretical chi-square distribution with $(S-1)\times(I-1)$ degrees of freedom. If the significance of the test is < 0.05, then we can infer that the stakeholders within a group have significant variations in the priorities given to the aspects studied.

In order to compare between different groups it is necessary to normalize the result given by Equation 2 as this result is dependent on the number of stakeholders and aspects studied. This can be achieved by dividing the chi-square result by the total number of points allocated by all stakeholders to all aspects within the group studied:

$$inertia = \frac{\chi^2}{\sum_{i=1}^{S}\sum_{j=1}^{I} n_{ij}} \tag{3}$$

As there are S stakeholders and each stakeholder must use 1000 points across all the aspects on which they vote then, the total number of points allocated by all stakeholders to all aspects within the group studied is $S \times 1000$:

$$\sum_{i=1}^{S}\sum_{j=1}^{I} n_{ij} = S \times 1000 \tag{4}$$

Thus it is possible to simply the inertia equation to:

$$inertia = \frac{\chi^2}{S \times 1000} \tag{5}$$

Inertia values for each group can be compared. If one group has a larger inertia value it signifies that there is a higher degree of variability within the group.

The inertia measure is part of a multivariate analysis methodology, called *correspondence analysis* (Greenacre, 1984). Correspondence analysis provides an interpretation approach for illustrating the inertia results. As the name of the method suggests, it is a way to explore the “system of associations” between the elements of two sets. Correspondence Analysis is a statistical visualization method for picturing the associations between the categorical variables of a two-way contingency table. These relationships are described by projecting the values of the variables as points on a two-dimensional space, in such a way that the resulting plot describes simultaneously the relationships between the variables. For each variable, the distances between points in the plot reflect the relationships between them.

Correspondence analysis can be used to further explore the relationships between stakeholders alignment (Greenacre, 1984). The method can be adapted to SAAM-SQ if the rows of the contingency table define the stakeholders and the columns define the issues. The observed frequencies within each cell are represented by the amount of units that a stakeholder assigned to an issue. From the contingency table we can compute the inertia value and moreover to visualize the relationships between stakeholders.

The stakeholders are represented by points in a two-dimensional space. The distance between the stakeholders' points is a measure of similarity between the stakeholders' profiles.

4. Case Studies

This section describes the three case studies included in this paper. An overview of these cases are presented in Table 1. The remainder of this section describes the three case studies, with *Case Study 1* and *Case Study 2* presented in Section 4.1 and *Case Study 3* presented in Section 4.2.

4.1. Ericsson

Case Study 1 and *Case Study 2* examine two major products developed by Ericsson, the Swedish-based telecommunication company. Ericsson is a

Table 1: Overview of Case Studies

	Case Study 1	Case Study 2	Case Study 3
Company	Ericsson	Ericsson	'Global Finance'
Industry	Telecommunications	Telecommunications	Financial Services
Product	'Product 1'	'Product 2'	'TravelBill'
Locations	India	China	America
	Sweden	India	Australia
		Sweden	India

world leading company in telecommunications, employing over $90,000$ people. They provide a wide range of products and services. These are developed and sold as generic solutions, although customized versions of the products are also developed for key customers.

4.1.1. Products

The authors collaborated with a group of managers at Ericsson in India and Sweden to study the alignment between success-critical stakeholder groups described by the "offshore insourcing" scenario for two products. Both products are leading solutions for the market they service. These products are referred to as *Product 1* and *Product 2* throughout this paper for reasons of confidentiality.

Product 1 has been developed and sold by Ericsson continuously over more than 10 years. The product is developed at Ericsson sites in India and Sweden. This paper represents *Phase 3* of this research on *Product 1*. *Phase 1* examined the alignment between success-critical stakeholder groups employed by Ericsson on *Product 1* and based in Sweden (Barney and Wohlin, 2009). *Phase 2* expanded this work to include subcontracted developers based out of the development centre in Sweden (Barney and Wohlin, 2010). The third phase, first presented in Barney et al. (2011b) and extended in this paper, examines the alignment between success-critical stakeholder groups employed by Ericsson on *Product 1* and based in India—furthermore, a couple of Swedish-based *Strategic Product Managers* and *Technical Product Managers* are included in the research. The relationship between these studies is shown in Figure 1.

Product 2 has been under development for approximately two years and sold to customers for approximately one year. This product is developed at sites in China, India and Sweden. It has not been previously studied as part

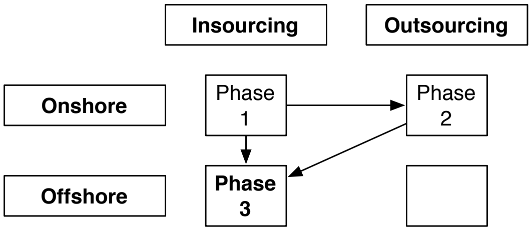

Figure 1: Previous Research on Product 1

of this work.

The focus of the research presented in this paper on *Product 1* and *Product 2* will be on the alignment between success-critical stakeholder groups for the development work based in India for both products. As with *Product 1*, a couple of Swedish-based *Strategic Product Managers* and *Technical Product Managers* are included in the research. For reasons of confidentiality, some product details and study results have been withheld.

4.1.2. GSD Strategy

GSD practices are employed with both products at Ericsson using the same distribution model. Work is split between the sites using a module-based approach. Sets of modules that make up the product are distributed to each site, reducing the coupling and cohesion between the sites. However, development of individual modules has been moved between sites to maximise the use of available resources.

4.1.3. Success-critical Stakeholder Groups

One of the authors worked with three senior managers to identify success-critical stakeholder groups at Ericsson. As the same organizational structure is employed for both *Product 1* and *Product 2*, the same roles and responsibilities were identified for both products. The *identified groups* are:

- *Architects* are responsible for the overall product design, assigning requirements to modules of the product.
- *Developers* are responsible for the implementation of requirements.
- *Product Support* provides support to users and product owners at customer sites.

- *Project Management* is responsible for planning and executing projects aligned with the priorities of the strategic product management.
- *Strategic Product Management* has the strategic product responsibility and decides the overall product development direction.
- *Tactical Product Management* supports the strategic product management with expert knowledge of the systems and their architecture. It is also responsible for providing analysis of pre-project requirements in the form of feasibility, impact and technical dependencies.
- *Testing* are responsible for the verification and validation of requirements.

The number of candidates identified by Ericsson and the number of interviewees for each group is shown in Table 2 for both products. In total, the number of candidates for Product 1 is 33 and the number of interviewees is 27, and for Product 2 the number of candidates are 26 and the number of interviewees is 22. Thus, it was possible to interview a large number of the identified candidates. For example, two *Tactical Product Managers* were identified for each product, but only one was interviewed for *Product 1*. Due to limited research time in India by the first author, a number of interviews were cancelled due to travel, sickness or unplanned leave. Most groups contained more members than the list of candidates provided by Ericsson, but in these cases the people that supplied the list of candidates aimed to provide a representative sample. It was anticipated that some people would not be able to attend an interview during the short time in India, so extra names were collected.

Table 2: Case Study 1 and Case Study 2 Candidates and Respondents

Role	Abbrev.	Product 1	Product 2
Architects	Arch	5/ 5	3/4
Developers	Dev	7/10	6/7
Product Support	PS	2/ 2	1/1
Project Managers	PM	3/ 4	3/3
Strategic Product Managers	SPM	2/ 2	3/4
Tactical Product Managers	TPM	1/ 2	2/2
Testers	Test	7/ 8	4/5

4.1.4. Model and Questionnaire

Given *Case Study 1* expands two previously conducted case studies of *Product 1* (Barney and Wohlin, 2009, 2010) the software quality model previously used was reviewed by a group of senior managers and selected for reuse in this case study. The managers deemed the model appropriate for both *Product 1* and *Product 2*. Copies of the quality model, including terms and definitions, and questionnaire used for these case studies are available online (Barney et al., 2011a). A summary of the aspects of quality used in each CV exercise is listed in Table 3.

The questionnaire was completed in an interview setting with the first author. All interviews were conducted in a one-on-one interview situation. Respondents were first introduced to the aims of the study and then an explanation was provided on how to complete the questionnaire. Participants were invited to share any comments or feedback with the author, and upon completing the questionnaire the interviewer reviewed and discussed the respondents completed questionnaire to confirm and further understand the results. Almost all interviews were conducted face-to-face, with the exception of a couple of *Strategic Product Managers* who selected to have a phone interview with screen sharing due to conflicting commitments. Interviews took on average 45 minutes.

4.2. Global Finance

Case Study 3 was conducted at a multinational financial institution. For reasons of confidentiality the company will be referred to by the pseudonym *Global Finance*.

Global Finance provides a wide range of payment, expense management and travel solutions. Its customers range from individual consumers to other multinational corporations. Global Finance is based in the US and employs approximately $60,000$ people globally. It operates throughout the North American, South American, European, African, Middle Eastern and Asia-Pacific regions.

Specifically this case study focuses on a product that is intended to replace and extend several back-end billing and bill management systems for travel services provided by Global Finance. This product is referred to as *TravelBill* in this paper.

The TravelBill development project follows Global Finance's methodology for large projects. This methodology is loosely-based on the waterfall

Table 3: Case Study 1 and Case Study 2 CV exercises and quality attributes

Exercise	Attributes
1. Features	Suitability
	Accuracy
	Interoperability
	Compliance
2. System Properties	Security
	Scaleability
	Maturity (Reliability)
	Recoverability (Reliability)
	Fault tolerance (Reliability)
	Learnability (Usability)
	Understandability (Usability)
	Operability (Usability)
	Time behaviour (Efficiency)
	Resource behaviour (Efficiency)
	Stability (Maintainability)
	Performance Management (Maintainability)
	Analysability (Maintainability)
	Changeability (Maintainability)
	Testability (Maintainability)
	Installability (Portability)
	Replaceability (Portability)
	Adaptability (Portability)
3. Project Management	Time
	Cost
4. Overall	Features
	System Properties
	Project Management

development methodology with overlapping phases—for example, the design phase will still be underway when the build phase begins.

Furthermore, the product is being rolled out in a series of markets following a phase-based approach. The first phase was to roll this product out in the US market. While the US rollout is underway the Australian rollout was started. The case study was conduct after the Australian rollout began. Further rollouts in other countries are planned for the near future.

The Global Finance TravelBill product was chosen to be the subject of this case study as it was aligned with the objectives of the study and one of the authors had access to the company as a business analyst on the project studied.

4.2.1. Success-Critical Stakeholders

The success-critical stakeholders for TravelBill were identified in collaboration with a high-level manager involved in the product development and classified into three perspectives:

- The *business perspective* covers business users and system sponsors. Everyone in this group is an internal employee of Global Finance.
- The *product perspective* includes business analysts, architects, developers and testers. The business analysts and architects are both internal employees and external contractors. Developers all come from IndiaSoft.
- The *project perspective* covers project managers. This role is fulfilled by a mixture of internal employees and external contractors.

IndiaSoft provides outsourced software development services. It was founded in India, employs over $160,000$ employees and operates in 42 countries. Global Finance has used IndiaSoft's services for approximately 10 years, during which time they have developed a good working relationship. While Global Finance has relationships with other software development providers, IndiaSoft was selected to work on TravelBill due to their prior experience with the current legacy system that TravelBill is replacing and other travel related software projects.

4.2.2. Development Sites

When the case study was conducted three sites were involved with the product:

- **Global Finance, America (US)** is the main site for Global Finance. It hosts people from the business, product and project perspectives. Everyone is co-located within one building, however, the business stakeholders are situated in a different part of the building to the other groups.

- **Global Finance, Australia (AU)** closely resembles the American site in both the roles represented, and their physical location within the building.
- **IndiaSoft, India (IN)** hosts product and project stakeholders from IndiaSoft that are contracted to Global Finance.

The first phase of product development aimed to rollout the product in the American site. This phase, however, involved the American, Australian and Indian sites. This phase of the project took much longer, and cost much more than expected. The second phase was underway at the time of the study, which aimed to rollout the product to the Australian site.

4.2.3. GSD Strategy

The GSD strategy employed by Global Finance for the TravelBill project differs to that used by Ericsson. Global Finance has subject matter experts, business analysts and project managers at each business site, with the core development work outsourced and offshored to IndiaSoft. This approach allows Global Finance to focus on their core strengths and employ experts to undertake non-core activities. This approach, however, increases coupling and cohesion between the sites as they must work together to define, develop and test requirements as part of the system being developed.

4.2.4. Quality Model

Global Finance does not use any specific models to describe software quality, and none had been defined for the product or projects prior to our study.

Given the absence of a quality model, the authors proposed the use of the ISO9126 (2001) quality model as a foundation. The second author then worked with two senior managers from the product and project perspectives to tailor it to the specific needs of the product over two one-hour discussions. This is a similar approach as for the quality model used at Ericsson in *Case Study 1* and *Case Study 2* (Barney and Wohlin, 2010). Based on the feedback from the two senior managers on the ISO 9126 quality model the following changes were made:

- *Security* was raised one-level in the hierarchy and given the sub-attributes *Security Accuracy* and *Security Compliance*. The managers reviewing the model perceived the profile of security for both the product and the company meant it should be a top-level attribute.

- The sub-attribute of *Compliance* existed for each of the top-level aspects of quality. These were removed to avoid confusion, with the exception of security where it was still perceived necessary.
- The sub-attribute *Conformance* was added under *Portability*.

4.2.5. Participants

The number of people involved with the Product during the study is shown in Table 4. This table breaks down the number by site and perspective. The number of people representing the business perspective changes with demand, so minimum levels have been provided.

Table 4: Case Study 3: Product Stakeholders by Site and Role

Perspective	America	Australia	India
Business	> 15	> 10	> 7
Product	11	7	12
Project	4	2	1

4.2.6. Questionnaire

A questionnaire was created using the SAAM-SQ method and the quality model developed for *Case Study 3*. Copies of the quality model, including terms and definitions, and questionnaire used for this case study are available online (Barney et al., 2011a).

Questionnaires and instructions on how to complete them were sent out to 44 candidate participants via email. In total 28 responses were received—Table 5 shows a breakdown of the returned results by site and perspective.

Table 5: Case Study 3: Survey Participants

Perspective	America	Australia	India	Total
Business	1	3	0	4
Product	10	6	4	20
Project	2	2	0	4
Total	13	11	4	28

4.2.7. Interviews

In addition to the SAAM-SQ process, interviews were also conducted with selected participants to help gain a deeper understanding. Interviewees were selected to create a diverse group in terms of both perspective and location. Where possible preference was given to people who had been working on the product longer, as it was thought these people would have more experiences from which to draw during the interview. A breakdown of the interviewees by site and perspective is shown in Table 6.

Table 6: Case Study 3: Interview Participants

Perspective	America	Australia	India	Total
Business	1	1	1	3
Product	4	1	0	5
Project	0	1	0	1
Total	5	3	1	9

Interviews were conducted in a one-on-one setting between the author working at the Global Finance and the interviewees. Interviews in Australia were conducted face-to-face during office hours, while interviews with people in America and India were conducted over the phone at a time convenient to both the interviewer and interviewee. Interviews lasted approximately 30 minutes each. Each interview was recorded, transcribed and emailed to the interviewee for validation.

5. Alignment Levels

This section answers RQ2, which aims to determine the level of alignment between success-critical stakeholder groups in the priorities given to aspects of software product quality. Each case study is presented individually in the following sections. These results are based on the questionnaires described in Section 4.1.4 for *Case Studies 1 and 2* and Section 4.2.6 for *Case Study 3*.

5.1. Case Study 1

The alignment levels for the priorities on quality today in *Case Study 1* are presented in Table 7 as Spearman rank correlation coefficients for the situation as it is understood today, with p-values less than 0.05 indicated with one asterisk, p-values less than 0.01 with two asterisks and p-values less than 0.001 indicated with three asterisks.

Table 7: Case Study 1: Spearman Rank Correlation Coefficients—Situation Today

	Arch	Dev	PS	PM	SPM	TPM	Test
Arch	1.00***	0.18	0.35	0.16	0.39	0.03	0.15
Dev		1.00***	0.42*	0.30	0.23	0.24	0.42*
PS			1.00***	0.68***	0.52**	−0.09	0.53**
PM				1.00***	0.26	−0.12	0.23
SPM					1.00***	0.26	0.35
TPM						1.00***	0.30
Test							1.00***

The results for *Case Study 1* show lower levels of alignment than seen in the previous case studies of this same product when describing the situation *today*. References to the previous work are provided in Section 4.1.1. At the $p < 0.05$ level, there is a significant relationship between *Product Support* and a number of other roles: *Developers*, *Project Managers*, *Strategic Product Managers* and *Testers*. It was said that this could be related to the requirement for *Product Support* to liaise with the other roles in the identification and resolution of issues, giving them a cross-functional understanding of the product. There is also a significant relationship between the priorities of the *Developers* and *Testers*, who are located together within agile teams. The levels of alignment between the *Tactical Product Managers* and other roles are low.

The alignment between success-critical stakeholders in the perceived *ideal* situation, for *Case Study 1* are presented in Table 8. The results show lower levels of alignment between the groups when compared to the onshore insourcing case study of the same product, however, the results are inline with those seen in the onshore outsourcing case study. Furthermore, it should be noted that in both of the previous case studies lower levels of alignment were seen in the *ideal* situation when compared with the situation *today*. This has been put down to different groups seeing different problems of quality that related specifically to their role (eg. *Testers* wanting a greater emphasis on the *testability* quality attribute than the other groups). In this study no groups were found to be aligned at the $p < 0.05$ level.

Table 9 provides the inertia and the significance of the chi-square test (p-value) in *Case Study 1* for all stakeholders (overall) and for each group separately. These results show that the variability within each group studied is statistically significant at the $p < 0.05$ level, suggesting low levels of align-

Table 8: Case Study 1: Spearman Rank Correlation Coefficients—Ideal Situation

	Arch	Dev	PS	PM	SPM	TPM	Test
Arch	1.00***	−0.27	0.30	0.02	−0.06	0.19	0.14
Dev		1.00***	0.10	0.21	0.39	−0.17	0.36
PS			1.00***	0.27	0.08	0.29	0.40
PM				1.00***	0.00	0.17	0.23
SPM					1.00***	0.08	0.19
TPM						1.00***	0.37
Test							1.00***

ment between the stakeholders in each group. The inertia was not calculated for the *Tactical Product Managers*, as only one person was in this group and thus, by definition, was in complete alignment. These results suggest that the members of each group do not agree on the priorities on quality for the product.

Table 9: Case Study 1: Inertia and p-values for Each Group

	Today	Ideal
	Inertia	Inertia
Overall	0.28***	0.22***
Arch	0.26***	0.19***
Dev	0.21***	0.15***
PS	0.36***	0.39***
PM	0.18***	0.21***
SPM	0.09***	0.05***
TPM	—	—
Test	0.15***	0.08***

The inertia is also shown visually using correspondence analysis in Figure 2 for the situation today, and in Figure 3 for the ideal situation. Each point represents one stakeholder with the distance between any set of points indicating the level of alignment.

These figures show that the stakeholders are not generally aligned in their views since the corresponding points are quite dispersed. For example, the group with the lowest level of alignment today is Product Support (PS), with an inertia value of 0.36. Figure 2 shows that this group consists of two members, who are not well aligned on the priorities placed on software

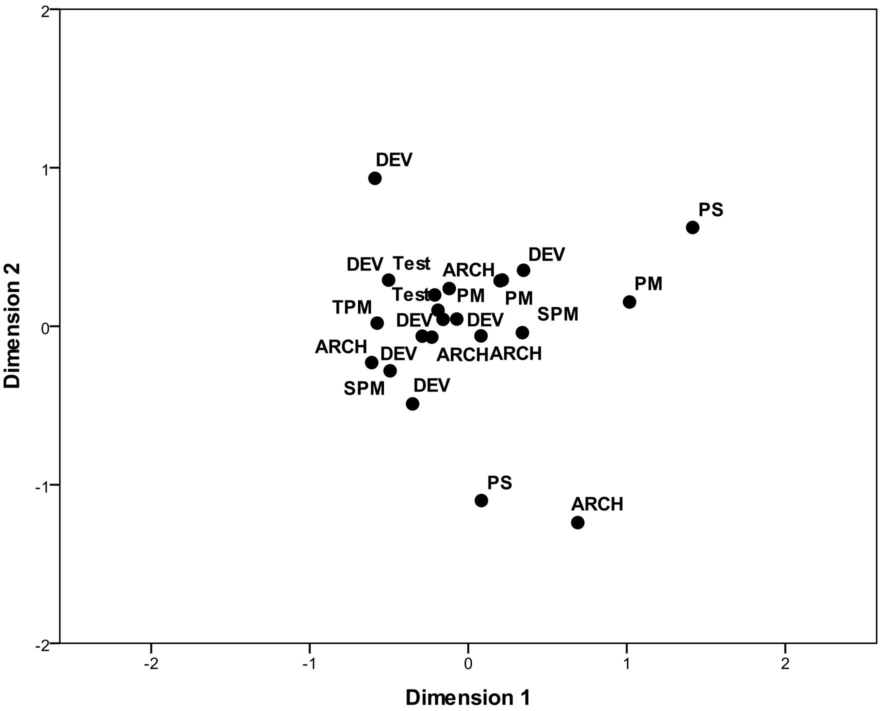

Figure 2: Case Study 1: Correspondence Analysis—Situation Today

quality—with the two stakeholders placed far apart in the figure. Furthermore both appear to be outliers from the general community for this product. This contrasts with Testers (Test), whose closer alignment is displayed in the figures with tighter clustering of the points.

It is also possible to compare the results from the participants descriptions of the situation *today* and the *ideal* situation, giving the level of change they would like to see on the priorities given to the aspects studied. The results for both products are presented in Table 10, with Spearman rank correlation coefficients and corresponding p-values. The Spearman rank correlation coefficients range from one (1) to minus one (-1). The greater the deviation from one, the more change desired by that group. P-values have also been calculated.

The results for *Product 1* show that all groups perceive some need for change. However, there is a significant relationship between the priorities as they are understood in the situation today and in the ideal situation when looking at most groups—*Overall*, *Product Support*, *Project Managers*, *Strate-*

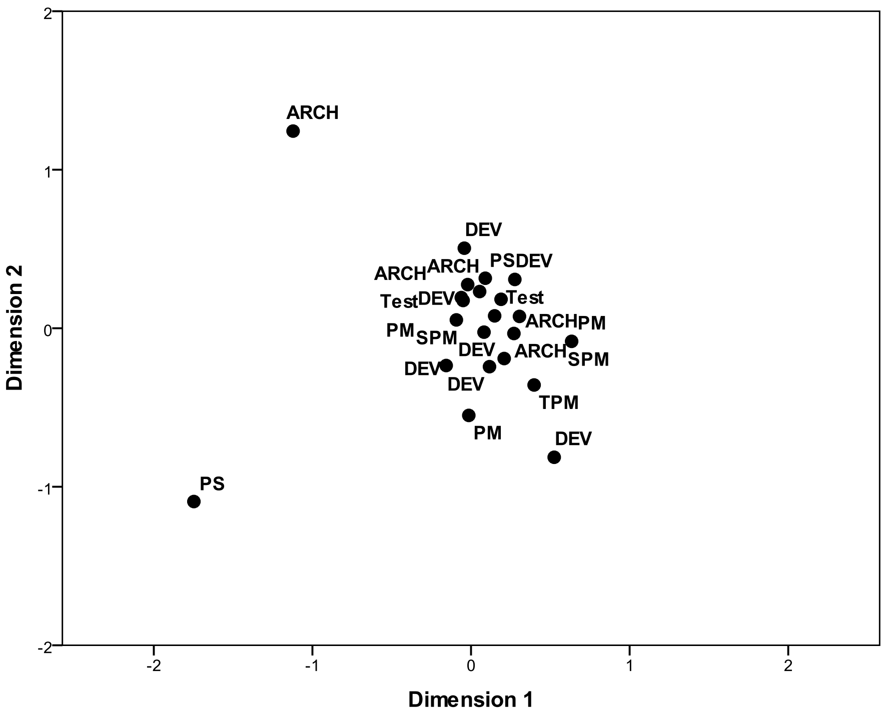

Figure 3: Case Study 1: Correspondence Analysis—Ideal Situation

Table 10: Case Study 1: Desired Change Correlation Coefficients and p-values

	Correlation
Overall	0.43*
Arch	−0.18
Dev	0.06
PS	0.41*
PM	0.77***
SPM	0.56**
TPM	0.51*
Test	0.26

gic Product Managers and *Tactical Product Managers*. The groups desiring the greatest level of change in priorities are the *Architects* and *Developers*.

5.2. Case Study 2

The level of alignment between success-critical stakeholder groups in *Case Study 2* for the perceived situation *today* are presented in Table 11. Overall the results show moderate levels of alignment. At the $p < 0.05$ level, there is a significant relationship between the priorities of the *Testers* and five other roles—*Architects*, *Developers*, *Strategic Product Managers* and *Tactical Product Managers*. The reason identified in the workshop for this result is that *Testers* are required to have a functional and technical understanding of the product and through project management are well aware of time and cost limitations, while other roles can be more isolated from some aspects of the product. A significant relationship is also seen between the priorities of the *Developers* and *Project Managers*.

Table 11: Case Study 2: Spearman Rank Correlation Coefficients—Situation Today

	Arch	Dev	PS	PM	SPM	TPM	Test
Arch	1.00***	0.30	0.01	0.32	0.22	0.39	0.47*
Dev		1.00***	0.03	0.42*	0.39	0.29	0.45*
PS			1.00***	0.08	0.23	0.15	0.22
PM				1.00***	0.36	0.09	0.32
SPM					1.00***	0.23	0.64*
TPM						1.00***	0.48*
Test							1.00***

The alignment levels in *Case Study 2* for the perceived *ideal* situation are presented in Table 12. A significant relationship between the priorities was found at the $p < 0.05$ level between *Testers* and four other roles—*Developers*, *Product Support*, *Strategic Product Managers* and *Tactical Product Managers*. Significant alignment is also seen between *Developers* and *Architects*; *Product Support* and *Strategic Product Managers*; and *Strategic Product Managers* and *Project Managers*.

Table 13 provides the inertia and the significance of the chi-square test (p-value) for *Product 2* for all stakeholders (overall) and for each group separately. These results show that the variability within each group studied is statistically significant at the $p < 0.05$ level, with the exception of the *Project Managers* in their description of the ideal situation. This indicates that the members of most groups do not agree on the priorities on quality for the product. The inertia was not be calculated for *Product Support* as

Table 12: Case Study 2: Spearman Rank Correlation Coefficients—Ideal Situation

	Arch	Dev	PS	PM	SPM	TPM	Test
Arch	1.00***	0.66***	−0.11	0.30	0.24	0.38	0.37
Dev		1.00***	0.07	0.40	0.05	0.23	0.49*
PS			1.00***	0.34	0.52**	0.25	0.50*
PM				1.00***	0.42*	0.36	0.27
SPM					1.00***	0.29	0.49*
TPM						1.00***	0.58**
Test							1.00***

only one person was in this group and thus, by definition, was in complete alignment.

Table 13: Case Study 2: Inertia and p-values for Each Group

	Today	Ideal
	Inertia	Inertia
Overall	0.26***	0.21***
Arch	0.36***	0.29***
Dev	0.19***	0.14***
PS	—	—
PM	0.08***	0.02***
SPM	0.19***	0.12***
TPM	0.08***	0.06***
Test	0.16***	0.19***

The inertia is also shown visually in Figure 4 and Figure 5 for the situation today and the ideal situation respectively.

It is noticeable that the Project Managers (PM) and Tactical Product Managers (TPM), the groups most closely aligned for both the situation today and ideal situation, are tightly clustered in the figure. This result implies agreement regarding their views on software quality aspects.

This result contrasts most noticeably with the result for the Architects (ARCH) perception of the situation today, with all stakeholders far apart from each other and appearing to be outliers.

Moreover, Testers' points seem to be close but in both perceptions one seems to be an outlier and is isolated from the other Testers regarding their views on software quality aspects.

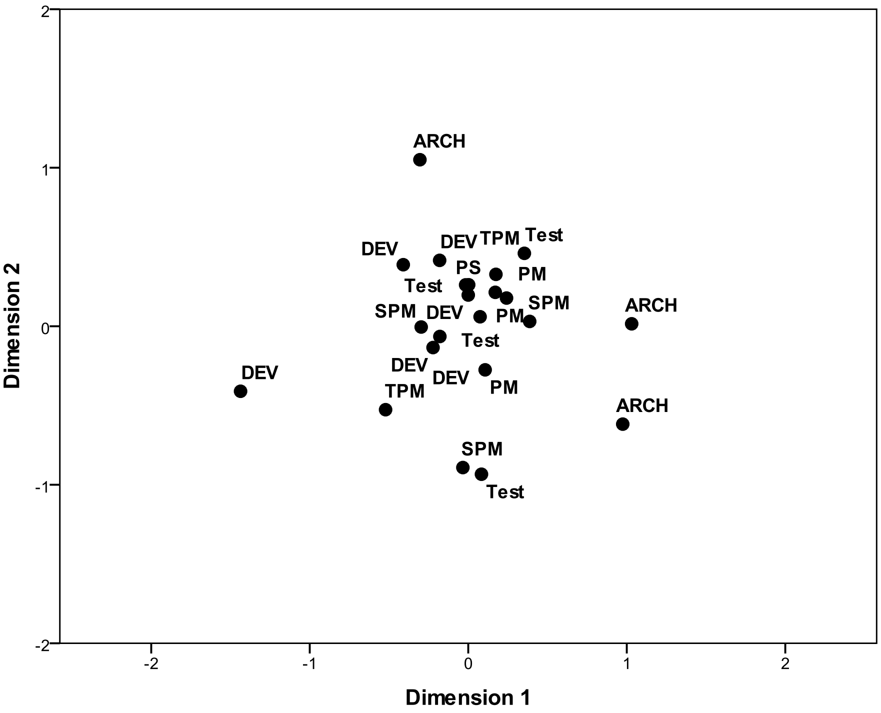

Figure 4: Case Study 2: Correspondence Analysis—Situation Today

Finally, Table 14 shows that while some change is desired by each group, the level of desired change is small. There is a significant relationship between the priorities *today* and the perceived *ideal* priorities for each group at the $p < 0.05$ level.

Table 14: Case Study 2: Desired Change Correlation Coefficients and p-values

	Change Correlation
Overall	0.65**
Arch	0.69***
Dev	0.69***
PS	0.76***
PM	0.60**
SPM	0.65**
TPM	0.41*
Test	0.73***

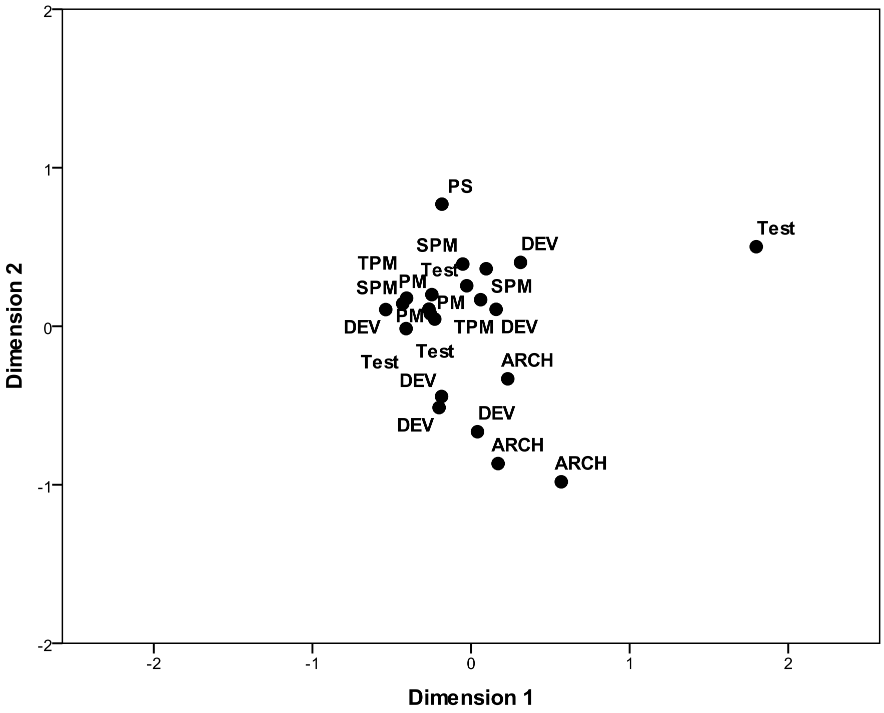

Figure 5: Case Study 2: Correspondence Analysis—Ideal Situation

5.3. Case Study 3

The anonymity of *Case Study 3* allows the prioritised lists of qualities to be published. The ranks given to each aspect in the situation as it is perceived *today* are presented in Table 15 for each location and overall.

The ranks given to each aspect in the perceived *ideal* situation for each location and overall are presented in Table 15.

During the interview process reasons where given by representatives of the success-critical stakeholders

The level of alignment between sites in *Case Study 3* for the situation as it is perceived today is presented in Table 17. The results for the situation today show the American and Australian sites to be reasonably aligned, and show the Australian and Indian sites to be reasonably aligned. However, the American and Indian site are not well aligned. The Indian site gave a much higher priority to *maturity*, *recoverability*, *standardisation/compliance* and *learnability* than the American site. A much lower priority was given by the Indian site to *adaptability*, *time*, *changeability*, *installability* and *testability*

Table 15: Case Study 3: Ranking of Qualities—Situation Today

Today	All	AU	IN	US
Cost	1	2	1	1
Time	2	1	2	2
Suitability	3	3	13	3
Time behaviour	4	6	3	5
Security Compliance	5	5	6	6
Accuracy	6	14	18	4
Understandability	7	4	9	15
Security Accuracy	8	8	12	7
Interoperability	9	7	16	8
Operability	10	10	6	10
Maturity	11	13	4	22
Learn-ability	12	9	8	20
Recoverability	13	11	5	23
Standardisation/Compliance	14	17	10	12
Installability	15	15	23	11
Changeability	16	19	19	9
Analysability	17	12	19	16
Adaptability	18	16	21	14
Fault tolerance	19	25	11	17
Testability	20	21	23	13
Attractiveness	21	23	14	18
Stability	22	22	17	19
Resource behaviour	23	20	15	21
Conformance	24	18	22	25
Replaceability	25	24	25	24

Table 16: Case Study 3: Ranking of Qualities—Ideal Situation

Ideal	All	AU	IN	US
Time	1	2	1	1
Cost	2	1	2	2
Accuracy	3	5	8	3
Time behaviour	4	4	4	7
Suitability	5	3	12	5
Security Compliance	6	6	9	8
Security Accuracy	7	8	13	4
Operability	8	11	6	11
Recoverability	9	9	5	19
Understandability	10	7	10	16
Maturity	11	15	3	14
Learn-ability	12	10	11	20
Fault tolerance	13	21	7	15
Standardisation/Compliance	14	13	14	10
Adaptability	15	19	20	6
Stability	16	20	18	9
Interoperability	17	14	15	13
Resource behaviour	18	12	17	18
Changeability	19	16	22	12
Conformance	20	17	16	22
Analysability	21	18	22	21
Testability	22	22	20	23
Installability	23	23	22	17
Attractiveness	24	24	19	24
Replaceability	25	25	25	25

than the American site. The Australian site, however, was placed somewhere between the American site and Indian site on all these issues.

Table 17: Case Study 3: Spearman Rank Correlation Coefficients and p-values—Situation Today

	AU	IN	US
AU	1.00***	0.64***	0.64***
IN		1.00***	0.32
US			1.00***

Similarly the alignment between the sites in how they describe the ideal situation is shown in Table 18. The results for the perceived ideal situation show all of the sites to be in a reasonable level of alignment on the prioritization of software quality issues.

Table 18: Case Study 3: Spearman Rank Correlation Coefficients and p-values—Ideal Situation

	AU	IN	US
AU	1.00***	0.75***	0.70***
IN		1.00***	0.53**
US			1.00***

The level of alignment within each site is shown through the inertia values in Table 19. The results show the groups to be less aligned than in *Case Study 1* and *Case Study 2* for the situation as it is perceived today.

Table 19: Case Study 3: Inertia and p-values for Each Group

Site	Today	Ideal
Overall	0.431***	0.271***
Australia	0.548***	0.278***
India	0.205***	0.157***
USA	0.349***	0.215***

The inertia is shown visually in Figure 6 and Figure 7 for the situation today and the ideal situation respectively.

In both perspectives, Indian stakeholders have the lowest inertia value among the three groups, indicating the highest level of internal alignment.

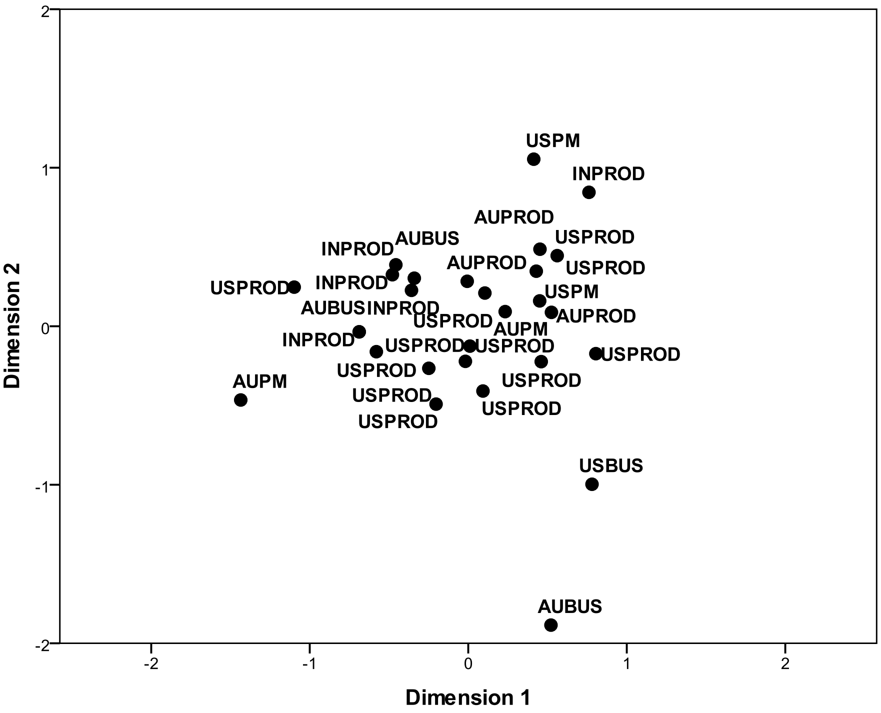

Figure 6: Case Study 3: Correspondence Analysis—Situation Today

As can be seen, from the figures, the Indian stakeholders (points with labels starting 'IN') are represented by points which are close to each other. It should be noted that Indian stakeholders work in the same role (product), which would be expected to improve alignment.

Australian stakeholders have the highest inertia values for both the situation today and ideal situation. They are represented by points which are generally far from each other (points with labels starting 'AU'). It is noticeable that one Australian stakeholder remains an outlier in both the situation today and ideal situation. Australian stakeholders hold three different positions and this may be the reason for their disagreement regarding their views on software quality aspects.

Finally in both the situation today and the ideal situation, the points representing the American stakeholders representing the *Product* (USPROD) perspective are close to each other. However, the two American stakeholders who hold a position of *Product Manager* (USPM) do not agree on the priorities since they are represented by points which are far apart. The only US

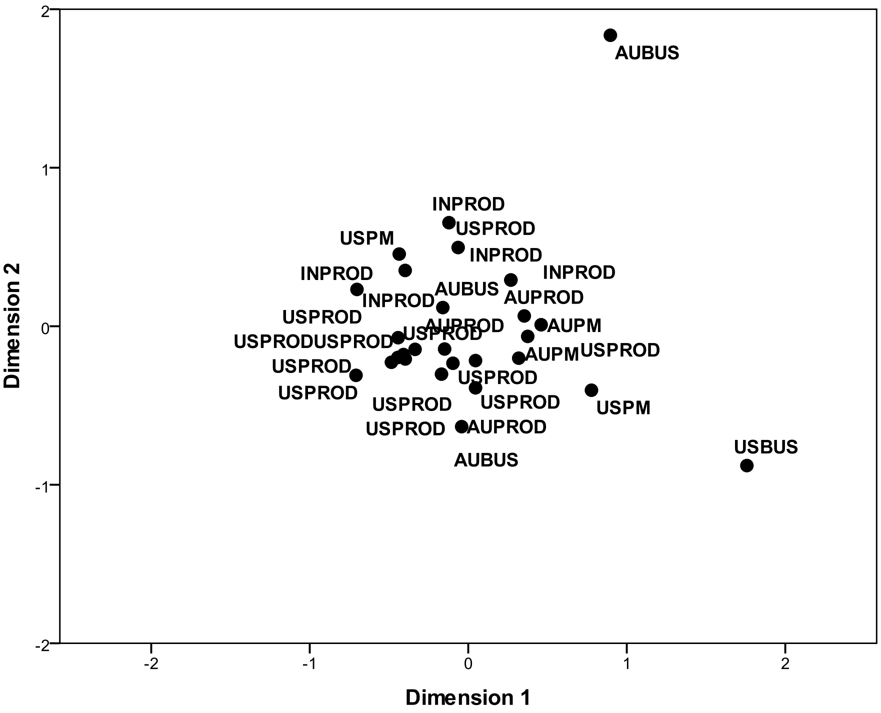

Figure 7: Case Study 3: Correspondence Analysis—Ideal Situation

stakeholder who holds a position on *Business* (US BUS) remains an outlier in both the situation today and the ideal situation.

The level of change desired by each group and overall is presented in Table 20. The results show little change is desired by any group with a significant correlation between the priorities today and in the ideal situation across all groups.

Table 20: Case Study 3: Desired Change Correlation Coefficients and p-values

Site	Correlation
Overall	0.92***
Australia	0.91***
India	0.57**
America	0.68***

6. Issues Impacting Alignment Levels

In analysing RQ3, this section examines the reasons for the level of software quality alignment between the success-critical stakeholder groups. There is a lack of studies on alignment of stakeholders with respect to software quality attributes in global settings. Thus, the main contribution is a study of the prioritisation of the non-functional requirements (software quality attributes) in companies having globally distributed development, where software products are developed jointly across different sites. This section shows that several issues are related to alignment in general, but there are also specific challenges related to aligning the prioritisation of software quality attributes.

Issues affecting alignment were identified from a number of sources. For *Case Study 1* and *Case Study 2* the questionnaire was conducted in an interview setting where issues affecting alignment were discussed. For *Case Study 3* interviews were conducted with selected stakeholders to develop a deeper understanding. Furthermore, for all three case studies workshops were held with representatives of the success-critical stakeholder groups to gain a deeper understanding of the results. The findings from the questionnaire, the interviews and the workshops are grouped under seven main groups of aspects that impact the level of alignment between the success-critical stakeholder groups: control variables, cultural factors, role based temporal perspective, quality cost-benefit, communication and coordination, inadequate quality requirement documentation and domain knowledge.

The issues identified as impacting alignment between success-critical stakeholder groups are summarised in Table 21. Each issue is discussed in the following sections.

Table 21: Reasons for reduced alignment

Reason	Case 1	Case 2	Case 3
Control variables	×	×	
Cultural factors	×	×	×
Role-based temporal perspective	×	×	
Quality cost-benefit	×	×	
Domain Knowledge			×
Inadequate documentation			×
Communication and coordination			×

6.1. Control Variables

Beck (2000) identified four control variables for software development projects in his book on Extreme Programming—*scope*, *quality*, *time* and *cost*. Beck postulated that those developing the software must be able to set at least one of these control variables to ensure they are able to deliver the other three.

The *Developers* and *Testers* in *Case Study 1* and *Case Study 2* described a situation where they are assigned functional requirements (*scope*) to develop and test by a given deadline (*time*) with a fixed-size team (*cost*). This leaves *Developers* and *Testers* with *quality* as the remaining control variable to ensure they deliver the set *scope* within *time* and *cost*.

Beck (Beck, 2000) identified *quality* to be the least optimal control variable. His research found this to be the most expensive to correct if planners were too aggressive with the *scope*, *time* and *cost*. It appears all groups within *Case Study 1* and *Case Study 2* are conscious of this situation, with all groups wanting to reduce the limitations of *time* and/or *cost* when describing their perceived *ideal* situation.

This phenomenon was not observed in *Case study 3*. This difference is likely to be related to the development methodologies employed. A waterfall-based development methodology is employed for *Case Study 3*, while the other case studies employ a Scrum-based development methodology. The extended periods between deadlines in the waterfall-based approach reduce the perceived time and cost stress, without the same constant need to deliver.

6.2. Cultural Factors

This section discusses cultural differences in terms of three cultural dimensions studied by Hofstede (1980)—power distance, masculinity and long-term orientation.

India differs greatly to America, Australia and Sweden in terms of power distance (Hofstede, 1980). Power distance defines the way people respond to perceived formal power in an organisational setting. Cultures with a low power distance are more consultative and democratic with people perceiving each other as largely equal. In cultures with a larger power distance, relationships are more autocratic, based on the relative rank in the organisational hierarchy. Of the countries studied by Hofstede, India was found to have a much higher power distance than America, Australia and Sweden. The impact of this difference was evident in all three case studies.

In *Case Study 1* and *Case Study 2*, *Developers* and *Testers* said during interviews that when overambitious work tasks were assigned to them they would deliver what they could by the deadline set, but would not challenge their managers judgement of what they could achieve. The workshop discussions showed that Swedish *Developers* and *Testers* saw the assignment of work tasks as a negotiation, in which they had a responsibility to push back if they did not think they could complete the task satisfactorily. The approach taken in Sweden lead to discussion of quality expectations that helped *Developers* and *Testers* reshape their understanding of the requirements to be aligned with their managers.

Power distance was identified as on of the greatest challenges for *Case Study 3*, with American and Australian workers being prepared to discuss, challenge and question issues with senior colleauges, while workers at the Indian site would accept and agree even if they did not understand or agree:

> In the US and Australia, a person doesn't have a problem speaking up and saying, 'hey, this doesn't work right.' Culturally in India, someone ... would never say that. They would leave that to their ... [senior manager] to bring that up. They will tell you 'yes, yes, yes, yes,' they understand when they don't.

India was found to have a shorter-term orientation to America, Australia and Sweden (Hofstede, 1980). A shorter-term orientation is associated with respect for tradition, fulfilling social obligations and protecting one's 'face.' A longer-term orientation is associated with thrift and perseverance. This emphasizes the culture seen in India of not challenging one's manager, and working hard to complete what has been requested by one's manager.

Sweden differs to America, Australia and India in terms of masculinity (Hofstede, 1980), which concerns personal attributes like assertiveness and competitiveness. Hofstede's results show India to be one of the most masculine countries, while Sweden is one of the least. It is likely that a less masculine culture finds it easier to collaborate and reach a mutual consensus on software quality—creating a cultural difference in *Case Study 1* and *Case Study 2*.

However, the cultural differences were perceived to become less problematic over the duration of *Case Study 3* as relationships developed between individuals at the three sites:

> From my perspective, regardless of culture we have established personal relationships. And it's easy to step past that. Once you establish that it makes it ... a lot easier to talk openly and gather information.

6.3. Role-based Temporal Perspective

When discussing the results with representatives of the success-critical stakeholder groups, it was asserted that some of the differences between the roles would be due to the 'time focus' of each role.

This is most evident with the *Product Support* role. Unlike all of the other roles, *Product Support* predominantly work on versions of the software that are released and in-use by customers, and are not as deeply involved in the ongoing work for future releases.

For example, the *Architects* for both products (*Case Study 1* and *Case Study 2*) identified one aspect of quality as being under-prioritized and requiring the greatest amount of additional attention. In explaining their priorities, *Product Support* acknowledged that there was internal concern from some technical groups for this aspects of quality, but noted that this aspect of quality had never been raised as an issue or potential issue by a customer. The *Architects*, however, were more concerned with the perceived future needs of customers setting their priorities. Thus the *Product Support* group was influenced by past experience, where the *Architecture* group was trying to anticipate future need.

Some representatives of the success-critical stakeholder groups also argued that the results displayed shorter-term and longer-term orientations based on the respondents role. In particular it was said that the technical groups were more focused on the current project or release, where product managers needed to manage a larger pipeline of future project and releases. The more technical groups were focused on optimising the work as part of the current activities, where product managers and to a lesser extent architects needed to also balance these needs against the ongoing future needs of the product.

6.4. Quality Cost-Benefit

The Quper model identifies the relationship between the level of quality, the benefit and the cost (Regnell et al., 2008). It shows that with increasing investment in software quality the product can transition through the following states—useless, useful, competitive and excessive. Failing to invest sufficiently in a software quality delivers a product that is not useable (or *useless*), but over-investing in software quality costs more than the benefit gained from the [*excessive*] investment.

Given that the products in *Case Study 1* and *Case Study 2* have been market leading solutions since they were first released, each fits into the *competitive* state of the Quper model. During the interviews *Strategic Product Managers* acknowledged that they were satisfied with the level of quality being delivered to customers, adding that they would like to see development times reduced for the current level of quality. These results were aligned with the experiences of the *Product Support*.

The view of the *Strategic Product Managers* and *Product Support* contrasts with that of the *Architects*, *Developers* and *Testers*, who perceive the need for additional investment in software quality. Given the *Strategic Product Managers* and *Product Support* have regular direct customer contact and the current state of the product and its ongoing success, it is likely that the level of quality that the *Developers* and *Testers* want to achieve is in the *excessive* range. This also supports the finding that *Architects* and *Developers* in *Case Study 1* want to see significant changes to their perception of the priorities *today*.

The same phenomenon was not observed in *Case Study 3*. This can be explained by the early stage of the product life-cycle, not yet being in production, and the range of roles covered not being as extensive as for the other case studies.

6.5. Communication and Coordination

Operating product development across three time zones proves problematic in *Case Study 3*, which negatively impacts the ability of the sites to reach a common understanding on quality issues. The American and Indian sites, which are not aligned in how they prioritize aspects of quality today, have no overlapping business hours. The Australian site is aligned with both the American and Indian sites, and shares approximately four business hours with the American site and three hours with the Indian site.

Multiple interviewees claimed that one of the main implications of the time zone difference is that it is very difficult to get the right people together, with sites missing out on key meetings. A senior architect described the take on the issue:

> When you are dealing across a global environment, the challenges are finding times were you can all get together. It's very difficult.

> It's extremely challenging ... And at sometime you miss out. There is only so much you can do. So that's a huge challenge. It is an issue.

One of the business analysts further emphasized that when a site misses out on a meeting they are not always updated on the discussions, severely limiting the ability of the sites to reach a mutual consensus on software quality issues:

> It is much more difficult to communicate with people across various time zones. Like for instance, you could have a quality discussion in the US and if someone does not obtain the information because they are not a participant in that meeting due to being in a different time zone, it's hard. It's not always communicated across. Sometime, the out of sight, out of context comes into play.

Communication and coordination issues are well documented in GSD contexts (Herbsleb, 2007; Mockus and Herbsleb, 2001; Herbsleb et al., 2005). This contrasts with co-located employees who share office hours, office space and can easily have planned and impromptu meetings.

Issues related to communication and coordination are greatly reduced in *Case Study 1* and *Case Study 2*. Ericsson has, in this case, tried to modularize the system to minimise coupling and cohesion between sites, and hence attempting to reduce the need for communication and coordination in the daily work between the sites. An alternative strategy is to address the coupling during development to avoid integration problems at a later stage. The strategy to choose ought to be decided from case to case depending on the risks and options involved.

6.6. Inadequate Quality Requirement Documentation

In *Case Study 3*, a number of interviewees highlighted various inadequacies in the requirements documentation, specific to quality, that they felt affected the ability of the sites to reach a mutual understanding of the quality expectations.

In the user testing phases of the product development a significant portion of the defects identified, "about 20–30%," related to "cosmetic changes." An interviewee identified a "lack of documentation" as the prime cause of these defects, however, the authors believe this problem to be made worse by the limited ability for the developers at the Indian site to access TravelBill users during the development phase.

Representatives from the business perspective recognized that they had made assumptions about the system that was to be developed that were not documented. This was also seen as negatively impacting the ability of the sites to come to a mutual agreement on what was expected in terms of quality.

> We went in with certain assumptions and we shouldn't have. So I think it should have been documented and lot of things should have been more clearly documented what they were. And it now affects our quality.

Furthermore, there were a number of changes to the TravelBill product that were not documented. As previously stated, discussions did not always involve all sites, and the missing sites were not always informed of decisions made. Coupled with a waterfall-style development methodology, this meant many problems were not picked up until the testing phase.

> We made decisions on the fly without documenting them. And that has burnt us. So we've incurred a lot more re-work because it breaks and it comes back in a bug report/issue.

A business stakeholder blamed the poor documentation on "aggressive timelines" set by high level management.

These results are aligned with other case studies in GSD, which have identified problems with a lack of formal documentation (Ågerfalk and Fitzgerald, 2006).

In *Case Study 1* and *Case Study 2*, *Developers* and *Testers* were not able to deliver the level of quality they desired given the development constraints. Thus they did not perceive a lack of understanding as a limiting factor in delivering the required level of quality.

6.7. Domain Knowledge

For *Case Study 3*, a lack of domain knowledge on the part of the developers based in India was perceived by the *business* perspective as negatively impacting quality. Despite having experience with the existing legacy system, domain knowledge was perceived as a problem. The findings showed that product stakeholder groups who are located in the US or Australia tend to be retained throughout the project, due to their domain expertise. However, product stakeholders presence in India is on a ‘need to know basis’ and as a senior project architect notes, the business processes ‘is like a black-box’ to the software engineers in India. Hence, as highlighted by a business analyst in the US, there is a lack of complete knowledge transfer to India. One of the business representatives explained how their language was interpreted differently by programmers to what they had intended:

> I would use terminology from a finance background, which is different to what a programmer would use. I think there was a big gap and problem. I think it created gaps in our system.

Lack out domain knowledge is often cited as one of the key challenges in GSD settings (Kommeren and Parviainen, 2007; Šmite and Wohlin, 2010). Thus, this finding is aligned with the related literature.

Ericsson did not face this lack of domain knowledge as they have run their Indian-based development operations for over 10 years.

7. Reflections on SAAM-SQ

The SAAM-SQ methodology is explained in Section 3. This method is used when examining RQ2 and RQ3. In analysing RQ1, this section examines the factors need to be in place to make the use of SAAM-SQ easier. This paper is the first to present the application of SAAM-SQ in a GSD environment, representing Phase 3 of this research. The method has been previously applied to settings described by onshore in-house (Phase 1) and onshore outsourcing (Phase 2). The authors’ experience applying SAAM-SQ has been ongoing, and this section presents the experiences based on the cases presented in this paper and Section 4.1.1, which are directly related to this paper. This section addresses RQ1.

At Ericsson, the work to define a quality model was a very valuable activity. As it brought all success-critical stakeholder groups to discuss quality holistically, the workshops alone helped identify areas that could benefit from improved communication and resulted in a number of groups meeting to discuss issues separately.

The same benefit was not seen in the creation of the quality model at Global Finance. As the quality model was not developed in a workshop setting, but in a series of one-on-one interviews, it did not facilitate conversations between groups that helped identify areas of potential improvement. Furthermore, the quality model developed for the Ericsson studies was better received by the questionnaire participants than the quality model developed for Global Finance. The study participants at Global Finance noted more changes that they felt were required to represent their understanding of software quality.

Participants from the studies generally did not enjoy using the HCV method, with many finding the process to make the numbers sum to 1000 frustrating. However, the lead author saw in the work with Ericsson that participants iterated through their answers repeatedly, taking care to balance the relative weight of individual aspects carefully in each iteration.

Much greater value is gained by involving all success-critical stakeholder groups in the analysis of the results. As with the workshops to create the model, involvement in this process starts conversations that allow a deeper level of analysis. This is also of greater value to the company as it helps to identify issues and differences and get the right people together to work on a solution.

Management support is critical for the successful use of SAAM-SQ. The authors have failed to collect sufficient results in settings where some senior managers have not prioritised this work, resulting in the work not being prioritised by other key stakeholders. Given the importance of understanding alignment across the whole development environment, such a result can severely limit the value of this work.

8. Discussion

This section draws upon the issues affecting alignment and misalignment presented in Section 6 to explain the levels of alignment between the sites as shown in Section 5. *Case Study 1* and *Case Study 2* are dealt with in the same section, with *Case Study 3* dealt with in the subsequent section. This section concludes with threats to the validity of this work.

8.1. Case Study 1 and Case Study 2

Case Study 1 and *Case Study 2* show similar levels of alignment between success-critical stakeholders, with the same reasons being identified as limiting higher levels of alignment. This result is not unexpected, with the environment in both case studies sharing a common development methodology, organisational structure, culture and resources coming common from a shared pool.

The results for RQ2 show low levels of alignment between identified groups in the priorities they give to the aspects that define software quality in both *Case Study 1* and *Case Study 2*.

The results for RQ3 point to a number of factors that could be leading to the levels of alignment seen. While the company has continued to create and deliver customer value with both *Product 1* and *Product 2*, not all stakeholders are aware how value is delivered to the customers in terms of the scope delivered, the quality delivered, the delivery date and the cost.

Quality requirements were often not explicitly stated as part of work assignments. This created ambiguity as to what was expected, which created an environment where *Developers* and *Testers* assumed the level of quality required. This was not always aligned with customer or management expectations.

A cultural difference was also observed between *Developers* and *Testers* in India and Sweden. When *Developers* and *Testers* in Sweden are assigned work, they see it as an opportunity to clarify their understanding and negotiate what they can achieve within given deadlines. The approach taken by employees in India respects the ability of management to know what can be achieved, and they do their best to deliver what has been requested. The approach in Sweden helps the groups reach a more common understanding, while the same groups in India are unsatisfied with what they are able to achieve in the time even though, unbeknownst to them, it meets customer needs.

A difference was also observed in the period considered by different roles when reflecting on quality requirements. For example, *Product Support* draws upon knowledge and experiences based on information about the product today and in the past. Other roles are more aware of future customer needs, and the longer-term implication of design decisions. Given a product that is successful today does not guarantee success tomorrow, it is important to draw on both perspectives.

With insufficient direction as to the quality requirements, it is understandable that technical people will aim for technically perfect solutions. However, such solutions are not feasible given the commercial realities of software development. It has long been recognized that software does not need to be perfect to be commercially viable (Yourdon, 1995), and that an over-investment in software quality requires an excessive investment without appropriate financial reward (Regnell et al., 2008).

This study further emphasized the importance of drawing upon customer experience to shape the product, but care must be taken so that this approach does not blind it to looking forward. How far a company chooses to look forward, however, must be taken on a case-by-case basis as it will change between different products.

8.2. Case Study 3

For *Case Study 3*, this section seeks to understand why:

- The American and Australian sites and the Australian and Indian sites are aligned for the situation today, but the American and Indian sites are not.
- Each site is aligned when asked to describe the ideal situation.

The Indian site is isolated from the users of TravelBill, with no representatives of this group onsite. This situation was compounded at the beginning of the product development, when there was no overlap in business hours between the two active sites—America and India. This situation can help explain the lack of alignment between the American and Indian sites in how they perceive software quality today.

It was much easier for the stakeholders at the Indian site to arrange meetings with users of the TravelBill system when Australia joined the product development. The Australian and Indian business days overlap. However, the Indian site was still limited in the level of interaction that could be achieved; for example, it was not easily possible to sit down and go through screen layouts. Thus it was easier for the Indian site to develop a more mutual understanding of quality with the Australian site than for the American site. However, this level of interaction was not sufficient to overcome all of the challenges faced.

The organizational structure may also play a role in alignment on quality. The American and Australian sites are both part of the Global Finance organization, while Indian site is that of the outsource partner IndiaSoft. Furthermore, the American site hosts the head office of Global Finance, making both the Australian and Indian sites offshore locations. Thus Australia shares common elements in terms of onshore/offshore and insourced/outsourced, while the American and Indian sites do not share any common elements.

The Indian site was also much more vulnerable to deficiencies in the requirements documentation than the American and Australian sites. The workers at the Indian site generally came with much less domain knowledge than those at the other sites, and without representatives of Global Finance's business perspective onsite it was much harder to clarify areas of ambiguity.

The Indian site was also the most hurt by the incomplete requirements documentation. With limited domain knowledge and access to business stakeholders, the Indian workers were very reliant on the information contained in these documents. This problem was compounded with changes to the system not always being documented or communicated. While the people in the American and Australian sites had knowledge, experience and expectations about the system, these were not effectively communicated and limited the ability of the Indian site to reach a common understanding of software quality with the other sites.

Over the course of the product development, through testing and bug reports, the requirements became more detailed. Thus the Indian site was able to develop a greater understanding as the project progressed. But ultimately this was too late for the project to be delivered on time and within budget.

Cultural differences also worked against the Indian site in the beginning. The Americans and Australians have a similar work culture, but this differs to aspects of the Indian work culture. However, the American and Australian sites felt that the cultural barriers decreased over the life of the product development as personal contacts were made and developed. This is another situation where it was unfortunate that the American and Indian sites do not share any part of their business day, limiting the communication, and limiting the ability of the groups to breakdown the cultural barriers.

8.3. Validity Threats

Not all stakeholders were included in these studies. For each case study the company selected a representative sample from each of the success-critical stakeholder groups. Additional stakeholders were added as it was recognised that some of the selected stakeholders may not be able to participate. The authors made contact with the stakeholders unable to participate and do not have any reason to believe that there is any systematic bias. The number of stakeholders unable to participate was small, represented a cross-section of the groups studied, and had reasons for being unable to participate. However, as no one representing the business perspective in India for *Case Study 3*, it is not possible to make any conclusion involving this group.

The study does not account for the level of influence each role has in shaping software quality. The results used for each group represent an average of the priorities of the members of the same group. While members of each group were not always aligned, the average represents a possible mutual consensus assuming all individual are equal. To help ensure reasonable results were gathered, the results were presented to key stakeholders for their input and feedback.

The quality models used in this paper were developed without the involvement of the offshore groups included in this paper, making it possible for some aspects to have been overlooked. Given that interview participants were given an opportunity for feedback and comments this risk is considered low.

The results in this paper have limited generalizability (Runeson and Höst, 2009). As shown, priorities will vary between groups, products and at different stages of a product's lifecycle. However, the SAAM-SQ method has been proven to work in different contexts and can be applied to other cases. More results will provide the possibility for more general findings.

9. Conclusion

This paper presents three case studies of offshore software development. Specifically the study aims to (1) determine the level of alignment between success-critical stakeholder groups on the priority given to aspects of software product quality, and (2) explain the reasons for differences in the priorities of the groups studied by (3) using the SAAM-SQ method to determine the level of alignment and explain the reasons for differences in priorities of the groups studied.

All three case studies discussed in this paper offer an insightful perspective on prioritization of software quality attributes in geographically dispersed teams. Although common patterns have been found across the three cases selected in this study, there was strong evidence in the results to support the uniqueness of context and its impact on the prioritization of software quality attributes as summarized in Table 21.

The main contribution of this research is the application of SAAM-SQ in a global context. The findings show that the SAAM-SQ method was successful in determining the level of alignment between teams and the reasons for the level of alignment in teams operating in a global context.

Applying the SAAM-SQ method to *Case Study 1* and *Case Study 2* shows low levels of alignment between the success-critical stakeholder groups. Both of these case studies were conducted in very similar development environments at Ericsson. The results for *Case Study 3* show Global Finance's developments sites in America, Australia and India to be mostly well aligned, with the exception of America and India.

The findings show that there was little overlap in the reasons identified as negatively impacting the level of alignment between *Case Study 3* and the other case studies. Cultural difference was the only common element, with Indian culture exhibiting a much larger power distance than the other cultures studied. In all three case studies this factor resulted in environments where Indian developers and testers would not question or challenge their managers when being assigned a task, even if they thought the task was ambiguous or not possible. This behaviour is not common in America, Australia or Sweden.

The findings also show that quality of relationship between teams may overcome problems due to cultural differences. Human and social aspects of the relationship can be more challenging than physical aspects of the relationship between the teams when it comes to cultural compatibility. The managerial implications of this mean when operating global setting managers need:

- To avoid dysfunctional relationships
- To be willing to develop social relationships as observed in *Case Study 2*
- To develop positive experience between the teams and acknowledge that negative relationships are a risk hazard for collaborative teamwork (Barney et al., 2012)

One reason for the small overlap between *Case Study 3* and the other cases regarding in the reasons found to negatively impact alignment is related to the way work is distributed between sites. In *Case Study 1* and *Case Study 2*, Ericsson has consciously worked to split development work between sites to minimize coupling and cohesion between globally distributed sites—allowing them to work as independently as possible.

The findings show that knowledge sharing between dispersed teams continues to be a challenge. Companies operating in global setting need to develop strategies to improve collaboration and knowledge sharing, for example, by minimizing coupling and cohesion among teams. In *Case Study 3*, Global Finance has high levels of coupling and cohesion between the three sites studied. The results show the reasons for reduced alignment in Global Finance to be primarily linked with inadequate knowledge sharing between the three sites. This result supports calls within the GSD literature to reduce coupling and cohesion between sites due to the issues related to knowledge sharing. The significant differences between the case studies suggest that there are no silver bullets for ensuring success-critical stakeholders are aligned when it comes to software quality.

This result emphasizes the need for any case to be analysed, so that the solution can be tailored to the specific needs of the case. To this end, SAAM-SQ provides a methodology capable of studying alignment between success-critical stakeholder groups and identifying reasons for the identified levels of alignment.

Further research is required on multiple fronts going forward. To better understand how conflicting priorities arise, work is planned to study social dynamics and the role of influencers in this space. Furthermore, organisations require help to overcome the challenges that lead them to have conflicting goals with respect to software quality.

Acknowledgements

The authors would like to thank Ericsson and Global Finance for their active involvement in and support of this research.

This work was partly funded by *Vinnova*, *Sony Mobile Communications*, *Ericsson*, *ST-Ericsson*, *Axis Communications* and *Softhouse Consulting Baltic* as part of the the Industrial Excellence Center for Embedded Applications Software Engineering (EASE).

References

P. J. Ågerfalk and B. Fitzgerald. Flexible and distributed software processes: Old petunias in new bowls? *Communications of the ACM*, 49:26–34, October 2006. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/1164394.1164416. URL `http://doi.acm.org/10.1145/1164394.1164416`.

D. Ameller, C. Ayala, J. Cabot, and X. Franch. How do software architects consider non-functional requirements: An exploratory study. In *2012 20th*

IEEE International Requirements Engineering Conference (RE), pages 41–50, 2012. doi: 10.1109/RE.2012.6345838.

G. Barczak and E. F. McDonough. Understanding the impact of contexts on factors leading to global team performance. *Management International Review*, 2003.

H. T. Barney, A. Aurum, G. C. Low, and K. Wang. Investigating post-outsourcing decision: Using the intellectual capital view. In *21st International Conference on Information Systems Development (ISD 2012)*, August 2012.

S. Barney and C. Wohlin. Software product quality: Ensuring a common goal. In Qing Wang, Vahid Garousi, Raymond Madachy, and Dietmar Pfahl, editors, *Trustworthy Software Development Processes*, volume 5543 of *Lecture Notes in Computer Science*, pages 256–267. Springer Berlin/Heidelberg, 2009. ISBN 978-3-642-01679-0. URL `http://dx.doi.org/10.1007/978-3-642-01680-6_24`.

S. Barney and C. Wohlin. Alignment of software product quality goals in two outsourcing relationships. In *14th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, April 2010. URL `http://www.bcs.org/server.php?show=conWebDoc.34786`.

S. Barney, V. Mohankumar, P. Chatzipetrou, A. Aurum, C. Wohlin, and L. Angelis. Software product quality definitions and questionnaires. 2011a. URL `http://sebseb.info/publications/ist2012b/`.

S. Barney, C. Wohlin, P. Chatzipetrou, and L. Angelis. Offshore insourcing: A case study on software quality alignment. In *6th IEEE International Conference on Global Software Engineering (ICGSE)*, pages 146–155, August 2011b. doi: 10.1109/ICGSE.2011.13.

K. Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, Reading, Massachusetts, 2000. ISBN 0-201-61641-6.

P. Berander and P. Jönsson. Hierarchical Cumulative Voting (HCV)—Prioritization of requirements in hierarchies. *International Journal of Software Engineering and Knowledge Engineering*, 16(6):819–849, December 2006. ISSN 02181940. URL `http:`

//search.ebscohost.com.miman.bib.bth.se/login.aspx?direct=true&db=buh&AN=23787584&loginpage=Login.asp&site=ehost-live.

C. Bird, N. Nagappan, P. Devanbu, H. Gall, and B. Murphy. Does distributed development affect software quality?: an empirical case study of windows vista. *Commun. ACM*, 52(8):85–93, August 2009. ISSN 0001-0782. doi: 10.1145/1536616.1536639. URL http://doi.acm.org/10.1145/1536616.1536639.

B. W. Boehm, J. R. Brown, and M. Lipow. Quantitative evaluation of software quality. In *Proceedings of the 2nd International Conference on Software Engineering*, ICSE '76, pages 592–605, Los Alamitos, CA, USA, 1976. IEEE Computer Society Press. URL http://dl.acm.org/citation.cfm?id=800253.807736.

B. W. Boehm, J. R. Brown, and H. Kaspar. *Characteristics of software quality.* TRW series of software technology; v. 1. Amsterdam: North-Holland Pub. Co.; New York: American Elsevier, 1978. ISBN 0444851054.

B. Boehm and A. Jain. An initial theory of value-based software engineering. In Stefan Biffl, Aybüke Aurum, Barry Boehm, Hakan Erdogmus, and Paul Grünbacher, editors, *Value-Based Software Engineering*, pages 15–37. Springer Berlin Heidelberg, 2006. URL http://dx.doi.org/10.1007/3-540-29263-2_2.

S. L. Brown and K. M. Eisenhardt. Product development: Past research, present findings, and future directions. *The Academy of Management Review*, 20(2):343–378, Apr 1995.

E. Carmel and P. Abbott. Why 'nearshore' means that distance matters. *Commun. ACM*, 50(10):40–46, October 2007. ISSN 0001-0782. doi: 10.1145/1290958.1290959. URL http://doi.acm.org/10.1145/1290958.1290959.

M. Cataldo and S. Nambiar. Quality in global software development projects: A closer look at the role of distribution. In *Fourth IEEE International Conference on Global Software Engineering (ICGSE)*, pages 163–172, 2009.

Y. E. Chan. Why haven't we mastered alignment? The importance of the informal organization structure. *MIS Quarterly Executive*, 1(2):97–112, June 2002.

Y. E. Chan and B. H. Reich. IT alignment: An annotated bibliography. *Journal of Information Technology*, 22(4):316–396, October 2007.

A. Colley. Police overhaul may head offshore, 2008. URL http://www.theaustralian.com.au/australian-it-old/police-overhaul-may-head-offshore/story-e6frgamf-1111115367540.

E. Conchúir, P. J. Ågerfalk, H. H. Olsson, and B. Fitzgerald. Global software development: Where are the benefits? *Communications of the ACM*, 52:127–131, August 2009. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/1536616.1536648. URL http://doi.acm.org/10.1145/1536616.1536648.

C. D. Cramton. The mutual knowledge problem and its consequences for dispersed collaboration. *Organization Science*, 12(3):346–371, 2001. doi: 10.1287/orsc.12.3.346.10098. URL http://orgsci.journal.informs.org/content/12/3/346.abstract.

J. L. de la Vara, K. Wnuk, R. Berntsson Svensson, J. Sánchez, and B. Regnell. An empirical study on the importance of quality requirements in industry. In *23rd International Conference on Software Engineering and Knowledge Engineering (SEKE 2011)*, 2011.

J. Dibbern, J. Winkler, and A. Heinzl. Explaining variations in client extra costs between software projects offshored to India. *MIS Quarterly*, 32(2):333–366, 2008.

R. G. Dromey. Concerning the chimera. *IEEE Software*, 13(1):33–43, January 1996. ISSN 0740-7459. doi: 10.1109/52.476284.

J. A. Espinosa, S. A. Slaughter, R. E. Kraut, and J. D. Herbsleb. Team knowledge and coordination in geographically distributed software development. *Journal of Management Information Systems*, 24(1):135–169, 2007.

M. J. Greenacre. *Theory and applications of correspondence analysis*. Academic Press, 1984. ISBN 9780122990502. URL http://books.google.com/books?id=LsPaAAAAMAAJ.

M. Haigh. Software quality, non-functional software requirements and it-business alignment. *Software Quality Journal*, 18(3):361–385, 2010. ISSN

0963-9314. doi: 10.1007/s11219-010-9098-3. URL http://dx.doi.org/10.1007/s11219-010-9098-3.

J. D. Herbsleb. Global software engineering: The future of socio-technical coordination. In *2007 Future of Software Engineering*, FOSE '07, pages 188–198, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2829-5. doi: http://dx.doi.org/10.1109/FOSE.2007.11. URL http://dx.doi.org/10.1109/FOSE.2007.11.

J. D. Herbsleb and A. Mockus. An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*, 29(6):481–494, June 2003. doi: 10.1109/TSE.2003.1205177.

J. D. Herbsleb and D. Moitra. Global software development. *IEEE Software*, 18(2):16–20, 2001. ISSN 0740-7459. doi: 10.1109/52.914732.

J. D. Herbsleb, D. J. Paulish, and M. Bass. Global software development at siemens: Experience from nine projects. In *Proceedings of 27th International Conference on Software Engineering (ICSE)*, pages 524–533, May 2005. doi: 10.1109/ICSE.2005.1553598.

G. Hofstede. Motivation, leadership, and organization: Do American theories apply abroad? *Organizational Dynamics*, 9(1): 42–63, 1980. ISSN 0090-2616. doi: DOI:10.1016/0090-2616(80)90013-3. URL http://www.sciencedirect.com/science/article/B6W6S-4BMTGC9-3/2/8d6de4b7d17bdbea968a6efed97e2615.

ISO9126. Software engineering – product quality – part 1: Quality model. International Standards Organization, 2001. URL http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=22749&scopelist=&showrevision=y.

E. Johansson, A. Wesslén, L. Bratthall, and M. Höst. The importance of quality requirements in software platform development-a survey. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS)*, January 2001. doi: http://doi.ieeecomputersociety.org/10.1109/HICSS.2001.927252.

B. Kitchenham and S. L. Pfleeger. Software quality: The elusive target. *IEEE Software*, 13(1):12–21, January 1996. ISSN 0740-7459. doi: 10.1109/52.476281.

R. Kommeren and P. Parviainen. Philips experiences in global distributed software development. *Empirical Software Engineering*, 12:647–660, 2007. ISSN 1382-3256. doi: 10.1007/s10664-007-9047-3. URL `http://dx.doi.org/10.1007/s10664-007-9047-3`.

T. P. Liang, J. Jiang, G. S. Klein, and J. Y.C. Liu. Software quality as influenced by informational diversity, task conflict, and learning in project teams. *IEEE Transactions on Engineering Management*, 57(3):477–487, August 2010. doi: 10.1109/TEM.2009.2033049.

C. H Loch and S. Kavadias, editors. *Handbook of New Product Development Management*. Butterworth-Heinemann, 2008.

E. MacGregor, Y. Hsieh, and P. Kruchten. The impact of intercultural factors on global software development. In *Canadian Conference on Electrical and Computer Engineering*, 2005. doi: 10.1109/CCECE.2005.1557127.

J. A. McCall, P. K. Richards, and G. F. Waiters. Factors in software quality volume 3, prelimlinary handbook on software quality for an aquisition manager. Technical report, General Electric Company, 1977.

J. H McDonald. *Handbook of Biological Statistics*. Sparky House Publishing, Baltimore, Maryland, 2 edition, 2009.

A. Mockus and J. Herbsleb. Challenges of global software development. In *Proceedings of the Seventh International Software Metrics Symposium (METRICS)*, pages 182–184, 2001. doi: 10.1109/METRIC.2001.915526.

L. Morgan. Consider the outsource. *Software Development Times*, (100): 28–30, April 2004.

B. Nicholson and S. Sahay. Some political and cultural issues in the globalisation of software development: case experience from britain and india. *Information and Organization*, 11(1): 25–43, 2001. ISSN 1471-7727. doi: DOI:10.1016/S0959-8022(00)00008-4. URL `http://www.sciencedirect.com/science/article/B6W7M-422G6PW-3/2/9e9e0d29d8a4f3b0d3bafb7bb78d1648`.

J. Noll, S. Beecham, and I. Richardson. Global software development and collaboration: Barriers and solutions. *ACM Inroads*, 1:66–78, August 2010. ISSN 2153-2184. doi: http://doi.acm.org/10.1145/1835428.1835445. URL http://doi.acm.org/10.1145/1835428.1835445.

M. Phongpaibul and B. Boehm. Improving quality through software process improvement in thailand: initial analysis. In *3-WoSQ: Proceedings of the third workshop on Software quality*, pages 1–6, New York, NY, USA, 2005. ACM. ISBN 1-59593-122-8. doi: http://doi.acm.org/10.1145/1083292.1083299.

B. Regnell, R. Berntsson Svensson, and T. Olsson. Supporting roadmapping of quality requirements. *IEEE Software*, 25(2):42–47, March 2008. ISSN 0740-7459. doi: 10.1109/MS.2008.48.

P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, April 2009. URL http://dx.doi.org/10.1007/s10664-008-9102-8.

S. Sahay, B. Nicholson, and S. Krishna. *Global IT outsourcing: Software development across borders*. MIT Press, 2003. ISBN 0521816041.

D. Šmite and C. Wohlin. Software product transfers: Lessons learned from a case study. In *5th IEEE International Conference on Global Software Engineering (ICGSE)*, pages 97–105, 2010. doi: 10.1109/ICGSE.2010.19.

D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt. Empirical evidence in global software engineering: A systematic review. *Empirical Software Engineering*, 15(1):91–118, February 2010. URL http://dx.doi.org/10.1007/s10664-009-9123-y.

L. Wang, K. L. Gwebu, J. Wang, and D. X. Zhu. The aftermath of information technology outsourcing: An empirical study of firm performance following outsourcing decisions. *Journal of Information Systems*, 22(1): 125–159, 2008. doi: 10.2308/jis.2008.22.1.125. URL http://link.aip.org/link/?JIS/22/125/1.

E. Yourdon. When good enough software is best. *IEEE Software*, 12(3): 79–81, May 1995. ISSN 0740-7459. doi: 10.1109/52.382191.