Offshore Insourcing: A Case Study on Software Quality Alignment

Sebastian Barney
Blekinge Institute of Technology, Sweden
sebastian.barney@bth.se

Claes Wohlin
Blekinge Institute of Technology, Sweden
claes.wohlin@bth.se

Panagiota Chatzipetrou
Aristotle University of Thessaloniki, Greece
pchatzip@csd.auth.gr

Lefteris Angelis
Aristotle University of Thessaloniki, Greece
lef@csd.auth.gr

Abstract—Background: 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.

Aim: This work seeks to understand the levels of alignment between key stakeholders on aspects of software quality for two products developed as part of an offshore insourcing arrangement. The study further aims to explain the levels of alignment identified.

Method: Representatives of key stakeholder groups for both products ranked aspects of software quality. The results were discussed with the groups to gain a deeper understanding.

Results: Low levels of alignment were found between the groups studied. This is associated with insufficiently defined quality requirements, a culture that does not question management and conflicting temporal reflections on the product's quality.

Conclusion: The work emphasizes the need for greater support to align success-critical stakeholder groups in their understanding of quality when offshoring software development.

Keywords-software, quality, global software development

I. INTRODUCTION

The rapid rise of global software development (GSD) [1] 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 [2], but there are also many challenges. The most commonly cited challenges in GSD contexts concern communication [3] and coordination [4], both essential elements in creating alignment between stakeholders [5].

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 [6], [7]. Alignment of stakeholders allows them to collaborate more effectively and produce results that support the long-term business strategies, while highly misaligned teams can cause conflict and eventually lead to the failure of a project.

This paper aims to expand the author's previous work into the alignment of success-critical stakeholder groups on software quality [8], [9] by looking beyond the onshore groups previously studied to include offshore software development. A common understanding of software quality has been found to be the most effective way for groups to achieve a common goal in a GSD setting [10]. Thus this study explores the offshore insourcing setting within GSD with the aim to understand (1) the levels of alignment between sites in terms of software quality, and (2) the reasons for the level of alignment.

The remainder of this paper is structured as follows. Key literature introducing the topic is presented in Section II. The research questions and methodology are presented in Section III. Information about the case study is presented in Section IV. Results are presented in Section V and discussed in Section VI. Finally conclusions are made in Section VII.

II. BACKGROUND

This section introduces key concepts and related work.

A. Software Quality

There are many definitions of software quality [11], with the most common in software engineering being 'conformance to specification' and 'fit for purpose.' The growing body of value-based software engineering (VBSE) literature recognizes that perceptions of software quality are individual and shaped by experiences [12]. 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 [13].

Software quality does not need to be perfect [14]. 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 [15]. 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.

B. 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 including McCalls quality model, Boehms quality model and ISO 9126 [11]. All of these models are criticized for various deficiencies. The major complaints about ISO 9126 are missing or insufficiently detailed aspects of quality or insufficient information on measuring the aspect defined.

An alternative approach to defining quality models was undertaken in Dromey's quality model [16]. This work proposing defining 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.

C. Key Stakeholder Alignment

Stakeholder alignment has been defined as "convergent intentions, shared understanding and coordinated procedures" [6], inline with goals of VBSE. There is overwhelming evidence showing aligned groups outperform those who are not aligned [7]. Alignment allow stakeholder to collaborate more effectively and produce systems that support the long-term business strategies. Highly misaligned teams can cause conflict and eventually lead to the failure of a project.

Stakeholder alignment requires ongoing effort to ensure the group remain aligned [6]. Communication, coordination, control, supervision, creating social bonds and building trust are key to creating alignment between stakeholders [5]. A lack of awareness and belief in alignment have been identified as main contributors to misalignment [7].

While GSD creates many opportunities, it also poses challenges that can negatively impact alignment if not properly addressed.

D. Global Software Development

GSD is defined as software development work undertaken across national boundaries at geographically separated locations. While teams are not co-located, they are still working towards a common goal with a commercially viable product. Groups are 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 [17], with a number of benefits and risks. It allows companies to call upon a global talent pool to supplement a locally scarce resource pool [18]. These specialized skills can have a positive impact on productivity and quality [3], [17]. Further, many studies cite the ability to reduce development costs [17], and to focus on strategic business functions with day-to-day operations off-loaded [18].

The most commonly cited challenges of GSD relate to communication [3] and coordination [4]. 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 [19]. While new technologies have helped to reduce the barriers created by distance, it is emphasized that distance still matters [5].

Intercultural factors, such as power distance, individualism, and uncertainty avoidance [20] can also be problematic in GSD settings. Studies on the impact of national and cross-cultural issues on systems development emphasize the need to take organizational and national culture seriously when working in these environments [20]. Further, cultural issues can exacerbate exiting communication and coordination problems [21].

III. METHODOLOGY

This paper aims to determine the level of alignment between key stakeholder groups involved in offshore software product development on the priorities they give to aspects of software product quality. It then seeks to understand the reasons behind the level of alignment between the groups.

A. Research Questions

This paper addresses two research questions:

- **RQ 1.** To what degree are success-critical stakeholder groups aligned in offshore software product development on the priorities given to aspects of software product quality:
  - **RQ 1.1.** In the perception of the situation *today*?
  - **RQ 1.2.** In the perception of the *ideal* situation? That is what should the priorities be today.
- **RQ 2.** What are the reasons for the level of alignment seen between these groups?

B. Method

This paper employs a method previously developed and used by the authors to determine the level of alignment between key stakeholder groups [8], [9]. Given the success of this method to meet the aims of the study it is reused.

The method draws heavily on Theory-W [13], which aims to create win-win scenarios by getting success-critical stakeholder groups to agree on how product development should proceed. This 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 method used to answer the research question are described below.

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.

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 ensures critical perspectives are not lost, while less important perspectives cannot dominate.

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 different 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 II-B is a good starting point, but company and product specific needs must be taken into account [8].

4) Develop a questionnaire: This method proposes the use of the hierarchical cumulative voting technique (HCV) [22] to elicit the priorities given the various aspects 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 [8]; and HCV provides a method for breaking the problem into a series of smaller direct comparison exercise, with a method to join these results back together.

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 participants have a common understandings of the questionnaire with the interviewer able to assist with questions or problems faced by the respondents.

6) Analyse the results: It is possible to transform the results of the HCV exercise into cumulative voting (CV) results [22]. 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 [23], determining the level of alignment between the success-critical stakeholder groups.

The method is extended in this paper to further examine the alignment between the members of each success-critical stakeholder group. As this has not previously been done, the process is described in Section III-C.

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?

C. Determining alignment within a group

Inertia measures the variation in a dataset [24]. 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 means 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-sqaure 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 the cumulative voting method gives 1000 points to assign across all the aspects, it is possible to simply this equation to:

$$inertia = \frac{\chi^2}{1000 \times S} \quad (4)$$

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.

IV. CASE STUDY

This paper presents a case study of two major products at Ericsson. Ericsson is a world leading company in telecommunications, providing 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.

This paper employs an exploratory case study on two products at Ericsson to gain insight and understanding from the current situation [25]. The key decisions made in the implementation of the methodology in the specific context are described in the following sections.

A. 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 "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.

Global software development practices are employed with both products. Work is distributed 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 maximize the use of available resources.

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 [8]. *Phase 2* expanded this work to include subcontracted developers based out of the development centre in Sweden [9]. The third phase, presented in this paper, examines the alignment between success-critical stakeholder groups employed by Ericsson on *Product 1* and based in India. 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 of this work.
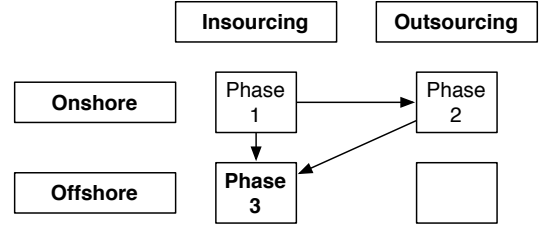

Figure 1. Case studies relating to Product 1

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

B. Success-critical Stakeholder Groups

The first author worked with three senior managers to identify success-critical stakeholder groups. 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 I for both products. For example, two *Tactical Product Managers* were identified for each product, but only one was interviewed for *Product 1*. The first author's limited time in India meant 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.

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

Table II
SPEARMAN RANK CORRELATION COEFFICIENTS DESCRIBING PRODUCT 1 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

Table III
P-VALUES FOR SPEARMAN RANK CORRELATION COEFFICIENTS DESCRIBING PRODUCT 1 TODAY

	Arch	Dev	PS	PM	SPM	TPM	Test
Arch	0.000	0.393	0.093	0.464	0.057	0.876	0.483
Dev		0.000	**0.042**	0.150	0.282	0.264	**0.039**
PS			0.000	**<0.001**	**0.009**	0.681	**0.008**
PM				0.000	0.221	0.570	0.281
SPM					0.000	0.221	0.095
TPM						0.000	0.154
Test							0.000

C. Quality Model and Questionnaire

Given this work expands two previously conducted case studies of *Product 1* [8], [9] 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*. The terms and definitions defining the model for the case study presented in this paper are available online [26].

D. Conduct the questionnaire

Interviews were conducted by the first author 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 *Stategic Product Managers* who selected to have a phone interview with screen sharing due to conflicting commitments. Interviews took on average 45 minutes.

V. RESULTS

This section presents the results for *Product 1* and *Product 2* individually.

A. Product 1

First, RQ1.1 aims to determine the degree to which the offshore success-critical stakeholder groups are aligned in how they perceive the priorities on aspects of software product quality today. The results for this question are presented in Table II and Table III. Table II contains Spearman rank correlation coefficients for each pair of groups. In this table results can range from one (1) to minus 1 (−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. Table III contains p-values for the Spearman rank correlation coefficients. Values indicating the presence of a significant relationship between the priorities of the two groups at the $p < 0.05$ level are marked in bold.

The results for *Product 1* show lower levels of alignment than seen in the previous case studies of this product when describing the situation *today*. 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*. 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. Levels of alignment between the *Tactical Product Managers* and other roles is low.

The results answering RQ1.2, describing the alignment between success-critical stakeholders in the perceived *ideal* situation, for *Product 1* are presented in Table IV and Table V. 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. Further 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.

To determine the degree to which members of a group were aligned, the inertia 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. Table VI provides the inertia and the significance of the chi-square test (p-

Table IV
SPEARMAN RANK CORRELATION COEFFICIENTS DESCRIBING PRODUCT 1 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

Table V
P-VALUES FOR SPEARMAN RANK CORRELATION COEFFICIENTS DESCRIBING PRODUCT 1 IDEAL SITUATION

	Arch	Dev	PS	PM	SPM	TPM	Test
Arch	0.000	0.194	0.151	0.921	0.787	0.369	0.518
Dev		0.000	0.634	0.329	0.057	0.434	0.082
PS			0.000	0.194	0.726	0.173	0.052
PM				0.000	0.999	0.433	0.289
SPM					0.000	0.699	0.371
TPM						0.000	0.071
Test							0.000

value) for *Product 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 alignment between the stakeholders in each group. The inertia could not be 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 the members of each group do not agree on the priorities on quality for the product.

Table VI
INERTIA WITHIN EACH GROUP FOR PRODUCT 1

	Today		Ideal 2	
	Inertia	P-value	Inertia	P-value
Overall	0.28	<**0.001**	0.22	<**0.001**
Arch	0.26	<**0.001**	0.19	<**0.001**
Dev	0.21	<**0.001**	0.15	<**0.001**
PS	0.36	<**0.001**	0.39	<**0.001**
PM	0.18	<**0.001**	0.21	<**0.001**
SPM	0.09	<**0.001**	0.05	<**0.001**
TPM	—	—	—	—
Test	0.15	<**0.001**	0.08	<**0.001**

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 VII, 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, with those at the $p < 0.05$ significance level marked in bold.

Table VII
DESIRED CHANGE CORRELATION COEFFICIENTS AND P-VALUES FOR PRODUCT 1

	Correlation	P-value
Overall	0.43	**0.035**
Arch	−0.18	0.390
Dev	0.06	0.784
PS	0.41	**0.048**
PM	0.77	<**0.001**
SPM	0.56	**0.005**
TPM	0.51	**0.011**
Test	0.26	0.220

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*, *Strategic Product Managers* and *Tactical Product Managers*. The groups desiring the greatest level of change in priorities are the *Architects* and *Developers*. Reasons for this will be discussed in Section VI.

B. Product 2

The results answering RQ1.1 for *Product 2*, determining the level of alignment between success-critical stakeholder groups in the perceived situation *today*, are presented in Table VIII and Table IX. 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 insulated from some aspects of the product. A significant relationship is also seen between the priorities of the *Developers* and *Project Managers*.

Table VIII
SPEARMAN RANK CORRELATION COEFFICIENTS DESCRIBING PRODUCT 2 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 results for RQ1.2 for *Product 2*, determining the level of alignment between success-critical stakeholder groups in the perceived *ideal* situation, are presented in Table X and Table XI. A significant relationship between the priorities was found at the $p < 0.05$ level between *Testers* and

	Arch	Dev	PS	PM	SPM	TPM	Test
Arch	0.000	0.152	0.954	0.132	0.295	0.059	**0.021**
Dev		0.000	0.882	**0.040**	0.058	0.164	**0.029**
PS			0.000	0.725	0.275	0.498	0.301
PM				0.000	0.081	0.671	0.127
SPM					0.000	0.280	**0.001**
TPM						0.000	**0.017**
Test							0.000

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*.

	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

	Arch	Dev	PS	PM	SPM	TPM	Test
Arch	0.000	**<0.001**	0.609	0.155	0.255	0.068	0.076
Dev		0.000	0.741	0.051	0.812	0.283	**0.016**
PS			0.000	0.104	**0.009**	0.243	**0.013**
PM				0.000	**0.039**	0.089	0.207
SPM					0.000	0.166	**0.015**
TPM						0.000	**0.003**
Test							0.000

Table XII 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. With one exception, this suggests that the members of each groups do not agree on the priorities on quality for the product. The inertia could not be calculated for *Product Support*, as only one person was in this group and thus, by definition, was in complete alignment.

Finally, Table XIII 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.

	Today		Ideal	
	Inertia	P-value	Inertia	P-value
Overall	0.26	**<0.001**	0.21	**<0.001**
Arch	0.36	**<0.001**	0.29	**<0.001**
Dev	0.19	**<0.001**	0.14	**<0.001**
PS	—	—	—	—
PM	0.08	**<0.001**	0.02	0.107
SPM	0.19	**<0.001**	0.12	**<0.001**
TPM	0.08	**<0.001**	0.06	**<0.001**
Test	0.16	**<0.001**	0.19	**<0.001**

	Correlation	P-value
Overall	0.65	**0.001**
Arch	0.69	**<0.001**
Dev	0.69	**<0.001**
PS	0.76	**<0.001**
PM	0.60	**0.002**
SPM	0.65	**0.001**
TPM	0.41	**0.050**
Test	0.73	**<0.001**

VI. DISCUSSION

The results show that each of the success-critical stakeholder groups for *Product 1* and *Product 2* are in some degree of conflict as to what the priorities are on quality both *today* and in their perceived *ideal* situation. There are still results that stand out from each group, however, such as a desire to reduce the *time* and/or *cost* limitations on development activities.

When describing the situation *today*, the results for *Product 1* show some alignment between some of the success-critical stakeholder groups. *Product Support* stands out as the group that is aligned with the priorities of most groups. However, each group has different ideas of quality when it comes to describing their perceived *ideal* situation. This divergence in the *ideal* situation has been seen in the previous studies, where each group prioritizes issues that more directly affect them and pay less attention to the issues that affect other groups [8], [9]. While all groups want to see some change, only the *Architects* and *Developers* want to see significant changes made to the priorities *today*.

The results for *Product 2* show some alignment between the success-critical stakeholder groups when describing both the situation *today* and the *ideal* situation. For this product it is the *Testers* that stand out as aligned with the priorities of the other groups. While all groups want to see some change, the level of change desired was not significant for any group—indicating all groups are reasonably content with the situation *today*.

During the interviews representatives of all groups expressed that *time*-based deadlines and *cost* restrictions were the major factor limiting the ability to achieve higher levels

of quality for *Product 1* and *Product 2*. This was most vocally expressed by the *Developers* and *Testers*, who felt less constraints were needed in product development to allow a higher level of quality to be achieved.

This study only examined the alignment between success-critical stakeholder groups, and did not explicitly examine these groups alignment to project, product or organisation strategies, as this is beyond the scope of the method used.

To answer RQ2, the following sections explore the reasons behind the alignment and misalignment, along with the tension between scope/quality and time/cost.

A. Control Variables

First, this section will examine the case study in terms of the control variables used in the agile development environment. Beck [27] identified four control variables—*scope*, *quality*, *time* and *cost*—and notes 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* for *Product 1* and *Product 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 control variable to ensure they deliver the *scope* within *time* and *cost*.

Beck [27] 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 are conscious of this situation for both products, with all groups wanting to reduce the limitations of *time* and/or *cost* for both *Product 1* and *Product 2* when describing their perceived *ideal* situation.

B. Cultural Factors

While time and cost where identified as limiting factors in the first case study [8], the constraints this had on quality were not as evident as in the case study presented in this paper. One possible reason for this difference relates to differing cultural factors at play in Sweden and India.

India and Sweden are very different when it comes to power distance [20], which refers to the degree to which "the less powerful members of organizations ... accept and expect that power is distributed unevenly." Of the countries studied by Hofstede, India was found to have the greatest power distance, while Sweden has a small power distance. This was evident in the interviews with Indian *Developers* and *Testers*, with many saying 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.

Another difference between India and Sweden refers to the masculinity [20], 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.

India was found to have a shorter-term orientation than Sweden [20]. 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.

C. Role-based Temporal Perspective

Shorter-term and longer-term orientations were also observed on a role-based basis. For example, the *Architects* for both products identified one aspect of quality identified as being under-prioritized and requiring the greatest amount of additional attention. *Product Support* acknowledged 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.

In the aforementioned scenario the *Architects* are concerned about the possibility of products failing to meet customers' future quality requirements. *Product Support* draws from their experience customers past experiences.

While *Product Support* are aware of customers immediate needs and concerns, it appears other roles have insight into a longer-term perspective of the products sustainability. This shows a potential incongruence between the cultural aspects and the roles, where roles may have dominance.

D. Quality Cost-Benefit

Given that both *Product 1* and *Product 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 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 inline 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* of *Product 1* want to see significant significant changes to their perception of the priorities *today*.

E. Validity Threats

Not all stakeholders were included in this study and it 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 model used in this paper was developed without the involvement of the offshore groups included in this paper, making it possible for some aspects to have been overlooked. Given this model was developed for the same product for a similar group of success-critical stakeholder groups, and interviews participants were given an opportunity for feedback and comments this risk is considered low.

The results in this paper have limited generalizability [25]. Priorities will vary between groups , products and at different stages of a product's lifecycle. However, the 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.

VII. Conclusion

This paper presents a case study of the offshore development effort of two products. For these products it seeks to determine the alignment between success-critical stakeholders groups on the priority given to aspects of software product quality. It then seeks to understand the reasons for the level of alignment between the groups.

The results for RQ1 show low levels of alignment between identified groups in the priorities they give to the aspects that define software quality for both *Product 1* and *Product 2*. Given the low levels of alignment, it was deemed not appropriate to make comparisons between the sites as there is no representative viewpoint from which to compare.

The results for RQ2 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 through *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. The main reasons behind the level of alignment are listed below.

- 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 inline 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 a 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 succeeding today is not guaranteed 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 [14], and that an over-investment in software quality requires an excessive investment without appropriate financial reward [15].

The results indicate an opportunity for improvement. Ensuring the success-critical stakeholder groups have a common understanding of software quality will help them work together more effectively towards that goal [10], [12]. This would improve the job satisfaction, as people would be more reconciled with the work that they deliver, especially knowing it met customers requirements.

This study further emphasized the importance of drawing upon customer experience to shape the product, but care must be taken to 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.

While these results are not generalizable, the solutions provides a tested methodology from which further academic and industrial work can be done. This paper extends the previous work in this area, most notably by looking into offshore insourcing, and by providing a method to calculate the level of alignment within a group—not just between groups, as has been done before. This also shows the methodology to work in range of differing contexts

Going forward research needs to focus on methods to help

groups reach a mutual understanding of software quality. This research highlights some of the challenges that are faced in different cultural environments, and in bringing different roles together.

REFERENCES

[1] J. Dibbern, J. Winkler, and A. Heinzl, "Explaining variations in client extra costs between software projects offshored to India," *MIS quarterly*, vol. 32, no. 2, pp. 333–366, 2008.

[2] D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt, "Empirical evidence in global software engineering: a systematic review," *Empirical Software Engineering*, vol. 15, no. 1, pp. 91–118, February 2010.

[3] A. Mockus and J. Herbsleb, "Challenges of global software development," in *Proceedings of the Seventh International Software Metrics Symposium (METRICS)*, 2001, pp. 182–184.

[4] J. D. Herbsleb, "Global software engineering: The future of socio-technical coordination," in *2007 Future of Software Engineering*, 2007, pp. 188–198.

[5] E. Carmel and P. Abbott, "Configurations of global software development: offshore versus nearshore," in *Proceedings of the 2006 international workshop on Global software development for the practitioner*, 2006, pp. 3–7.

[6] Y. E. Chan, "Why haven't we mastered alignment? the importance of the informal organization structure," *MIS Quarterly Executive*, vol. 1, no. 2, pp. 97–112, June 2002.

[7] Y. E. Chan and B. H. Reich, "IT alignment: An annotated bibliography," *Journal of Information Technology*, vol. 22, no. 4, pp. 316–396, October 2007.

[8] S. Barney and C. Wohlin, "Software product quality: Ensuring a common goal," in *Proceedings of the International Conference on Software Process (ICSP)*, Q. Wang, R. Madachy, and D. Pfahl, Eds., May 2009, pp. 256–267.

[9] 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.

[10] 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*, 2005, pp. 1–6.

[11] B. Kitchenham and S. L. Pfleeger, "Software quality: The elusive target," *IEEE Software*, vol. 13, no. 1, pp. 12–21, January 1996.

[12] 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.

[13] B. Boehm and A. Jain, "An initial theory of value-based software engineering," *Value-Based Software Engineering*, pp. 15–37, 2006.

[14] E. Yourdon, "When good enough software is best," *IEEE Software*, vol. 12, no. 3, pp. 79–81, May 1995.

[15] B. Regnell, R. Berntsson Svensson, and T. Olsson, "Supporting roadmapping of quality requirements," *IEEE Software*, vol. 25, no. 2, pp. 42–47, March–April 2008.

[16] R. G. Dromey, "Concerning the chimera," *IEEE Software*, vol. 13, no. 1, pp. 33–43, January 1996.

[17] E. Ó. Conchúir, P. J. Ågerfalk, H. H. Olsson, and B. Fitzgerald, "Global software development: Where are the benefits?" *Communications of the ACM*, vol. 52, pp. 127–131, August 2009.

[18] 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*, vol. 22, no. 1, pp. 125–159, 2008.

[19] J. Noll, S. Beecham, and I. Richardson, "Global software development and collaboration: barriers and solutions," *ACM Inroads*, vol. 1, pp. 66–78, August 2010.

[20] G. Hofstede, "Motivation, leadership, and organization: Do american theories apply abroad?" *Organizational Dynamics*, vol. 9, no. 1, pp. 42–63, 1980.

[21] J. D. Herbsleb and D. Moitra, "Global software development," *IEEE Software*, vol. 18, no. 2, pp. 16–20, 2001.

[22] P. Berander and P. Jönsson, "Hierarchical Cumulative Voting (HCV) — Prioritization of requirements in hierarchies," *International Journal of Software Engineering and Knowledge Engineering*, vol. 16, no. 6, pp. 819–849, December 2006.

[23] J. H. McDonald, *Handbook of Biological Statistics*, 2nd ed. Baltimore, Maryland: Sparky House Publishing, 2009.

[24] M. J. Greenacre, *Theory and applications of correspondence analysis*. Academic Press, 1984.

[25] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, April 2009.

[26] S. Barney and C. Wohlin, "Software product quality questionnaire," 2008, http://www.bth.se/tek/aps/sba.

[27] K. Beck, *Extreme Programming Explained: Embrace Change*. Reading, Massachusetts: Addison-Wesley, 2000.