Effort Estimation in Large-Scale Software Development: An Industrial Case Study

Muhammad Usman, Ricardo Britto, Lars-Ola Damm, Jürgen Börstler

Abstract

Context: Software projects frequently incur schedule and budget overruns. Planning and estimation are particularly challenging in large and globally distributed projects. While software engineering researchers have been investigating effort estimation for many years to help practitioners to improve their estimation processes, there is little research about effort estimation in large-scale distributed agile projects.

Objective: The main objective of this paper is three-fold: i) to identify how effort estimation is carried out in large-scale distributed agile projects; ii) to analyze the accuracy of the effort estimation processes in large-scale distributed agile projects; and iii) to identify the factors that impact the accuracy of effort estimates in large-scale distributed agile projects.

Method: We performed an exploratory longitudinal case study. The data collection was operationalized through archival research and semi-structured interviews.

Results: The main findings of this study are: 1) underestimation is the dominant trend in the studied case, 2) re-estimation at the analysis stage improves the accuracy of the effort estimates, 3) requirements with large size/scope incur larger effort overruns, 4) immature teams incur larger effort overruns, 5) requirements developed in multi-site settings incur larger effort overruns as compared to requirements developed in a collocated setting, and 6) requirements priorities impact the accuracy of the effort estimates.

Conclusion: Effort estimation is carried out at quotation and analysis stages in the studied case. It is a challenging task involving coordination amongst many different stakeholders. Furthermore, lack of details and changes in requirements, immaturity of the newly on-boarded teams and the challenges associated with the large-scale add complexities in the effort estimation process.

Keywords: Effort estimation, large-scale software development, global and agile software development

1. Introduction

In today’s globalized market, software is increasingly developed in globally distributed projects [1, 2, 3, 4]. However, geographical, temporal, and cultural distances make coordination and communication more challenging in such projects, which may lead to more software defects [5] and schedule and budget overruns [6, 7].

The literature indicates that practitioners have fallen short of providing accurate and reliable effort estimates in both collocated and distributed projects. A secondary study by Moløkken and Jørgensen shows that 60–80% of the projects they surveyed had schedule and/or budget overruns [7]. More accurate effort estimates could help project managers to plan software projects in a better way, increasing the chances that projects do not get delayed or exceed budgets. More accurate effort estimates are not only associated with better effort estimation methods; to account for adequate predictors (i.e. size metrics and cost drivers) is equally important [8]. Identifying adequate predictors requires empirical research.

*Corresponding author

Email addresses: muhammad.usman@bth.se (Muhammad Usman), ricardo.britto@bth.se (Ricardo Britto), lars-ola.damm@ericsson.com (Lars-Ola Damm), jurgen.borstler@bth.se (Jürgen Börstler)
While a lot of research has covered effort estimation in software development projects in general [9, 10, 11], very little research has been conducted about effort estimation in large-scale distributed projects [12].

To address this gap, we conducted a longitudinal exploratory industrial case study, to investigate how effort estimation is performed in large-scale distributed agile projects and which factors (cost drivers) impact the accuracy of the effort estimates the most. Identifying these factors and their relationships with effort estimates would benefit managers at the case company, and other similar cases, to improve their estimation and planning processes. For the context of this work, we adopt Dikert et al.’s definition of large-scale projects, i.e. projects that involve at least 50 people or more than 10 teams [12]. The case is a large-scale distributed agile software project with a large amount of complex legacy code at Ericsson AB, a Swedish company that develops telecommunication-related products.

In this case study, we address the following research questions. The context for all research questions is large-scale distributed agile projects comprising a large amount of complex legacy code.

- RQ1: How is effort estimation carried out?
- RQ2: How accurate are the effort estimates?
- RQ3: Which factors impact the accuracy of the effort estimates?
- RQ4: How do the identified factors impact the accuracy of the effort estimates?

The main contribution of this paper are described in the following:

- Understand and describe how effort estimation processes are managed in a large scale distributed agile project.
- Analysis of the effort estimates’ accuracy, and how re-estimation impacts estimates in a large scale distributed agile project.
- A better understanding of how certain factors affect effort estimates’ accuracy in a large-scale distributed agile project.

The remainder of the paper is organized as follows. Related work is presented and discussed in Section 2. Section 3 describes the research methodology employed in this case study. The results of the case study are presented in Section 4 and discussed in Section 5. Threats to validity are discussed in Section 6. Finally, Section 7 presents our conclusions and view on future work.

2. Related work

In this section, we discuss some primary and secondary studies about effort estimation in the context of global software development.

A review by Jørgensen [13] covering empirical research published 1990–2002, showed that expert estimation is the most commonly used estimation strategy in software projects. Jørgensen and Shepperd [11], furthermore noted that the proportion of studies on expert judgment based estimation is increasing, and that the available evidence does not support that the use of formal estimation models improves the estimation accuracy. More recent reviews [9, 10] on effort estimation in agile and global software development contexts also showed that the most frequently applied estimation methods are the ones that rely on subjective assessment by experts, such as planning poker, analogy, and use case points.

Britto et al.’s systematic literature review [10] identified a wide range of factors contributing to the effort of globally distributed projects, e.g., socio-cultural, geographical and temporal distances. However, none of the primary studies provided detailed information about the effort estimation processes. A follow-up study by El Bajta et al. [14] corroborated these results. Usman et al. [9] identified team skills and prior experience as the most frequently used cost drivers in studies on effort estimation in agile projects.
A survey among practitioners by Britto et al. [15] indicated that the effort estimation processes in collocated and globally distributed agile projects are very similar, although the cost drivers depend on the development operational model (collocated or distributed). The main factors affecting effort estimation accuracy are requirements-related issues and communication overhead that is not properly accounted for.

In a similar survey, Usman et al. [16] found that similar effort estimation techniques are used in agile collocated and distributed contexts (mainly planning poker). There are differences in cost drivers though; geographic, socio-cultural and temporal distances were found to require more communication and coordination efforts in distributed projects than in collocated projects. In both contexts, requirements and project management-related issues impact negatively the accuracy of the effort estimates.

Lenarduzzi et al. [17] replicated a case study on functional size measures and effort estimation in a small agile project. Their results show that the effort estimated by developers was more accurate than the estimates obtained through functional size measures.

Evbota et al. [18] investigated the challenges of scaling up the planning game in large-scale projects. In a qualitative case study at Ericsson they identified significant challenges in making long-term effort estimates in large-scale projects, because the associated product backlogs are too large. This made the teams involved in the study sceptical about effort estimation in general.

A case study by Tanveer et al. [19] showed that the accuracy of effort estimation in agile projects is affected by factors such as developers’ knowledge and experience, and the complexity of changes. The authors concluded that the explicit consideration of the aforementioned factors in the estimation process can support practitioners in obtaining more accurate estimates. Furthermore, they suggested that a tool combining expert knowledge and explicit consideration of cost drivers may improve the effectiveness of software effort estimation processes.

Based on the existing literature, we identified that:

- Expert judgment based effort estimation is the most commonly used approach in software industry.
- Similar approaches are used to estimate effort in both collocated and distributed contexts. The main differences are related to the cost drivers; geographic and temporal distances demand more communication and coordination effort.
- There is a lack of in-depth research about effort estimation processes in large-scale distributed projects and the unique factors related to this context that may impact effort estimation accuracy.

This study fills the aforementioned gap through a longitudinal case study. It is unique in its combination of quantitative and qualitative data to analyze the effort estimation process from multiple perspectives, such as the impact of a multi-staged estimation process and various factors impacting effort estimation accuracy.

### 3. Research methodology

To address the research questions, we have conducted an exploratory longitudinal case study [20]. In this section, we describe the case, the unit of analysis and the data collection, preparation, and analysis processes.

#### 3.1. Case description

The case studied in this paper is a large-scale distributed agile project with an increasing degree of global distribution during the studied period. The actual project was selected through convenience sampling in consultation with company representatives.

The project is concerned with the development and maintenance of a large telecommunication software product at Ericsson. The product originated in Sweden and has evolved for over 15 years and comprises a considerable amount of legacy code. Many technical and methodological changes were introduced during this time, such as changing the programming language from C++ to Java and changing the software development methodology from plan-driven to agile. The product is part of a large business solution that comprises other products.

During the period covered in this study, the case involved 188 employees (15 product-level architects, 134 developers working in 24 teams and 15 other supporting roles) distributed across Sweden (eight software development teams),
India (eleven software development teams), Italy (one software development team), USA (one software development team), Poland (two software development teams) and Turkey (one software development team).

The offshore locations were added in response to the growing demands for resources and to implement market-specific customizations. The last expansion was to Poland, where two teams were on-boarded in early 2016. An expansion to India took place between late 2014 and late 2015, where 10 teams were on-boarded in total. The site located in Turkey was decommissioned in 2014.

The developments teams are cross-functional and use agile practices in their daily work. Each team has from four to seven developers and a design lead, who is a senior developer. Due to the scale and level of distribution of the case, project managers use a mix of agile and plan-driven practices to manage and coordinate the work of the software development teams.

All software development teams receive an end-to-end responsibility for designing and implementing a task, which can be product customizations, trouble report fixing, product improvements, standardizations of market features and business use cases. These types of tasks resemble independent projects; they have specific start and end dates, responsible teams (one or more) and expected results. They can also be substantial in extent; the duration of a product customization, for example, varies from one to six months.

Software architects located in Sweden support all teams by responding to questions related to the product software architecture and by providing feedback on the teams’ work through code reviews. In urgent or particularly complex situations, the software architects also participate in actual coding.

3.2. Unit of analysis

In this case study, the unit of analysis is the effort estimation process employed in the sub-projects of the selected large-scale distributed agile project. We based our analysis on a subset of the collected data, focusing on product customizations (PCs), which are driven by Ericsson customers. Product customization projects are negotiated as fixed price contracts; failing to accurately estimate their effort may lead to budget overruns for the company. Accurate effort estimates for PC projects are therefore particularly important.

3.3. Data collection

To support methodological and data triangulation [20], we used the following data collection methods:

- **Archival research** – We analyzed managerial documents (plans, progress reports, time reports, solution specifications, effort estimation spreadsheets, process descriptions and team setup reports) of 60 PCs involving 18 different teams in total (9 in India, 7 in Sweden, 1 in Italy and 1 in the USA). We extracted the following data from these documents: effort estimation process description, effort estimates, actual effort, PC size, teams involved, team maturity, size of teams and customer type.

- **Unstructured individual interviews** - We held several unstructured interviews with a project manager to verify the consistence of the collected data.

- **Semi-structured individual interviews** – We interviewed two managers, a software architect, and a design lead to collect details about the effort estimation process employed in the case. We also talked to them to clarify the results of our data analysis, such as the main reasons that lead to the identified effort overruns. We covered with the interviews all roles involved in the case’s effort estimation processes. More details about the interviews are presented in Table 1.

Table 2 maps the research methods to the research questions. Note that we used both data collection methods for each of the research questions.

3.4. Data preparation

We used a significant amount of data from different sources that was extracted using different data collection methods. To increase data reliability, we asked people involved with the data point (e.g., software architects or project managers) for clarification whenever we identified an issue or inconsistency with a data observation and corrected the data observation, if necessary.
Table 1: Description of semi-structured interviewee sample.

<table>
<thead>
<tr>
<th>Role</th>
<th>Approach</th>
<th>Duration</th>
<th>Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>System manager</td>
<td>face-to-face interview</td>
<td>70 minutes</td>
<td>9 years</td>
</tr>
<tr>
<td>Architect</td>
<td>face-to-face interview</td>
<td>45 minutes</td>
<td>7 years</td>
</tr>
<tr>
<td>Project manager</td>
<td>face-to-face interview</td>
<td>1 hour</td>
<td>15 years</td>
</tr>
<tr>
<td>Design lead</td>
<td>interview via Skype</td>
<td>45 minutes</td>
<td>10 years</td>
</tr>
</tbody>
</table>

Table 2: Mapping RQs with the used data collection methods.

<table>
<thead>
<tr>
<th>RQ</th>
<th>Main method</th>
<th>Secondary method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interviews</td>
<td>Archival research</td>
</tr>
<tr>
<td>2</td>
<td>Archival research</td>
<td>Interviews</td>
</tr>
<tr>
<td>3</td>
<td>Archival research</td>
<td>Interviews</td>
</tr>
<tr>
<td>4</td>
<td>Archival research</td>
<td>Interviews</td>
</tr>
</tbody>
</table>

Furthermore, before running the quantitative part of the data analysis, it was necessary to aggregate the collected data. To do so, we created R scripts\(^2\), which were used to create a unified data set.

3.5. Data analysis

To analyze the quantitative data (archival research), we calculated descriptive statistics, plotted charts to graphically identify trends in the data and employed inferential statistics (hypothesis testing and regression analysis) to answer RQ2–RQ4. To answer RQ1 and RQ3, we analyzed the qualitative data using light-weight qualitative analysis\(^2\).

Before analyzing the accuracy of the effort estimates, we analyzed existing accuracy metrics. Traditionally, the Magnitude of Relative Error (MRE) is used frequently to calculate estimation accuracy. However, during the last decade, MRE has been criticized\(^2, 23\),\(^2, 24\) for its uneven treatment of under- and over-estimation. BRE and BREbias are more balanced metrics, i.e. it evenly balances the over and underestimation. Due to this reason, Balanced Relative Error (BRE and BREbias) have been used in many recent studies on software effort estimation\(^2, 26\). BRE just calculates the magnitude of the estimation error, while BREbias measures both the size and the direction (over- or under-estimation) of the estimation error. Therefore, we used BRE (Equation 3.5) and BREbias (Equation 3.5) in our analysis of the effort estimates’ accuracy and bias.

\[
BRE = \frac{|\text{actual effort} - \text{estimated effort}|}{\min(\text{actual effort}, \text{estimated effort})}
\]

\[
BREbias = \frac{(\text{actual effort} - \text{estimated effort})}{\min(\text{actual effort}, \text{estimated effort})}
\]

Before selecting an appropriate statistical test to analyze our results, we ran a Shapiro-Wilk normality test, which indicated that the collected data does not come from a normal population. Therefore, we applied non-parametric tests Wilcoxon signed-rank test\(^2\) for RQ2 and Kruskal-Wallis\(^2\) and Mann-Whitney\(^2\) for RQ4. For the same reason, we used a non-parametric measure of Cliff’s delta\(^2\) to measure effect size.

\(^2\)cran.r-project.org
4. Results

4.1. RQ1: How is effort estimation carried out?

The aim of RQ1 is to explain how effort estimation processes are carried out in this very large distributed project. As shown in Figure 1, several people in different roles work collaboratively to perform effort estimation. First, we briefly describe these roles and the tasks that they perform in the effort estimation processes.

![Effort estimation process (PC = Product customization).](image)

- Customers: Ericsson customers demand new features in the product and have to approve the quoted price before development work on a product customization (PC) can begin.
- Customer Unit: Customer unit is part of Ericsson and is responsible for negotiations with the customers. It initiates the request for a new PC by specifying the requirements received from customers.
- System Managers: As stated previously, the product is part of a larger system comprising several products. System managers work at the overall system level. They receive request for new PCs from the customer unit, propose high level solutions and coordinate with the project managers and the architects of each concerned product in the system.
- Project Manager: Each product has a project manager, who is responsible for managing development teams across different sites and for planning, scheduling and coordinating the work on the PCs.
- Technical Writers: Technical writers are responsible for preparing PC documentation for the customers.
- Software Architecture Team (SAT): Each product has a team of architects, which is responsible for managing the evolution and integrity of the architecture. Besides proposing an initial solution and estimates for the requested PCs, they also provide design support, when required, to the development teams.
- Development Team: The development team is responsible for actually implementing and verifying the PCs.

The system managers, project managers, SAT and the development teams are part of the development unit. To fully understand the estimation processes in the case, we interviewed a representative from each of these roles. The interviewee sample (see Table 1) was selected by the unit manager, who was the contact person at Ericsson for this study.

Effort estimation is carried out at two levels in the case project, both using an expert judgment based estimation approach. First, a high level *quotation estimate* is prepared when a request for a new PC is initially analyzed. At this stage, it is not known which development team will actually develop the PC. Next, in a more detailed analysis phase, a more refined *analysis estimate* and a solution are proposed. At this stage, the development team that is going to develop the PC is known in most cases. We describe these estimation processes in the following.
**Quotation Estimate (QE).** The process starts when a new PC request is initiated by the customer unit. The customer unit interacts with customers, and specifies the customer requirements. Once a new PC request is initiated, the following steps are performed:

- The system manager scans the new PC requests for an initial analysis. The PCs with relatively clear requirements are selected for further analysis, the remaining ones are sent back for further clarification.
- The selected PC requests are compared with previous PCs to identify any potential for reuse. The PCs are assigned to relevant system managers depending upon their availability. In case a PC request is identical to a previously developed PC, the customer unit is asked to order a resell instead of a new implementation.
- The assigned system manager analyzes the type of feature, and suggests a high level solution.
- The solution is shared with the SAT of the concerned product for further analysis. The SAT is also required to provide the QE. Normally, One member of the SAT leads this analysis.
- The QE is an interval estimate wherein the interval is marked by three points: min, max and average most likely estimate in hours. These intervals represent the size and complexity of PCs and are referred as small, medium, large and extremely large. Since the case has been going on for more than 15 years, an empirical classification has been developed and evolved, which provides min, max and average hours for each interval estimate. The SAT lead uses this empirical classification to specify the relevant interval estimate for the assigned PC.
- The QE along with the high level solution are shared with the customer unit.
- The customer unit approves or disapproves the QE and the high level solution.

This process for preparing the quotation estimate takes up to four or five days. It is important to note that at the quotation stage the goal is to support the customer unit in deciding whether there is a strong business case in moving to the next phase of the PC development. The focus of effort estimation at this stage is on providing a best cost indication without spending too much time. Therefore, small estimation errors are not a problem at this stage.

**Analysis Estimate (AE).** This process is carried out only for those PCs whose quotation estimates have been approved by the customer unit. The following steps are performed in this process.

- The system manager asks the SAT for the relevant products to prepare their respective designs, and to provide effort estimates in person hours.
- At this point, it is usually known which development team is going to develop the PC.
- One SAT member leads the work on each PC.
- The SAT lead specifies the design in detail. The SAT lead also estimates the effort, in person hours, required to develop and verify the PC. If the development team has a sufficient maturity and design competence (maturity level C and D, see Subsection 4.3 for details), the team’s design lead works together with the SAT lead during the design and estimation activities.
- The SAT lead presents the design and associated estimate to other SAT members for further discussion.
- The project manager reviews the estimate, and adjusts it, if required, based on the recent productivity and capacity of the concerned development team.
- The system manager forwards the design solution and corresponding estimates to the customer unit for approval.
4.2. RQ2: How accurate are the effort estimates?

The question aims to analyze the accuracy levels of the two estimates (i.e. QE and AE). This analysis is performed from the following two aspects:

- Overall trend analysis: To identify the dominant trends, we analyze how frequently the effort is over/under estimated corresponding to different estimation error levels (e.g., 25% to 50%). In this part, we do not use the mean or median of BRE and BREbias, as a single number (i.e. mean or median) does not fully convey the trend.

- Estimation error analysis: We analyze the accuracy of the effort estimates using BRE\(^3\) and BREbias, and compare the accuracy of the two estimates (QE and AE) to establish whether the re-estimation at the analysis stage (AE) improves the estimation accuracy over the original estimate (QE).

4.2.1. Overall trend analysis

As described in Subection 4.1, the QE is an interval estimate. Our results show that actual effort falls within the estimated interval in 26.7% of the cases (16 PCs), while 36.7% (22 PCs) are overestimated and 36.7% (22 PCs) are underestimated.

Table 3 summarizes for both estimates, the frequency of over/under/accurately estimated PCs corresponding to different levels of estimation error expressed here as percentages. For QEs the PCs are almost evenly distributed in the three categories (over-, under- and accurately estimated PCs), while in case of AEs underestimation is more common (50% PCs) as overestimation (16.7%). The gross overestimation instances are more common for QEs, where relatively less details about PCs are available, and the team that is going to develop the PC is not known.

Table 3: Accuracy distribution of the two estimates (QE and AE).

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of overestimated PCs by</th>
<th>No. of accurately estimated PCs</th>
<th>No. of underestimated PCs by</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Above 100%</td>
<td>51–100%</td>
</tr>
<tr>
<td>QE</td>
<td>18 (30%)</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>AE</td>
<td>10 (16.7%)</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

4.2.2. Estimation error analysis

In this part we analyze and compare the accuracy of the two effort estimates using BRE and BREbias measures. The results are displayed in Table 4. The QEs’ magnitude of error (mean BRE 1.26, median BRE 0.69) is considerably higher than the AEs’ magnitude of error. The mean and median BREbias values for both estimates show that the estimates are biased towards underestimation, i.e. optimism. The underestimation bias, however, is more significant for AEs (mean BREbias 0.41, median BREbias 0.26).

Table 5 displays the results of a Wilcoxon signed-rank test to see if the differences between the two estimates, for our sample of 60 PCs, are statistically significant. The statistical tests are conducted in SPSS, and we report all relevant data from SPSS, and also the effect size measure (Cliff’s delta), when applicable. The results show a statistically significant difference (p = 0.04) between the BREbias of the two estimates with small to medium size effect (Cliff’s delta -0.14). However, the p-value of the test for the BRE of the two estimates is slightly over 0.05 (0.06) and therefore not statistically significant. The results in Table 4 and 5 show that the AEs are more accurate (median BRE 0.49) than the QEs, but contain significantly higher optimism (i.e., underestimation) bias.

4.3. RQ3: Which factors impact the accuracy of the effort estimates?

Through the interviews, we identified the following factors that potentially impact the accuracy of the two effort estimates:

---

\(^3\)for QE we used mid value, which is the most likely estimate, to compute BRE.
Table 4: Mean and Median BRE and BREbias for both estimates (QE and AE).

<table>
<thead>
<tr>
<th></th>
<th>BRE QE</th>
<th>BRE AE</th>
<th>BREbias QE</th>
<th>BREbias AE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>1.26</td>
<td>0.81</td>
<td>0.25</td>
<td>0.41</td>
</tr>
<tr>
<td>Median</td>
<td>0.69</td>
<td>0.49</td>
<td>0.05</td>
<td>0.26</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>1.48</td>
<td>0.97</td>
<td>1.93</td>
<td>1.20</td>
</tr>
</tbody>
</table>

Table 5: Testing the two estimates (QE and AE) for significant differences.

<table>
<thead>
<tr>
<th>BRE AE – BRE QE</th>
<th>N</th>
<th>Mean rank</th>
<th>Sum of ranks</th>
<th>Z</th>
<th>P-value</th>
<th>Size effect (Cliff’s delta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative Ranks</td>
<td>34</td>
<td>33.32</td>
<td>1133</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positive Ranks</td>
<td>25</td>
<td>25.48</td>
<td>637</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ties</td>
<td>1</td>
<td>-1.87</td>
<td>.06</td>
<td>-0.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BREbias AE – BREbias QE</th>
<th>N</th>
<th>Mean rank</th>
<th>Sum of ranks</th>
<th>Z</th>
<th>P-value</th>
<th>Size effect (Cliff’s delta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative Ranks</td>
<td>20</td>
<td>30.80</td>
<td>616</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positive Ranks</td>
<td>39</td>
<td>29.59</td>
<td>1154</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ties</td>
<td>1</td>
<td>-2.03</td>
<td>.04</td>
<td>0.14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **PC size** – Understandably, the estimation of larger PCs is perceived to be more challenging. The interviewees observed that PC size, measured in actual effort (work hours), beyond a certain level brings with it all the complexities of scale, and therefore impact the accuracy of the effort estimates. Using the actual effort spent, the PCs were classified based on the classification mechanism described in Section 4.1 into four types: small, medium, large and very large. In our regression analysis, the variable PC size is used as an ordinal variable with values 0 (small), 1 (medium), 2 (large) and 3 (very large).

- **Customer priority** – A set of PCs (26) in our sample belong to one large customer project. The case company ensures a dedicated team capacity for the implementation of these PCs, in return for the guarantee that the company will be paid for the work done on these PCs. The provision of dedicated capacity for these PCs is meant to prioritize them during planning phase. They are not required to wait for the availability of capacity to start the work. Therefore, the work is initiated relatively quickly on most of the high priority PCs after the quotation approval by the customer unit. These 26 PCs are categorized as PCs having a customer with high priority, while the remaining 34 as PCs from customers with a normal priority.

- **Maturity of the development team** – The development teams, working across the globe, have varying levels of maturity. As teams become more mature, they work more independently and need less support from the product level architects. We use the same maturity levels as Britto et al. [31], who studied the same case company.
  - **Maturity level A** – Teams at level A have very little understanding about the product’s code and architecture. Therefore, they need much support and guidance from the architects, even for less complex PC projects. Typically, newly on-boarded teams are at level A.
  - **Maturity level B** – Teams at level B can implement non-complex tasks independently without much help from the product level architects. The architects still review most of the design and code.
  - **Maturity level C** – Teams at level C have a good architectural understanding and are able to implement complex solutions. These teams perform code reviews and approvals independently. When critical components are affected, the product level architects are responsible for approval. The architects also support these teams with the design of the technical solution.
Maturity level D – Teams at level D are very experienced, and are able to autonomously develop complex solutions that affect or include critical components. Their code does not need the approval from the product level architects.

In our sample of 60 PCs, 19 PCs were developed by level A teams, 12 by level B teams, 26 by level C teams, and 3 by level D teams. The variable team maturity is used as an ordinal variable in our regression analysis with the four values: 0 (maturity level A), 1 (maturity level B), 2 (maturity level C), 3 (maturity level D).

Multi-site development – In our sample, 6 PCs were developed in a multi-site arrangement wherein development teams in different geographical locations collaboratively worked together to develop the PCs. Multi-site development is also perceived as a factor impacting the actual effort spent, and thus the accuracy of the effort estimates. However, we decided not to include this factor in our regression analysis for two reasons: 1) Very few PCs were in this category, and 2) the multi-site PCs were particularly large. It would therefore not be possible to differentiate the impact of PC size and multi-site setting. We, however, analyze the impact of multi-site development in more detail in Section 4.4.4.

We performed multiple linear regression to investigate whether the factors identified above significantly impact the accuracy of the effort estimates. We used simple linear models to understand the relationship between the effort estimates' accuracy and the factors described above. We applied stepwise regression in SPSS using backward elimination with an alpha value of 0.1 to remove variables.

4.3.1. Regression model of quotation estimates (QEs)

The results of the regression analysis showed that PC size (p < 0.005) and team maturity (p = 0.007) have a statistically significant relationship with BREbias of QEs (see Table A.1 and A.2 in Appendix A for details), while PC priority (p = 0.863) has no such relationship.

Given the details in Table A.2, the regression model is described as:

\[
\text{BREbias of QE} = -0.73 + 1.22 \times \text{PC Size} - 0.56 \times \text{Team Maturity}
\] (1)

Model (1) suggests that the QEs are overestimated by 73% for the small size PCs (PC size = small = 0) and immature teams (Team Maturity = Level A = 0). However, increases in PC size increase the BREbias of QEs, i.e. underestimation bias, while increases in team maturity decrease the BREbias of QEs, i.e. overestimation bias. In other words mature teams are more likely to complete the work within the estimated time. These factors are investigated in detail in Section 4.4 and are further discussed in Section 5.

Model (1) is the one with highest value (0.40) of adjusted \(R^2\) (see Table A.3 in Appendix A). The value of adjusted \(R^2\) does not necessarily increase when more predictors are added in the model, and thus is a suitable measure in identifying appropriate predictors for a model. Including variable PC Priority (p = 0.863) slightly reduces the value of adjusted \(R^2\).

4.3.2. Regression model of analysis estimates (AE)

Our regression analysis showed that PC size (p = 0.001) and priority (p = 0.028) have a statistically significant relationship with BREbias of AEs, while team maturity (p = 0.929) has no such relationship. The regression details are shown in Tables B.1 and B.2 in Appendix B.

Given the details in Table B.2, the regression model is described as:

\[
\text{BREbias of AE} = -0.07 + 0.57 \times \text{PC Size} - 0.69 \times \text{Customer Priority}
\] (2)

Model (2) shows that for small sized (PC size = small = 0) PCs with normal priority (customer priority = normal = 0), effort is slightly overestimated by 7%. An increase in PC size (PC size = medium = 1 or PC size = large = 2 or PC size = very large = 3) results in an increase in underestimation bias. However, the negative sign of priority in model (2) indicates that an increase in priority from normal to high (i.e. customer priority = normal = 0 to customer priority = high = 1) decreases the underestimation bias or leads to an overestimation bias. In other words, the high priority PCs incur lesser magnitudes of effort overruns with respect to the analysis stage estimates. However, PC size
is still a huge factor even with high priority PCs. When PC size becomes large or very large, the overall result is still a considerable underestimation bias. These factors are further analyzed in detail in Section 4.4, and further discussed in Section 5. Team maturity is not significantly impacting BREbias of AEs, since the estimators at the analysis stage are normally aware of which teams are going to develop a PC. Therefore, Team maturity already is accounted for in the analysis stage estimates.

Model (2) is the one with highest value (0.40) of adjusted $R^2$ (see Table B.3 in Appendix B). Including variable Team Maturity ($p = 0.929$) reduces the value of adjusted $R^2$.

4.3.3. Regression assumptions

The examination of the residuals of both models revealed that the regression assumptions are not violated, with one minor exception. The histograms and normal plots show some minor deviations from the normal distribution (see Figures A.1–A.3 and B.1–B.3 in the Appendix). Besides visual analysis using these plots, we also tested these assumptions using relevant tests.

- Independence of residuals was tested using a Durbin-Watson test [32]. The results for both models were in the required range of 1.5 and 2.5 (2.01 for model (1) and 2.09 for model (2)).
- Constant variance in residuals (Homoscedasticity) was tested using a Breusch-Pagan and Koenker tests [33]. For both models, the null hypothesis of homoscedasticity was not rejected due to p-values > 0.05: Breusch-Pagan (model (1), $p = 0.08$; model (2), $p = 0.41$), Koenker (model (1), $p = 0.28$; model (2), $p = 0.75$).
- Normal distribution of residuals was tested using a Kolmogorov-Smirnov test [34]. For both models the tests showed that the residuals are normally distributed, i.e. the null hypothesis of normality could not be rejected (model (1), $p = 0.20$; model (2), $p = 0.08$). According to a Shapiro-Wilk test, the residuals of model (1) are normally distributed ($p = 0.17$), but not the residuals of model (2) ($p = 0.003$).

Furthermore, tolerance values (see Table A.2 and B.2 in Appendix A and B respectively) show the absence of multicollinearity for both models. As for the outliers diagnostics, only one observation was outside the three sigma limit in case of model (1). The maximum cook’s distance [35] for any observation was 0.11 (much less than critical value of 1), indicating that there are no highly influential observations. Likewise for model (2), there is only one observation that is outside the three sigma limit, and the maximum cook’s distance was 0.31.

Besides BREbias, we also attempted to apply regression analysis using BREs of both QEs and AEs. In case of BRE of QEs, regression assumptions of homoscedasticity and normality were violated. For AEs, the analysis did not find any significant relationship between the outcome and independent variables.

4.4. RQ4: How do the identified factors impact the accuracy of the effort estimates?

This question aims to investigate in detail how the factors, identified as statistically significant above, impact the accuracy of the effort estimates.

4.4.1. PC size

The 60 PCs included in this study are of varying sizes in terms of actual effort spent. These PCs are divided into four categories (see Table 6) using the intervals of QEs (see Section 4.1). The aim of this question is to investigate in depth how PC size impacts the accuracy of the two effort estimates.

<table>
<thead>
<tr>
<th>PC size</th>
<th>No.</th>
<th>Mean (hours)</th>
<th>Median (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>10</td>
<td>284.9</td>
<td>290</td>
</tr>
<tr>
<td>Medium</td>
<td>28</td>
<td>833.1</td>
<td>699</td>
</tr>
<tr>
<td>Large</td>
<td>12</td>
<td>1931.8</td>
<td>1561</td>
</tr>
<tr>
<td>Very Large</td>
<td>10</td>
<td>5735.5</td>
<td>4811.5</td>
</tr>
</tbody>
</table>

Table 6: Mean and median actual effort for all 60 PCs in work hours grouped by PC size.
Quotation Estimate (QE). Figure 2 depicts the QEs’ BREBias (vertical axis) for all 60 PCs that are sorted from left to right in increasing order of actual effort spent. The bars below 0 represent overestimated PCs (i.e. estimate > actual), while above 0 are underestimated PCs (i.e. estimate < actual).

The results clearly show a contrasting pattern with respect to over- and underestimation; smaller PCs tend to be overestimated, while larger PCs tend to be underestimated.
Grouping BREBias by PC size shows that an increase in size seems to be related to an increase in estimation error and underestimation bias (see Figure 3). We therefore applied a Kruskal-Wallis test [28] to see whether there are statistically significant differences between the QE’s BREbias of the four PC sizes. The results shows that the BREbias of QEs correlates significantly with PC size ($p < 0.0001$ at $\alpha = 0.05$).

Figure 4: BREbias of AEs of PCs arranged on actual effort.

Analysis estimate (AE). AEs show a similar trend, though the division of over- and underestimated PCs is not that clear (see Figure 4). Some smaller PCs are underestimated, and a few larger PC are slightly overestimated. However, underestimation becomes more frequent as we move to large and very large PCs.

Figure 5 shows AEs’ BREBias grouped by PC size. The results show that underestimation bias increases with the increase in PC size. A Kruskal-Wallis test showed that there are statistically significant differences between the BREbias of different PC sizes ($p < 0.0022$ at $\alpha = 0.05$).

For both estimates (QE and AE), there are statistically significant differences in accuracy levels for different PC sizes. The differences are pronounced though for QEs.

Table 7 lists the mean and median BREbias of the two estimates for the four PC sizes. The results show two trends: 1) For both QEs and AEs, an increase in size is related to an increase in underestimation bias. The increase in bias is larger for QEs, indicating that PC size impacts QEs more strongly; 2) For almost all PC sizes, the magnitude of estimation error decreases as we move from QEs to AEs, i.e. the re-estimation at the analysis stage (AE) improves the accuracy of the effort estimates.

<table>
<thead>
<tr>
<th>PC size</th>
<th>Mean (QE)</th>
<th>Mean (AE)</th>
<th>Median (QE)</th>
<th>Median (AE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>-1.18</td>
<td>-0.40</td>
<td>-0.99</td>
<td>-0.23</td>
</tr>
<tr>
<td>Medium</td>
<td>-0.34</td>
<td>0.35</td>
<td>-0.21</td>
<td>0.34</td>
</tr>
<tr>
<td>Large</td>
<td>1.31</td>
<td>0.76</td>
<td>0.40</td>
<td>0.29</td>
</tr>
<tr>
<td>Very Large</td>
<td>2.10</td>
<td>0.97</td>
<td>1.30</td>
<td>0.56</td>
</tr>
</tbody>
</table>
4.4.2. Customer priority

Our dataset comprises 26 PCs with high priority (1 small, 10 medium, 7 large, 8 very large) and 34 PCs with normal priority (9 small, 15 medium, 7 large, 3 very large). To make the samples and the results comparable, we excluded the small PCs, since there was only 1 small PC with high priority. The mean and median BREBias of the remaining 50 PCs (25 PCs each with high and normal priority) are summarized in Table 8. These results show the following interesting patterns:

- The estimation accuracy improves considerably as we move from quotations (QE) to analysis (AE) stage estimation for high priority PCs. However, for normal priority PCs it is other way around.
- At the quotation stage, the estimates (QE) of normal priority PCs are more accurate as compared to the high priority PCs.
- At the analysis stage however, the estimates (AE) of the high priority PCs are more accurate.

These results indicate that there is a relation between customer priority and estimation accuracy. However, a Mann-Whitney test [29] did not detect a statistically significant difference in the distributions of BREBias of QEs and AEs across the two priorities ($p < 0.46$ for QEs and $p < 0.23$ for AEs; $\alpha = 0.05$).

<table>
<thead>
<tr>
<th>Customer priority</th>
<th>Mean (QE)</th>
<th>Mean (AE)</th>
<th>Median (QE)</th>
<th>Median (AE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>0.68</td>
<td>0.31</td>
<td>0.32</td>
<td>0.20</td>
</tr>
<tr>
<td>Normal</td>
<td>0.41</td>
<td>0.84</td>
<td>0.10</td>
<td>0.54</td>
</tr>
</tbody>
</table>

4.4.3. Team maturity

Team maturity varies from level A (lowest level) to level D (highest level), see Section 4.3. To facilitate an analysis, we categorized teams at levels A and B as immature teams and teams at levels C and D as mature. PCs of
all sizes are developed by these two types of teams.

- Number of PCs by immature teams: small (6), medium (14), large (7), very large (4)
- Number of PCs by mature teams: small (4), medium (14), large (5), very large (6)

Table 9: Mean and median BREBias for both estimates (QE and AE) for immature and mature teams.

<table>
<thead>
<tr>
<th>Team Maturity</th>
<th>No. of PCs</th>
<th>Mean (QE)</th>
<th>Mean (AE)</th>
<th>Median (QE)</th>
<th>Median (AE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immature teams</td>
<td>31</td>
<td>0.68</td>
<td>0.60</td>
<td>0.03</td>
<td>0.33</td>
</tr>
<tr>
<td>Mature teams</td>
<td>29</td>
<td>-0.20</td>
<td>0.26</td>
<td>0.08</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 9 shows that the mean effort estimate accuracies are better for mature teams, but quite similar for the medians. As for the estimation bias, the results show a considerable reduction in the underestimation bias for mature teams. Together, it means that immature teams are more likely to incur larger effort overruns as compared to the mature teams. However, a Mann-Whitney test did not show any statistically significant differences in the distributions of BREbias for QEs and AEs across two types of teams.

4.4.4. Multi-site development

Six PCs in our dataset were developed by geographically distributed teams (referred here as multi-site development), while the remaining 54 were developed by co-located teams with remote support provided by software architects located in Sweden. The development of these 6 PCs is led by a team at one site. The teams in other sites contribute by developing relatively smaller and specific parts due to special competencies at these sites. Multi-site development could also have an impact on the accuracy of the effort estimates.

The six multi-site development PCs have mainly large sizes (very large: 3, large: 2, medium: 1), whereas the PCs developed in a co-located setting have all types of sizes. To make a fair comparison, we excluded all small PCs, leaving behind 44 PCs of medium, large and very large size for the co-located case. Table 10 summarizes effort estimation accuracies for co-located and multi-site PCs and shows that both mean and median BREbias are higher for multi-site PCs.

These results indicate that multi-site development in the case leads to relatively larger effort overruns. However, a Mann-Whitney test did not find the differences in the distribution of the BREbias for both QEs and AEs (44 co-located PCs and 6 multi-site PCs) to be statistically significant across two types of PCs, i.e. co-located and multi-site. This may be due to the very small number of the PCs in the multi-site category.

Moreover, in order to see whether multi-site development phenomena has a statistically significant impact on the results of the statistical tests reported in 4.2 (Table 5), Section 4.4.1, 4.4.2 and 4.4.3, we removed the multi-site PCs from our sample, and re-performed these tests. All tests returned similar results with minor changes in the resulting p-values. The changes were so insignificant that they did not lead to any reversal in the corresponding null hypotheses testing results.

Table 10: Mean and median BREBias for both estimates (QE and AE) for co-located and multi-site PCs.

<table>
<thead>
<tr>
<th>PC type</th>
<th>Mean (QE)</th>
<th>Mean (AE)</th>
<th>Median (QE)</th>
<th>Median (AE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-located</td>
<td>0.41</td>
<td>0.41</td>
<td>0.09</td>
<td>0.34</td>
</tr>
<tr>
<td>Multi-site</td>
<td>1.51</td>
<td>1.79</td>
<td>1.28</td>
<td>1.18</td>
</tr>
</tbody>
</table>

4.5. Results from the interviews

Besides understanding the effort estimation processes used in the case, the interviews were also used to discuss the results with the participants. Information about the four interviewees are provided in Table 1 in Section 3. The interviewees were asked about the following: 1) their background and experience in the company and the case, 2) their role in the effort estimation processes, 3) how various factors impact the accuracy of the effort estimates, and 4) the main reasons for effort overruns. The information will be used in the discussion section to further explain the results presented above.
5. Discussion

In this section we further discuss the results presented in Section 4 in the light of our discussion with interviewees.

5.1. Effort estimation processes (RQ1)

Our interviews show that it is mainly the scale of a PC project that makes it challenging to estimate and plan. Scale includes a number of factors beyond PC size, such as the number of sites involved, the number of stakeholders involved in the process, the size and complexity of the legacy code etc. The coordination between different types of stakeholders involved in the estimation processes also introduces challenges (see Section 4.1 for stakeholders details). Research shows that coordination challenges exacerbate issues in multi-team projects [12]. Despite these challenges, a reasonable proportion of PCs (1/3, see Table 3) are accurately estimated within a 25% error margin at both quotation and analysis stages.

The involvement of teams in the analysis stage estimation process is an important practice. The interviewed architect and project manager both suggested that it helps in arriving at better estimates. Whenever possible, the teams should provide their input in the estimation process. However, such a practice is only feasible with mature teams, i.e. teams including team members with sufficient expertise and knowledge about the product’s design and architecture. The newly on-boarded teams need more time to come to this level.

5.2. Accuracy of estimates (RQ2)

Our analysis showed that, overall, underestimation is the dominant trend at both quotation and analysis stages. This is in line with the results in other estimation studies [7]. The interviewees attributed the inaccuracies in the effort estimates mainly to the following challenges that they encounter in this very large-scale distributed agile project:

- Requirements related issues such as lack of details and changes in the requirements.
- Lack of expertise of newly on-boarded teams results in delays.
- Dependencies (such as for code reviews) on specific human resources (e.g., product architects) introduce delays.
- Project scale and distribution across multiple sites.
- Underestimating the technical complexity of some large PCs.

All interviewees stressed the importance of better requirements to be able to estimate more effectively. There is a lot of uncertainty at the quotation stage, and late changes/additions in the requirements eventually result in underestimation.

Since the goal of the quotation stage estimation (QE) is to provide a relatively quick estimate to the customer unit, expectations on accuracy are not as high as they are for analysis stage estimates (AE). Our results show that the AEs actually are more accurate than the QEs. However, underestimation bias increases at the analysis stage. The interviewees suggest that the availability of more detailed information and knowledge of the teams and their input at the analysis stage might have increased the estimators’ optimism, resulting in relatively higher underestimation.

Underestimation bias is relatively less evident at quotation stage, where in 30% of the PCs the effort is overestimated (see Table 3). In these cases, the estimators are more conservative in their estimates mainly due to a high level of uncertainty, lack of detail in the requirements, and that the PCs have not yet been assigned to a development team (i.e. the development team’s maturity level is unknown).

5.3. Factors affecting the effort estimates (RQ3 and RQ4)

We identified four factors that impact the accuracy of the effort estimates (Section 4.3) and analyzed how these factors affect the effort estimates (Section 4.4). We now discuss these factors in the light of the interview results.
5.3.1. PC size

It is more challenging to estimate larger PCs. We found statistically significant differences in the distribution of BRebias, for both types of effort estimates (QE and AE), across PCs of different sizes. For smaller PCs there is tendency to overestimate, while for large and very large PCs our results show a large underestimation bias. The interviewees attribute this to the inherent difficulty in estimating large PCs, whereby estimators underestimate the likely complexities and technical challenges. Moreover, the development of large PCs involve more unanticipated challenges and risks related to project management, coordination, design and architecture, resulting in potentially large delays. Moløkken-Østvold and Furulund [36] also identified a tendency of larger effort overruns for projects with larger actual effort.

5.3.2. Team maturity

Team maturity also impacted the effort estimates. Our results show that PCs developed by mature teams have more accurate effort estimates. In the present case 11 teams were on-boarded relatively recently. They are the most immature teams in our dataset, and are incurring higher effort overruns than the more mature teams. In order to mature, newly on-boarded teams require a lot of time and mentoring support from product architects in Sweden [31]. The mentoring support provided by the product architects is critical to achieve technical consistency, which otherwise is hard to maintain in such projects that scale up involving several agile teams at different levels of maturity [12].

At the quotation stage, the estimators do not know which team(s) will work on the PC being estimated. This might lead them to estimate pessimistically. The interviewees suggested that mature teams perform better than the estimates due to a better understanding of the product architecture and the high technical competence. At the analysis stage estimation, team maturity is not a significant factor according to our regression analysis (see Section 4.3). This might be due to the fact that the team’s maturity is already accounted for in the AEs, as the estimators are aware of the team which is going to work on the PC being estimated.

5.3.3. Customer priority

Customer priority was also identified as a factor impacting the effort estimates’ accuracy. The results show a contrasting pattern, i.e. effort estimates of PCs with high priority are more accurate at analysis stage (AE) and less accurate at quotation stage (QE), while it is the other way around in case of PCs with normal priority. The interviews revealed that for high priority PCs there is relatively more emphasis to shorten lead times by ensuring dedicated capacities for these high priority PCs. Moreover, as the work is initiated relatively quickly on most of the high priority PCs, more is known when the estimation at the analysis stage is performed. More information implies less uncertainty, which could be a reason for the relatively better analysis stage estimates (AE). The interviewees attributed the relatively higher inaccuracy of QEs for high priority PCs mainly to the late changes in requirements after the quotation stage, which is much more common in case of high priority PCs.

5.3.4. Multi-site development

Six PCs in our dataset were developed in multi-site settings. It is too small a number to perform a generalizable quantitative analysis. Nevertheless, in our limited sample the multi-site PCs have a stronger underestimation bias than co-located PCs, i.e. they incur larger effort overruns (as compared to the estimates). The interviewees attribute this to the inherent difficulties related to communication and coordination present in multi-site arrangements. Global barriers related to time, distance and culture in multi-site development have been identified as important cost contributors in other studies as well [15]. Herbsleb and Mockus [6] found that multi-site development introduces significant delays during the task development. Dikert et al. [12] identified that managers in several organizations find it challenging to create and estimate user stories in large scale distributed agile projects. This impact of multi-site development on effort estimation needs to be further investigated, preferably with a larger dataset.

6. Threats to validity

For our discussion, we use the classification of threats to validity by Runeson and Höst [20]. Furthermore, we also discuss conclusion validity [37].
Reliability is related to the repeatability of a study and how data is collected and analyzed [20]. To minimize this threat, we designed and followed an explicit, detailed case study protocol, following the guidelines by Runeson and Höst [20]. Furthermore, several researchers were involved in the design and execution of our investigation to minimize dependencies based on particular individuals. In addition, our observations and findings were verified with the company representatives to avoid false interpretations.

Internal validity is related to factors that researchers are unaware of or cannot control regarding their effect on the variables under investigation [20]. The effort estimates’ accuracy, how the estimates are obtained (estimation process) and actual effort in projects are constructs influenced by different factors, such as team maturity. It is therefore difficult to identify causal relationships between them, since there are many confounding factors. To mitigate this threat, we accounted for factors that are reported by related literature and also by interviewing people in the case company who are involved with effort estimation in the studied case (data triangulation). Regarding the qualitative part of our research (semi-structured interviews), the main internal validity threats are investigator bias and interviewee bias. To mitigate these threats, three researchers were involved with the design of the interview guides (investigator triangulation). We mitigated interviewee bias by interviewing people with different roles (data triangulation).

Construct validity reflects how well the measures used actually represent the constructs the study intends to measure [20]. The main threat of this category related to our investigation is the mono-method bias (only one method to measure a construct). Although just one measurement method was used to measure each construct, we cross checked different documents to increase the reliability of the collected data, with special focus on the collected actual effort. For example, we compared the data on progress reports and time reports to see whether the hours reported were in line with the number of people working in a given week in a particular task. Finally, we also conducted unstructured interviews with a project manager to further verify the consistency of the collected data.

External validity is concerned with the generalizability of the findings [20]. Since we employed the case study method, our findings are strongly bounded by the context of our study. In addition, the investigated case involved only one project in one company. To mitigate this threat, we made an attempt to detail the context of our study as much as possible, complying with corporate confidentiality concerns. To mitigate this threat, we made an attempt to describe the context of our study in as much detail as possible. This makes it easier to relate the present case to similar cases.

Conclusion validity is concerned with the correctness of conclusions regarding relationships in the analyzed data [37]. The main threats of this category related to our investigation are the low reliability of measures due to noise and low statistical power. To mitigate the first threat, we conducted an unstructured interview with a project manager to verify the consistency of the collected data. Regarding statistical power, we investigated the largest number of tasks possible. However, it is important to emphasize that it is just a sample and does not cover all product customization tasks carried in the investigated case.

7. Conclusion

This study reports the results of a case study conducted in Ericsson AB with the aim to investigate the effort estimation in a large-scale distributed agile project. Ericsson AB is a large multi-national telecommunication vendor with headquarters in Sweden. The case is related to a large telecommunication product, which is being evolved for over 15 years by agile teams distributed across Sweden, India, Italy, USA, Poland and Turkey. The studied product is part of a large business solution that includes other products as well. We used archival research to collect data about 60 Product Customization (PC) tasks involving 18 different teams in Sweden, India, Italy and USA. The collected data included effort estimates, actual effort, teams involved, team maturity and customer type. We conducted interviews to understand the context, and also to discuss and explain the results.

RQ1: effort estimation process. Effort estimation process is carried out at two stages in the case: quotation and analysis. At the quotation stage, the system manager along with the SAT prepares the QE to support the customer unit in deciding whether there is a business case in approving the PC for development. At the analysis stage, a more detailed analysis is performed of those PCs whose QEs have been approved by the customer unit. The assigned SAT member leads this analysis to propose a solution and effort estimate of the PC. At this stage, in most case the development team is already assigned to the PC under analysis. If the assigned team has design competence, the SAT member involves the team’s design lead in the analysis process.
RQ2: accuracy of the effort estimates. The results showed that underestimation is the more frequent trend at both quotation and analysis stages. The estimates at the analysis stage (AEs) were found to be more accurate than the QEs. However, they have significantly higher optimism bias as compared to QEs.

RQ3 and RQ4: factors impacting effort estimates. Based on our analysis of 60 PCs from one project within one organization, we identified PC size, team maturity, PC priority and multi-site development as factors that impact the accuracy of the effort estimates. Through regression analysis we found that: 1) PC size and team maturity have statistically significant relationship with BREbias of QEs, and 2) PC size and PC priority have a statistically significant relationship with BREbias of AEs. The results indicate that PC size has a strong influence on the effort estimates’ accuracy and bias, i.e. increase in PC sizes was found to be strongly related to the rise in the underestimation bias. We also identified that the mature teams, due to their experience and technical competence, are less likely to incur large delays. Furthermore, PCs with high priority in terms of dedicated capacity have better analysis stage estimates mainly due to the early start of work on them. Furthermore, the results also indicated that the multi-site PCs have larger magnitude of effort overruns as compared to the co-located PCs. The identified correlations between effort estimates and these factors (e.g. PC size, team maturity, customer priority and multi-site development) needs to be further investigated before drawing any conclusions.

These results have important implications for practice. Software practitioners, working in similar contexts, should consider the following aspects when estimating:

- As underestimation is the more dominant trend, it is important to consider factors (e.g., PC size, team maturity, etc.) that could potentially add delays in the project. The identification and consideration of these factors would be helpful in reducing over-optimism bias.
- It is important to involve all concerned stakeholders (e.g., system managers, project manager, architects, design lead) in the estimation process. Specifically, software architects have a pivotal role in the estimation process due to their understanding of the product’s architecture, which allows them to effectively perform an impact analysis of new requirements/tasks at the system level.
- Re-estimation at the analysis stage improves the effort estimates. At this stage estimators know more about the requirements and the assigned team(s). These refined estimates at the analysis stage support project managers in monitoring the progress on the task development.
- PC size was found to be strongly related to the effort estimates’ accuracy and bias. Large sized requirements/tasks are more likely to incur large effort overruns.
- Mature teams should be involved in the effort estimation process as they have architectural knowledge and expertise.
- Immature teams need more time to complete the tasks, and hence are more likely to incur effort overruns.
- Multi-site development, wherein geographically distributed teams collaboratively work on a task, exacerbates the underestimation bias.

As for research, the identified factors need to be further investigated in other contexts. The practitioners in the case use expert judgment to estimate the development effort. Expert judgment is the most widely used estimation technique [13]. We plan to investigate how the explicit consideration of relevant factors compliments expert judgment in arriving at better and informed effort estimates.

Acknowledgments

This research work is partially supported by the Swedish Knowledge Foundation under the grant 20120200 (2013–2016) and Ericsson Software Research.
References


A. BREbias QE regression model details and plots

Table A.1: BREbias QE regression analysis: ANOVA.

<table>
<thead>
<tr>
<th>Sum of Squares</th>
<th>df</th>
<th>Mean Square</th>
<th>F</th>
<th>Sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>93.21</td>
<td>2</td>
<td>46.60</td>
<td>20.98</td>
</tr>
<tr>
<td>Residual</td>
<td>126.61</td>
<td>57</td>
<td>2.22</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>219.82</td>
<td>59</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.2: BREbias QE regression coefficients' details.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-0.73</td>
<td>0.41</td>
<td>-1.78</td>
<td>0.08</td>
<td>0.996</td>
</tr>
<tr>
<td>Team Maturity</td>
<td>-0.56</td>
<td>0.20</td>
<td>-0.28</td>
<td>-2.78</td>
<td>0.007</td>
</tr>
<tr>
<td>PC Size</td>
<td>1.22</td>
<td>0.20</td>
<td>0.61</td>
<td>6.01</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table A.3: BREbias QE regression models’ summary.

<table>
<thead>
<tr>
<th>Variables</th>
<th>R²</th>
<th>Adjusted R²</th>
<th>PC Size</th>
<th>PC Priority</th>
<th>Team Maturity</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.42</td>
<td>0.39</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>0.42</td>
<td>0.40</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Figure A.1: Histogram of residuals.
Figure A.2: Normal PP Plot of regression standardized residuals.

Figure A.3: Scatter plot standardized residuals and predicted values.
Table B.1: BREbias AE regression: ANOVA.

<table>
<thead>
<tr>
<th></th>
<th>Sum of Squares</th>
<th>df</th>
<th>Mean Square</th>
<th>F</th>
<th>Sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>16.36</td>
<td>2</td>
<td>8.18</td>
<td>6.79</td>
<td>0.002</td>
</tr>
<tr>
<td>Residual</td>
<td>68.69</td>
<td>57</td>
<td>1.21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>85.05</td>
<td>59</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table B.2: BREbias AE regression coefficients’ details.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-0.07</td>
<td>0.25</td>
<td>-0.27</td>
<td>0.79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC Priority</td>
<td>-0.69</td>
<td>0.31</td>
<td>-0.29</td>
<td>-2.251</td>
<td>0.028</td>
<td>0.862</td>
</tr>
<tr>
<td>PC Size</td>
<td>0.57</td>
<td>0.16</td>
<td>0.45</td>
<td>3.54</td>
<td>0.001</td>
<td>0.862</td>
</tr>
</tbody>
</table>

Table B.3: BREbias AE regression models’ summary.

<table>
<thead>
<tr>
<th>Variables</th>
<th>R²</th>
<th>Adjusted R²</th>
<th>PC Size</th>
<th>PC Priority</th>
<th>Team Maturity</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.19</td>
<td>0.15</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>2</td>
<td>0.19</td>
<td>0.16</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

Figure B.1: Histogram of residuals.
Figure B.2: Normal PP Plot of regression standardized residuals.

Figure B.3: Scatter plot standardized residuals and predicted values.