State of the practice: An exploratory analysis of schedule estimation and software project success prediction

J.M. Verner a,*, W.M. Evanco b, N. Cerpa c

a National ICT Australia, Australian Technology Park, Alexandria, Sydney, NSW 1430, Australia
b College of Information Science and Technology, Drexel University Philadelphia, PA 19104, USA
c Department of Systems Engineering, Faculty of Engineering University of Talca, Talca, Chile

Received 15 November 2004; received in revised form 21 April 2006; accepted 17 May 2006
Available online 14 July 2006

Abstract

During discussions with a group of U.S. software developers we explored the effect of schedule estimation practices and their implications for software project success. Our objective is not only to explore the direct effects of cost and schedule estimation on the perceived success or failure of a software development project, but also to quantitatively examine a host of factors surrounding the estimation issue that may impinge on project outcomes. We later asked our initial group of practitioners to respond to a questionnaire that covered some important cost and schedule estimation topics. Then, in order to determine if the results are generalizable, two other groups from the US and Australia, completed the questionnaire. Based on these convenience samples, we conducted exploratory statistical analyses to identify determinants of project success and used logistic regression to predict project success for the entire sample, as well as for each of the groups separately. From the developer point of view, our overall results suggest that success is more likely if the project manager is involved in schedule negotiations, adequate requirements information is available when the estimates are made, initial effort estimates are good, take staff leave into account, and staff are not added late to meet an aggressive schedule. For these organizations we found that developer input to the estimates did not improve the chances of project success or improve the estimates. We then used the logistic regression results from each single group to predict project success for the other two remaining groups combined. The results show that there is a reasonable degree of generalizability among the different groups.

Keywords: Software effort estimation; Software schedule estimation; Software project staffing; Software project success

1. Introduction

A factor often suggested as a major contributor to software project failure is poor estimation of effort and schedule. Over a quarter century ago, Brooks [2] stated that more projects have gone awry for lack of calendar time than from all other causes combined. He pointed out that cost and schedule estimations are poorly developed and reflect an unstated assumption that all will go well. Kitchinham and Linkman [11] indicated that traditional software development cost estimates are usually based on the assumption that the project will encounter no exceptional problems and that managers do not use estimates correctly since they really do not understand risks and uncertainty. The problem apparently still exists as more recently Charette [4], included inaccurate estimates of resources in his list of the most common factors responsible for software project failure.

Brooks [2] also suggested that cost and effort estimates are uncertain and software managers do not stiffen their backbones and stubbornly support them. With poor estimation histories and the use of “seat-of-the-pants estimating techniques”, this outcome is not surprising. Moreover, the political agenda in many organizations may also be stacked against the project manager, making it difficult to convince senior management to accept realistic estimates.
Brooks [2] further pointed out that manpower added to a late project makes it later. McConnell suggested that this continued to be one of the software industry’s worst practices [14].

So it appears that today, problems with cost and schedule estimations persist in spite of on-going research into effort and schedule estimation. For example, Reel found that 26% of all software projects fail, and another 46% experience cost and schedule overruns or significantly reduced functionality [18], and Charette [4] describes a number of projects with unrealistic and overly optimistic schedules and describes estimation as much of an art as it is a science.

Software cost estimation is not just as simple as using some model or method to get good estimates. We not only need to use good estimation methods but we also require good requirements, coupled with knowledge of the organization’s performance, working practices, and software experience [3,12,13]. Kansala [10] suggests that cost estimation in isolation, is not enough; initial estimates need to be combined with quantitative risk assessment to yield realistic estimates, suggesting that knowledge of the project team, the technology to be used, and the state of the project requirements is essential.

In this exploratory study, we want to quantitatively explore how project estimates may influence the perceived success or failure of software development projects. Our objective is not only to explore the direct effects of cost and schedule estimation on the success or failure of a software development project, but also to quantitatively examine other factors surrounding estimation that may impinge on project outcomes. While much anecdotal evidence has been brought to bear upon these issues, there have been virtually no statistically grounded studies based on collections of data across organizations. Quantitative information of this kind is likely to be useful to organizations as they learn lessons from both their failures and successes. Given the pressure under which many project managers must function, better knowledge of the impact of both direct and indirect estimation factors will help them focus effort on project areas that are likely to be the most fruitful. This paper contributes to a better understanding of what actually happens in organizations that develop software and which estimation and scheduling factors are predictors of software success.

There are many different and overlapping definitions of project success, some narrowly related to scheduling, such as being “on time” [20] and comparing the “actual versus planned elapsed time for the entire project” [17]. Other definitions are broader, such as “successful projects are on time and within budget” [13] or giving “what the customer wants, when they want it, at the agreed upon price” [5]. However many authors focus on critical success factors for project development rather than on success outcomes [9,15,18]. In this study, we do not define success in advance, but instead allow the respondents to use their own implicit definitions.

2. Our investigation

We conducted structured discussions with 21 senior software practitioners who were employed by a large U.S. financial institution that developed software for its own use (Group 1). These discussions covered a number of important software development topics specifically focusing on a development project with which each respondent had recently been associated. Based on these discussions, and on an extensive review of the literature, we developed a questionnaire to address major software project success factors in seven broad categories, namely: (1) requirements, (2) management, (3) customers and users, (4) estimation, schedule, and staffing, (5) the project manager, (6) the software development process (including risk practices), and (7) development personnel.

In addition, we asked:

(a) did the respondents believe that management considered the project to be a success (MSUCCESS), and
(b) from their own point of view, did the respondents consider that the project was a success (RSUCCESS)?

When we refer to management’s perception of success, we are actually describing the developer’s perception of management’s view. Although this may appear a little strange, at the time this work was done we did not have access to a sufficient base of managers to obtain their views directly. In either case, we re-emphasize that the success measures are based on respondents’ implicit definitions of success [16].

Through a pilot study, we investigated the current state-of-the-practice of software development with practitioners from Group 1. These developers were based in a number of different U.S. locations and they reported on 42 different projects. We followed the pilot study with another two studies that questioned developers from a variety of organizations in order to investigate if the results from the first case study were generalizable. Our second study (Group 2), that targeted a group of software developers from a number of different business organizations (i.e., insurance, financial, pharmaceutical, and local utilities) in the United States involved 70 projects; nearly all these projects were developed for in-house use, i.e., not developed for a third party. The third study (Group 3), with 41 projects involved a group of software developers from Sydney, Australia, who were also mainly developing in-house software for commercial organizations (i.e., insurance, financial, telecommunications etc.). All three developer groups completed our questionnaire. Our sampling of practitioners was not random, but rather a convenience sample of developers to whom we had access; all respondents participated in various project management courses taught by the authors.

The substantial body of previous research that was explicated prior to any item development and our extensive review of the literature leads us to believe that we have identified relevant estimation/scheduling attributes that impinge on software development success and their
corresponding empirical measurement scales. Thus, we are confident that construct validity, concerned with whether or not the measurement scales represent the attributes being measured [1,7], is not a major problem in our study.

Given the large number of questions in our survey instrument, we might expect to see intricate relationships among the responses and, in some cases, high collinearity. Therefore, we chose to conduct an initial exploratory study focusing on cost/scheduling issues followed by a confirmatory study as a first step in “making sense” out of this data. In subsequent studies we intend to develop composite constructs based on the results of our current study and other studies.

We next describe our results (Section 3) and follow this with Section 4, which presents a discussion of our findings and conclusions.

3. Results

Each respondent answered either one or two questionnaires related to a recent project in which he/she had participated. The first group of developers (Group 1) focused on one project they considered to be a success and one that they regarded as a failure. Forty-five percent successful projects and 55% unsuccessful projects were reported. Groups 2 and 3 each completed a single questionnaire related to a project of their choice. We investigated the data for duplicate projects (a) through the organization involved followed by (b) a comparison of data fields, however, none were apparent. Some general results are shown in Table 1. Fourteen percent of the projects were maintenance or enhancement projects. In the rest of this paper, we refer to either data specific to a particular respondent group or to the combined data set from all of the respondent groups.

The projects described by our respondents ranged in size from very small one person projects to a large 180 person project. The size of projects described by our respondents was: <6 persons = 37%, 6–24 persons = 44%, 25–35 persons = 6%, and >35 persons = 13% (median 8). Overall, the respondents considered fewer projects to be successful than they perceived management did. (χ² tests comparing MSUCCESS and RSUCCESS were significant at 0.006.) Twenty-six projects considered successful by management were considered failures by developers. Conversely, only one project considered a failure by management was considered successful by the developer. Groups 2 and 3 describe more successful projects. It appears that when given a choice of what kind of project to describe most respondents in these groups preferred to discuss a successful one or, alternatively, there were just more successful projects to choose from. However, project success rates reported in [21–23,8] would suggest that our respondents preferred to focus on successful projects. There is no significant difference between the two success variables within Group 2 and Group 3 projects. There was also no significant relationship between size of project and success from either the management or developer perspective.

Questions relating to estimation and scheduling factors and the responses are shown in detail in Tables 2–4. Table 2 shows the percentages of “yes” answers to the questions asked for the three groups as well as for all of the groups combined (the “All” column represents the percentages for all three groups combined). In one case, for reasons of clarity and space, we have consolidated the answers originally answered on a five point Likert scale to above average and average/below average¹. For reader clarity, we have slightly reworded the original question to provide a label that fits this revision. Table 3, shows correlations between the variables studied.

The rest of our results, are organized as follows: Section 3.1 provides details of responses to the questions in Table 2. Section 3.2 describes managements’ and respondents’ views of success, and Section 3.3 focuses on predicting project success and failure.

3.1. Detailed responses

The exploratory analyses conducted in this section are correlational in nature because of the non-parametric

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Percentage of successful projects</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Projects</td>
</tr>
<tr>
<td>Group 1</td>
<td>42</td>
</tr>
<tr>
<td>Group 2</td>
<td>70</td>
</tr>
<tr>
<td>Group 3</td>
<td>41</td>
</tr>
<tr>
<td>Overall</td>
<td>153</td>
</tr>
</tbody>
</table>

1 In such cases, the “yes” or “no” answers might be more aptly be interpreted as “above average” and “average/below average”. For purposes of conciseness, we used the “yes” or “no” designation.
Table 3
Tests for variable relationships: Spearman measure of association and significance levels <0.05

<table>
<thead>
<tr>
<th></th>
<th>Project had a schedule</th>
<th>Project manager had input into initial delivery date</th>
<th>PM could negotiate schedule</th>
<th>Schedule changed through the project</th>
<th>Estimation done with appropriate reqts info</th>
<th>Developers had input into the estimates</th>
<th>Staff leave was taken into account</th>
<th>Estimates were good</th>
<th>Project was underestimated</th>
<th>Delivery date affected the process</th>
<th>Adequate staff assigned to the project</th>
<th>Staff added late to meet aggressive schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project had a Schedule</td>
<td>-0.007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project manager had input into initial delivery date</td>
<td>0.929</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PM could negotiate schedule</td>
<td>-0.027</td>
<td>0.0511</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schedule changed through the project</td>
<td>0.778</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimation done with appropriate reqts info</td>
<td>0.026</td>
<td>0.216</td>
<td>-0.029</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developers had input into the estimates</td>
<td>0.748</td>
<td>0.010</td>
<td>0.762</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Staff leave was taken into account</td>
<td>0.032</td>
<td>0.240</td>
<td>0.420</td>
<td>-0.140</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimates were good</td>
<td>0.693</td>
<td>0.004</td>
<td>0.000</td>
<td>0.084</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project was underestimated</td>
<td>-0.030</td>
<td>0.000</td>
<td>-0.137</td>
<td>0.027</td>
<td>-0.276</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delivery date affected</td>
<td>0.728</td>
<td>0.998</td>
<td>0.107</td>
<td>0.757</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adequate staff assigned to the project</td>
<td>0.125</td>
<td>0.148</td>
<td>-0.185</td>
<td>0.012</td>
<td>0.114</td>
<td>0.112</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Staff added late to meet aggressive schedule</td>
<td>0.123</td>
<td>0.081</td>
<td>0.052</td>
<td>0.879</td>
<td>0.162</td>
<td>0.198</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adequate staff assigned</td>
<td>0.124</td>
<td>-0.052</td>
<td>-0.030</td>
<td>-0.220</td>
<td>0.433</td>
<td>-0.220</td>
<td>0.201</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adequate staff assigned</td>
<td>0.122</td>
<td>0.534</td>
<td>0.757</td>
<td>0.006</td>
<td>0.000</td>
<td>0.006</td>
<td>0.013</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adequate staff assigned</td>
<td>0.022</td>
<td>-0.029</td>
<td>0.088</td>
<td>-0.092</td>
<td>-0.036</td>
<td>0.282</td>
<td>-0.114</td>
<td>0.072</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adequate staff assigned</td>
<td>0.784</td>
<td>0.732</td>
<td>0.355</td>
<td>0.254</td>
<td>0.661</td>
<td>0.001</td>
<td>0.161</td>
<td>0.372</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adequate staff assigned</td>
<td>0.265</td>
<td>-0.076</td>
<td>-0.160</td>
<td>0.028</td>
<td>-0.189</td>
<td>0.201</td>
<td>-0.073</td>
<td>-0.142</td>
<td>-0.076</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adequate staff assigned</td>
<td>0.001</td>
<td>0.363</td>
<td>0.059</td>
<td>0.729</td>
<td>0.019</td>
<td>0.011</td>
<td>0.371</td>
<td>0.077</td>
<td>0.342</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adequate staff assigned</td>
<td>0.016</td>
<td>-0.099</td>
<td>-0.076</td>
<td>-0.199</td>
<td>0.368</td>
<td>-0.293</td>
<td>0.208</td>
<td>0.415</td>
<td>-0.013</td>
<td>-0.309</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adequate staff assigned</td>
<td>0.837</td>
<td>0.237</td>
<td>0.426</td>
<td>0.013</td>
<td>0.000</td>
<td>0.000</td>
<td>0.010</td>
<td>0.000</td>
<td>0.871</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adequate staff assigned</td>
<td>0.116</td>
<td>0.005</td>
<td>0.122</td>
<td>0.241</td>
<td>-0.220</td>
<td>0.078</td>
<td>-0.134</td>
<td>-0.234</td>
<td>-0.038</td>
<td>0.419</td>
<td>-0.295</td>
<td></td>
</tr>
<tr>
<td>Adequate staff assigned</td>
<td>0.148</td>
<td>0.950</td>
<td>0.200</td>
<td>0.002</td>
<td>0.006</td>
<td>0.326</td>
<td>0.098</td>
<td>0.003</td>
<td>0.633</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>
nature of our data. We employed $\chi^2$ tests to determine the significance of correlations among our data. A number of the schedule/effort estimation attributes have statistically significant relationships with software development success as well as amongst themselves. With regard to the internal validity of this study, these relationships do not prove causality, but only provide empirical evidence for it. Only controlled experiments could identify a causal relationship and such experiments are not practical in this particular context.

In the ideal, once requirements are complete, the project manager, possibly in consultation with one or more of the other stakeholders\(^2\), develops estimates of effort and schedule that include an allowance for identified risks. If the stakeholders do not accept the project manager’s estimate, then negotiation would occur, resulting in either reduced functionality or the assignment of more talented or experienced team members. We anticipated that such negotiation would also occur in those cases where a budget had been fixed earlier. Of course estimates of effort and schedule lead to assigned project staffing which can have an effect on schedule, software development practices and project success and failure. We will discuss in the following subsections, the extent to which this ideal is practiced in today’s software development environments.

Section 3.1 is divided into three sub-sections: (1) estimation, (2) schedule, and (3) staffing. Each sub-section includes a summary of the results, then cycles through the questions asked, a discussion of the responses to that question, and includes where appropriate:

- The frequency of responses by group and for the combined data set (Table 2), including significant differences between groups (i.e., $<0.05$ using a Kruskal–Wallis, or Mann–Whitney test as appropriate).
- Significant relationships the responses to the question have with MSUCCESS and RSUCCESS (Table 4) using a $\chi^2$ test.
- Important relationships the responses to the question have with responses to other questions using $\chi^2$ tests (Table 3).

In the rest of the paper, if a $\chi^2$ test between variables is significant, we refer to this as a significant relationship and only if the correlation between the variables is negative do we comment on the direction of the relationship.

### 3.1.1. Estimation

Summarizing the results of our analyses, our data suggests that:

- If the project manager had input into the initial delivery date, further schedule negotiation is likely to occur, and estimation will be done with appropriate requirements
information. In such cases, these estimates are likely to be good and the schedule is unlikely to affect the development process. In addition, adequate staff will be assigned to the project and staff will not be added late. Projects with good estimates are likely to be perceived as successful by developers. For projects with good estimates, developers tend not to have input into the schedule, the project will not have schedule changes.

- On the other hand, projects estimated without adequate requirements information will probably not have had project manager involvement in the initial estimates, nor will the PM be able to negotiate the schedule. These projects will not have good estimates, will likely have developer input into the estimates, inadequate staff will be assigned, the delivery date will affect the development process and staff will be added late to meet an aggressive schedule.

These results support the common wisdom that schedule estimates made with appropriate requirements information are likely to be better, that poor project estimates result in too few staff with further staff added late. When the estimates did not involve the project manager then appropriate requirements information was less likely to be used. However, interestingly, the developers were asked for input into estimates when appropriate requirements information was not available.

Detailed analyses of the specific questions regarding estimation procedures are given below:

1. Estimation was done with appropriate requirements information? \([\text{Yes} = 1, \text{No} = 0]\)

- Overall 52% of projects were estimated with appropriate requirements information. This result violates the ideal situation that effort/schedule estimates be made on the basis of full requirements information.
- Only 30% of Group 1’s projects had appropriate requirements information when their estimates were made; for Group 2 the figure was 64% and for Group 3, 50%. Group 1 is the exception here and is significantly different from Group 2 (0.000) and Group 3 (0.067) though we must note that Group 1 had, by our own design, more unsuccessful projects than either of the other two groups; this result illustrates the importance of appropriate requirements information in project outcomes (see Table 2).
- MSUCCESS was significantly related to responses to this question for Group 2 only. RSUCCESS was significantly related to responses to this question for both U.S. Groups (1 & 2) but not for Group 3 (see Table 4).
- Responses to this question were significantly related to the project manager had input into the initial delivery date, PM could negotiate the delivery date, adequate staff were assigned to the project, and the estimates were good. This variable was significantly related and negatively correlated with developers had input into estimates, the delivery date affected the development process, and staff were added late to meet an aggressive schedule (see Table 3).

2. Developers had input to the estimates? \([\text{Yes} = 1, \text{No} = 0]\)

- In 49% of projects developers had input into the schedule.
- For this variable, Group 2 is significantly different from the other two groups (0.000). Sixty-three percent of the Group 1 projects had developer input to estimates, while Group 2 had 27%, and Group 3, 67% (Table 2).
- This variable significantly negatively related to RSUCCESS and MSUCCESS for Group 1 data (0.000). For Group 1, 78% of those projects where the developers had input to the estimates were considered a failure by developers, suggesting that developer involvement in making project estimates is detrimental to project success. Developers may make estimates without adequate estimation experience or, given poor requirements for projects, (only 30% of Group 1 projects had adequate requirements information), they may try to estimate something mainly unknown, or developers may only asked for their opinion when the requirements are poor. Developers may also tend to be optimistic and too often underestimate the work necessary to complete at project. Group 2 projects were not badly affected by developer input to the estimates. Group 3 projects were somewhere in the middle, with 40% of projects with developer estimate inputs perceived as unsuccessful.
- This variable was significantly related and negatively correlated with estimation done with appropriate requirements, the estimates were good, and adequate staff were assigned to the project. It was significantly related to the delivery date affected the development process and the project was underestimated.

In summary, for these projects the most likely scenario is that software developers are involved in estimates if the project does not have appropriate requirements information, underestimates are likely, inadequate staff are then assigned to the project and the delivery date affects the development process.

3. The estimates of effort and schedule were good? \([\text{Above average} = 1, \text{average and below} = 0]\)

- For all projects 38% of effort and schedule estimates were regarded as good (above average).
- Group 1’s results are significantly different from the other two groups for this variable but they had more failed projects. The respondents in Group 1 thought that

---

3 By developers we exclude the project manager.
17% of their project estimates were above average, while Group 2 respondents indicated 44%, and Group 3, 45% were above average.

- The responses are not significantly related to MSUCCESS for any of the groups on their own but are significantly related for the combined data set. This variable is significantly related to RSUCCESS for Groups 1 and 2 and for the combined data set, which suggests that good estimates are instrumental in perceived project success from the developers’ point of view. Project success or failure is often viewed in terms of keeping to costs and meeting a schedule. Poor estimates will lead to deviations of results from the estimates, making it difficult to justify a project as being successful.

- The responses to this question are significantly related to staff leave was taken into account, estimates were made with appropriate requirements information, and adequate staff were assigned to the project. They are significantly related and negatively correlated with schedule changed through the project, developers had input to estimates, and staff added late to meet an aggressive schedule.

In summary, the most likely scenario for these projects is that good estimates are made with appropriate requirements information and adequate staff are then assigned to the project. Such projects will not have schedule changes, developers will not have input into their schedules, and staff will not be added late to meet an aggressive schedule.

(4) The project was underestimated? [Yes = 1, No = 0]

- Not surprisingly, the majority of projects were underestimated (69%).
- Seventy percent of projects in Group 1, 76% of Group 2, and 55% of Group 3 were underestimated. Group 2 is different from Group 3 at a significant level (.05) but there are no other significant differences between the groups. Three-quarters of the projects that the respondents indicated had “estimates of average quality” were underestimated; this suggests that developers were so accustomed to underestimates that they did not consider this situation to be unusual. Even among those projects that supposedly had above average estimates worse, 66% of were underestimated. Group 1 did not do as badly as Group 2 as only 57% of their “good” estimates were underestimate; for Group 2, 89% of the “good” estimates were underestimates. For Group 3 only 32% of the projects with supposedly “good” estimates were underestimated. The only overestimated projects in the entire data set were from Group 3, where two projects were overestimated, suggesting that it is still true that most of the time we are far too optimistic and assume that things will go well.
- This variable was significantly negatively related to RSUCCESS and MSUCCESS for Group 3 but not for the other two groups.
- The responses to this question were significantly related to developers had input to the estimates.

In summary, most of these organizations’ projects were underestimated though overall this did not make any difference to their perceived success. Developers were likely to have input into the underestimated projects.

Given the number of project failures cited in research, such as that done by the Standish group, we were not surprised that 69% of projects were underestimated. Because on time and within budget are two frequently mentioned criteria for project success we were surprised that this did not make any difference to the perceived success of the projects.

(5) The delivery date affected the development process? [Yes = 1, No = 0]

- Unexpectedly, the delivery date affected the development process in a colossal 84% of projects.
- Responses to this question were significantly related and negatively correlated with RSUCCESS for the combined data set. Eighteen percent of projects where the estimated delivery date affected the development process were viewed as failures by management, while developers perceived that 36% of these same projects were failures. This result is to be expected as developers have better insights into the development process than management and are likely to be more concerned with shortchanged activities that management may be unaware of, e.g., testing adequacy, and overall quality etc.
- The responses were significantly related to the project had a schedule, developers had input to the estimates, and staff added late to meet an aggressive schedule. Finally, the responses were also significantly related and negatively correlated with estimates were made with appropriate requirements information, and adequate staff were assigned to the project.

In summary, the most likely scenario is that, if the delivery date affected the development process then estimates were made without appropriate requirements information, inadequate staff were assigned to the project, developers probably had input into the estimates and staff were added late to meet an aggressive schedule.

3.1.2. Scheduling

Summarizing the results of our analyses for scheduling, the data suggests:

- Most projects will have a schedule, although the majority of project managers do not have input into the initial delivery date.
- However, if the project manager is involved in estimating the initial delivery date or in negotiating the schedule at some later time, then the estimates are likely to have been made with appropriate requirements information, and the developers will perceive the project as being successful.
• Poor project schedule estimates are associated with inadequate requirements information. Most poorly estimated projects will have schedule changes, will have inadequate numbers of staff assigned, in which case staff will be added late to meet an aggressive (poor) schedule. Consequently, the development process will be affected by the delivery date.

The observations made above are supported by the detailed analyses below:

(1) The project has a schedule? [Yes = 1, No = 0]

By schedule, we mean more than just delivery date, as scheduling involves the time phasing of activities throughout the project.

• Eighty-seven percent of the projects had a schedule. The Australian group of projects is significantly different from the U.S. projects: 98% had a schedule versus 78% and 86% for the two U.S. groups.
• No significant relationships were established with the two success variables for responses to this question.
• Responses to the question, “Did the project have a schedule?” were significantly related (0.000) with responses to the question “Did the schedule affect the development process?”. Eighty-nine percent of the projects with a schedule had the development process impacted by the schedule, compared with 52% of projects without a schedule. Although these latter projects may have begun without an explicit schedule over half of them acquired an implicit schedule at some stage in the development process.

These results would seem at first peculiar – that having a schedule is not correlated to perceived success outcomes. But if in most cases, the actual schedule estimates are poor, then these results would not be surprising – no schedule might be better than a poor schedule; if there is no schedule to overrun, then the project may be less likely to be viewed as a failure. However, the likely scenario for these projects is that most will have a schedule with a delivery date at some point and the schedule is likely to affect the development process.

(2) The project manager had input into the initial delivery date? [Yes = 1, No = 0]

Our initial assumption was that the project manager would be involved in deciding the initial delivery date as he/she is likely to have better knowledge than anyone else of the organization’s development productivity, technology, people, and development practices to be used for the project. However, this assumption was incorrect for the majority of the projects.

• For only a minority of projects (40%) was the project manager involved in deciding the initial delivery date (Table 2). In the other cases higher-level management, marketing, or the customer/user decided when the project was to be delivered. There were no significant differences among the groups for this variable.
• Three projects did not have a project manager and two of these were viewed as unsuccessful by both management and developers. There were no significant relationships between responses to the question “Did the project manager have input into the initial delivery date?” and either of the success variables (Table 4). If the project manager did not have input into the initial delivery date, developers were more than twice as likely as management to consider the project a failure. For the projects where the project manager did not have input into the initial delivery date, 40% were considered to be failures by developers, while they thought that only 24% of those projects in which the project manager did have input were failures.
• This variable is significantly related to the schedule changed during the project.

In summary, the likely scenario for these projects is that most project managers do not get to provide input into the initial delivery date, however, when the project manager is involved then the estimate is based on appropriate requirements information though the schedule is likely to be changed during the project.

We were very surprised at these results as we had expected that most estimates would have project manager input. Even after 25 years of advancement with cost and schedule estimation, decisions regarding these matters seem to be made with minimum technical input by non-technical stakeholders. We were even more surprised that this did not apparently affect the perceived success of the project.

(3) If the project manager did not have input into the initial delivery date, could he/she negotiate the schedule? [Yes = 1, No = 0]

• For 36% of projects, project managers not only had no input into the initial delivery date but could not subsequently negotiate the schedule.
• For projects without any kind of project manager input to the delivery date, 81% were perceived as successful by management and 56% by the developers. Negotiation of the schedule by the project manager is significantly related to the developers’ view of project success.
• Responses to this question about the project manager’s ability to negotiate a schedule, were significantly related to responses to the question as to whether the estimation was done with appropriate requirements information, i.e., project managers want good requirements when they make or revise an estimate. Common sense tells us that without appropriate requirements the ability of the project manager to negotiate a schedule will be affected.

In summary, the most likely scenario for these projects is that if the project manager can negotiate the schedule then
project estimates will have been made with appropriate requirements information, and developers will perceive the project as successful. None of these results is surprising. We had expected that project manager involvement in negotiating the schedule, due to his/her knowledge of the technology and people involved, would have a positive impact on project outcomes.

(4) The schedule was changed through the project [Yes = 1, No = 0]

- Overall, 69% of our projects had schedule changes during the lifecycle. Schedule changes are not necessarily a bad thing. For example, if the requirements are initially poor, we would expect schedule changes during development.
- The combined responses to this question were not significantly related to MSUCCESS, although there was a negative relationship with RSUCCESS. When our respondents were asked about the causes for schedule changes, they mentioned requirements and scope changes, low priority given to the project, poor initial estimates, and lack of risk management.
- This variable was significantly related to staff added late to meet an aggressive schedule. A significant relationship between the schedule was changed through the project and the project manager had input into the initial delivery date is a little strange and difficult to explain. Reasons for the schedule changes needs further investigation. Seventy-four percent of the poorly estimated projects had schedule changes and schedule changes was significantly related and negatively correlated with the estimates were good. This result agrees with Rainer and Shepperd [17] who noted that re-plans were mainly in response to poor original estimates. Although the questions we are concerned with here address estimation, scheduling and staffing, we asked a number of questions about requirements in an earlier section of the questionnaire. The adequacy of requirements is important when we consider schedule changes throughout the project. Schedule changes was also significantly related, and negatively correlated, with adequate numbers of staff with 80% of projects with schedule changes having inadequate numbers of staff; 61% of those projects staffed adequately had schedule changes.

In summary, the most likely scenario for these projects is that poor project schedule estimates are based on inadequate requirements information and such projects will have schedule changes. These projects will probably have inadequate numbers of staff assigned to them and staff added late to meet the aggressive (poor) schedule. These results are not surprising since it is common wisdom that schedule changes are made in response to poor estimates of effort and schedule with such estimates probably due to poor initial requirements.

3.1.3. Staffing

Summarizing the results of our analyses for staffing, the data suggests that:

- The assignment of adequate staff to a project is associated with good estimates of effort/schedule. Adequately staffed projects account for staff leave and tend to have the PM involved in the initial delivery date decision.
- Inadequate staffing is associated with schedule changes, adding staff late to meet aggressive schedules, and with developer involvement in the schedule/effort estimates.

The observations made above are supported by the following specific empirical observations:

(1) Adequate staff were assigned to the project? [Yes = 1, No = 0]

- Overall, 64% of the projects had adequate staff assigned.
- For this variable Group 1 had 53% of projects staffed adequately, which was significantly different from Group 2 with 71% of its projects staffed adequately. Group 3 had 62% of projects adequately staffed and was not significantly different from the other groups.
- The adequate staff variable was significantly related to MSUCCESS and with RSUCCESS for both the combined data set and for Group 2.
- The relationship of effort/schedule estimation to staffing is shown quite clearly in Table 3. This variable is significantly related to the estimates were good, staff leave was taken into account, estimation was done with appropriate requirements information, and was negatively correlated with schedule changed through the project, delivery date affected the process, developers had input to the estimates, and staff added late to meet an aggressive schedule. Interestingly, only 47% of projects that had developer involvement in the estimates had adequate staff, while 76% of the projects that did not have developer involvement in estimates had adequate staff. It would appear that developers involved in “seat of the pants” estimates without adequate requirements information, may lead to inappropriate staffing.

In summary, if a project has adequate staff assigned to it, staff leave will have been taken into account in the estimates, estimates will have been done with appropriate requirements information, the estimates will be good, staff will not be added late to meet an aggressive schedule, the schedule will not be changed, the delivery date will not affect the development process and developers will not have had input into the estimates. If inadequate staff were assigned to a project, estimates were below average, the project will have schedule changes, and developers will probably have been involved with the estimates.

(2) Staff were added late to meet an aggressive schedule? [Yes = 1, No = 0]
In 33% of the projects, staff were added late in order to meet an aggressive schedule.

The results were very similar for each of the three groups (36%, 30%, and 33%).

This variable was significantly related and negatively correlated with both MSUCCESS and RSUCCESS. Overall, only 42% of projects that had staff added late to meet an aggressive schedule were considered a success by the developers. However, this factor did not appear to make much difference to management’s view of success; they considered 76% of these projects a success compared to the 83% of all projects that they considered to be a success.

This variable was significantly related to the schedule was changed through the project; and the delivery date affected the development process. It also has a number of negative correlations with other variables such as the estimates were good, estimation was done with appropriate requirements information, and adequate staff was assigned to the project.

Overall 47% of project estimates took staff leave into account.

Groups 2 and 3 were significantly different on the basis of this variable.

This variable was significantly related to RSUCCESS for the combined data set and for Group 2, but not significantly related to MSUCCESS overall. We found that 66% of projects where staff leave was not taken into account were considered failures by developers.

This variable was significantly related to estimates were good and adequate staff were assigned to the project. For those projects that had good estimates, 60% took staff leave into account and, of those that did not have adequate staff assigned to the project, 68% did not take staff leave into account.

The most likely scenario for these projects is that if staff leave is taken into account when the estimates are made the estimates will be above average, and adequate staff will be assigned to the project. This result is not surprising as commonsense tells us that one source of project underestimates might be not taking staff leave into account.

3.2. Management’s and developer’s views of success

Our developers commented that the projects they regarded as failures were unpleasant experiences, mainly because of the excessive overtime required. Given the lack of post mortem reviews (only 33% overall), and the fact that their overtime is normally unpaid, this extra effort is probably not highly visible to higher level management. Our results in Table 4 suggest that a late project does not usually matter nearly as much to management as it does to the developers, agreeing with observations by Collier et al. [6]. If we consider the successful versus the unsuccessful projects, from the management point of view (MSUCCESS), four factors are significant in our $\chi^2$ tests for the combined data set (see Table 5):

- Estimation was done with appropriate requirements information (positive correlation).
- The estimates were good (positive correlation).
- Adequate staff were assigned to the project (positive correlation).
- Staff were added late to meet aggressive schedule (negative correlation).

However, if we look at the developers’ views of the project success (RSUCCESS), then additional variables come into play giving us twice as many variables that are correlated with developers’ views of project success in addition to the ones cited above:

- Project manager could negotiate the schedule (positive correlation).
- Staff leave was taken into account when making estimates (positive correlation).
- Schedule was changed through the project (negative correlation).
- The delivery date affected the development process (negative correlation).

The developers obviously have a broader range of concerns when evaluating the success of a project.

3.3. Predicting success

This section provides us with a perspective on the prediction of project success and the generalizability of our results across groups. Since the success variable is coded as zero for “unsuccessful” and one for “successful,” we were not able to use ordinary least squares regression, which relies on a continuous dependent variable. Instead, we conducted a logistics analysis.

### Table 5

<table>
<thead>
<tr>
<th>Variable</th>
<th>Parameter estimate</th>
<th>Standard error</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimation was done with appropriate requirements info</td>
<td>1.73</td>
<td>.47</td>
<td>.0002</td>
</tr>
<tr>
<td>Estimates were good</td>
<td>1.00</td>
<td>.52</td>
<td>.05</td>
</tr>
<tr>
<td>Estimates took staff leave into account</td>
<td>.82</td>
<td>.44</td>
<td>.05</td>
</tr>
<tr>
<td>Staff added late to meet aggressive schedule</td>
<td>-1.30</td>
<td>.44</td>
<td>.003</td>
</tr>
<tr>
<td>Constant</td>
<td>- .24</td>
<td>.38</td>
<td>.53</td>
</tr>
</tbody>
</table>

This analysis provides insight into the factors that influence project success, helping to inform management and developers in future projects.
We found only a few explanatory variables for MSUCCESS, and, because this data was obtained second hand, we will not consider this variable further at this stage. In this section, we develop models for project success with RSUCCESS as the dependent variable. We conducted a multivariate analysis to explore the impacts that the variables discussed in Sections 3.1 have on perceptions of success from the developers’ perspective. Based on these variables, we calibrated models to estimate success for each of the groups and then used the calibrated model to predict the success or failure of projects in the other groups. Although, as discussed above, a total of eight variables were correlated with RSUCCESS, we could not reasonably expect all eight to enter into our regression analysis. First, we have a limited number of observations which constricts the number of explanatory variables that can be reasonably supported. Second, the eight variables exhibit multicollinearity among themselves as shown in Table 3, which additionally restricts the explanatory variables.

The parameter estimates for project success from the developers’ perspective for the total sample are shown in Table 5 and for each of the groups in Tables 6–8. The parameter estimates associated with the variables are all significant to at least the 5% level, and the parameters enter with the appropriate signs. The probabilities are calculated from:

\[ \text{Prob(Success)} = \frac{e^Z}{1 + e^Z} \]  

(1)

where \( Z = a_0 + \sum_{i=1}^{n} a_i \times X_i \), the \( a_i \) are the parameters from the table below, and the \( X_i \) are the associated variables.

The logistics model for the combined data set has four significant explanatory variables as shown in Table 5. The logistics models for Groups 1 and 2 have two significant explanatory variables, while Group 3 has one explanatory variable as shown in Tables 6–8, respectively. The combined data exhibits greater variation among the variables and, hence, supports a richer model.

We used the results from Tables 6–8 to estimate the successes and failures for each of the groups and compared the estimated results with the actual results. These comparisons are shown in Tables 9–11 for Groups 1, 2, and 3, respectively. We used equation (1) to compute the probability of success and for Prob(Success) \( \geq 0.65 \) we assumed that the project was a success, otherwise it was a failure. We initially took the cutoff to be .5 but found that we had poorer estimates of failures with this cutoff level.

If we use the prediction equations shown in Tables 6–8 to predict the project outcomes for the group on which the equation was developed we get: for Group 1, we correctly estimated the success or failure for 32 out of 40 projects (80%), for Group 2, 47 out of 63 projects (75%), and for Group 3, 26 out of 41 projects (63%). The Group 3 estimation yields results that are inferior to those of Groups 1 and 2. This result is most likely due to the fact that only a
single variable enters into the estimation of the logistics equation for Group 3.

Finally, on the basis of the logistic equation results in Tables 6–8, we used the probability equation computed for one group to predict the successes and failures for the other two groups combined. Once again, the cutoff to determine success or failure was taken to be 0.65. The results are shown in Tables 12–14. Using the Group 1 logistic equation to predict for Groups 2 and 3 yielded 71 correct predictions out of 104 projects (68%). Using the Group 2 equation yielded 56 correct predictions out of 81 projects (69%), while the Group 3 equation gave 44 correct predictions out of 112 projects (39%). Once again the single significant explanatory variable for the Group 3 estimation yielded significantly poorer results when the Group 3 model was used for prediction purposes.

From the results above, Group 3 appears to be an outlier group. From Table 2, we see that substantially more Group 3 projects had a schedule (98%) than Groups 1 and 2. More Group 3 projects had delivery dates with project manager input and these estimates took account of staff leave. Group 3 had fewer projects that were underestimated and the development process was less influenced by delivery date. What we may be seeing here are the impacts of differences in both cultural and technical skill levels between the projects developed in the U.S. and those developed in Australia. The software development process in these Australian organizations appears to be more orderly as evidenced by the fact that virtually all projects had schedules, the estimates were better, and the development process was not compromised because of delivery date. The urgency with which so many projects in our U.S. organizations are developed militates against an orderly development process. In addition, project manager involvement in deciding the initial delivery date may be an indication that Australian project managers have a better grasp of estimation techniques and, hence, may be more credible. For some period of time Sydney had an active measurement group, the Australian Software Metrics Association (ASMA), who had monthly meetings. The presence of this group may have contributed to a greater awareness of estimation and scheduling practices. Under such conditions, we might expect that a greater fraction of projects would be regarded as successes for Group 3.

External validity is concerned with the generalizability of the results to other environments than the one in which the initial study has been conducted [1,19]. The results above show that there is a reasonable degree of generalizability among the different groups. We also noted, however, that due to cultural considerations, Group 3 in some ways was substantially different from the other groups. For example, Group 3 had many fewer failed projects on which to base our analysis. Further Australian data may help in this regard.

4. Discussion and conclusions

Almost half the projects in our samples were estimated with unclear requirements. Inadequate requirements severely handicap the developers’ ability to apply estimation techniques and methodologies that might provide reasonable cost and schedule estimates. The most surprising result of this study is that project manager involvement in the initial effort and schedule estimates was not significantly correlated with project success from both the management and developer points of view. From the developer point of view, our results suggest that success is more likely if the project manager is involved in schedule negotiations. These results imply that the project managers involved in our sample of projects will have found it difficult to follow Brook’s [2] advice to “stiffen their backbones and stubbornly support” their estimates.

Similarly, developer input to the estimates did not improve the chances of success or improve the estimates. In fact, in some cases, the inclusion of developer input made it more likely that the project would result in failure. In many cases, developers do not have a global perspective of the project and, hence, may be handicapped in producing project-wide estimates. In addition, past histories of poor estimates may be affecting the ability of both project managers and developers to participate in the estimation process. Perhaps if project managers were better educated in estimation techniques and methodologies, they could improve their effort and schedule estimation capability.
and credibility. The software research community is perhaps not getting its message across regarding estimation modeling and methods that are now available to the appropriate software development decision-makers. Optimism still prevails — 69% of our projects were underestimated, and 65% of successful projects were underestimated — reiterating that it is still true that we are optimistic.

Our study supports Brooks [2] assertion that “cost and schedule estimations are poorly developed and reflect an unstated assumption that all will go well”, suggesting a lack of risk assessment. If adequate risk assessments had been done on these projects one would expect that the project manager/risk officer would have identified the poor state of the project’s requirements and have done something to improve them. Since schedule/cost underestimates are significantly correlated with the state of requirements and staffing levels, a lack of risk assessment, in turn, may result in staffing itself being a major risk factor that affects the development process. Although, inadequate staffing was related to developers’ perception of success it was not related to management’s perceptions of success. Our developers indicated that management seemed to have little concern regarding project underestimation and the unpaid overtime that tended to be mandatory. Adding staff late to meet an aggressive schedule is still a problem and is significantly correlated with both management’s and developers’ views of project failure.

Where should project managers initially look to improve their chances of project success? While we cannot show causality, our research suggests that many factors impinge on project success and failure, but that the most important of these are:

- adequacy of the requirements information when estimates are made,
- goodness of the initial effort estimates,
- taking staff leave into account, and
- effect of adding staff late to meet an aggressive schedule.

From the management point of view, we are unable to predict the success or failure of a project based on the sample of projects in our study. This result may have to do with the fact that the “management point of view” is really the developers’ perception of management’s point of view. Future studies will include data that directly reflects the point of view of management.

It is clear from the above discussion that risk and the lack of management awareness of risks have an important impact on project outcomes. Our thesis is that there will be fewer software development failures if project managers improve their understanding of project success determinants at a conceptual level.

Acknowledgment

This research has been supported by the Chilean Grant FONDECYT 1030785.

References