What requirements engineering practices are useful for predicting software project success?

J. M. Verner  
Empirical Software Engineering  
National ICT Australia  
Australian Technology Park  
Sydney, Australia  
june.verner@nicta.com.au

K.A. Cox  
Empirical Software Engineering  
National ICT Australia  
Australian Technology Park  
Sydney Australia  
karl.cox@nicta.com.au

S. J. Bleistein  
CSE, University of New South Wales & ESE  
National ICT Australia  
Sydney Australia  
steven.bleistein@nicta.com.au

N. Cerpa  
Departamento de Ingeniería de Sistemas, Universidad de Talca, Camino Los Niches Km. 1, Curicó, Chile  
ncerpa@utalca.cl

Abstract

During discussions with a group of U.S. software developers we explored the effect of software development practices on software project outcomes. Later we asked our practitioners to respond to a questionnaire that covered many important project risk topics. Then, in order to compare practices in different environments, another group of practitioners from the U.S. and a group of practitioners from Australia completed our questionnaire. Most software engineering research stresses the importance of good requirements engineering practices and suggests that they are essential to successful projects. We report on requirements engineering risk practices, including 1) the effects of the sponsor and customer/users, 2) requirements issues, and 3) the project manager and management of the development process, in order to identify early lifecycle factors useful in predicting project outcomes. If project managers are able to identify problematic projects early enough, they can take mitigating action. Our results suggest that the best predictor of project success is that the requirements are completed at some stage during the development process.

Key Words: software project risk, software project success, requirements engineering, software development practices

1. Introduction

Software projects are notoriously difficult to manage and too many of them end in failure [13]. A report from the U.K. stated that only 16% of software projects could be considered truly successful [9] while another report suggests software development projects success rate has risen to a “modest” 28%. [24]. Over the past few years, in Australia alone, there have been major project failures resulting in loss of millions of dollars [16, 1]. One explanation for the high failure rate is that managers are not taking prudent measures to assess and manage the risks involved in software projects [13]. But, before we can develop meaningful risk management strategies we must identify project risks and determine their relative importance [13]. Many organizations, however, have not learned very much about good software development since Brooks [3] wrote his landmark book in 1975 [25]. Research suggests that many project failures are the result of poorly defined and incomplete requirements. For example, the Standish reports on the state of software engineering practice indicate that requirements engineering (RE) is critical to software success [23]. This report also suggests that three of the top ten reasons for challenged projects or outright project failure are lack of
user involvement, unstable requirements and poor project management. A more recent survey of twelve U. K. companies also found that requirements problems accounted for 48% of all software problems [8]. These reports are in agreement with Neil and Laplante [17], who report on developers who thought that their companies did not do enough requirements engineering.

In this study, we explore how RE practices influence the success or failure of software development projects. Our objective is to not only explore the direct effects of RE practices but also to examine other factors surrounding the gathering and management of requirements that may impinge on project outcomes. Quantitative information of this kind is likely to be useful to organizations as they understand their project risks and learn lessons from both their failures and successes. Given the pressure under which many project managers function, better knowledge of the impact of both direct and indirect RE practices will help managers 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 RE practices are important predictors of software project outcomes.

The rest of the paper is organized as follows. In Section 2 we describe our investigation and the development of a questionnaire. In Section 3, we discuss some details of the questionnaire responses, develop success prediction equations and test their generalisability. Section 4 concludes the paper with a discussion and suggestions for further research.

2. Our Investigation

To document practitioners’ views regarding software project success or failure and the practices they consider risks to project success, we conducted wide ranging structured discussions with twenty-one senior software practitioners employed by a large U. S. financial services organization. This discussion focused on software projects with which the practitioners had been recently involved. We later had similar discussions with another group of software practitioners from the U. S. who worked in a variety of commercial organizations and also developed in-house software. Based on these discussions we developed a questionnaire in order to investigate those software development factors that lead to successful project outcomes. We chose a survey because of its simplicity and our wish to find relationships amongst variables; a survey gave us coverage of a greater number of projects at lesser cost than would an equivalent number of interviews or a series of case studies.

The original practitioner group (Group 1) responded to our questionnaire by answering it twice, once for a project they considered successful and once for a project they considered a failure. Our questionnaire was later distributed to the second group of practitioners based in the U. S. (Group 2). The questionnaire was then answered by a group of Australian practitioners based in Sydney (Group 3). Our sample is not random but rather a convenience sample of practitioners known to us. In the rest of this paper we refer to either data specific to each of the respondent groups (Group1, Group2 or Group3) or to the overall data set, i.e., the combined data from all respondent groups.

The questionnaire was organized into a number of sections covering the entire software development process. We also asked respondents if they considered the project they referenced, when answering the questionnaire, to be a success or a failure. There are many different and overlapping definitions of project success, some narrowly related to scheduling, such as being on time [21] and comparing actual versus planned elapsed time for the entire
project [19]. Other definitions are broader such as successful projects are on time and within budget [15], or “giving what the customer wants when they want it, at the agreed upon price” [4]. However, many authors focus on critical success factors for project development rather than on success outcomes [10, 18, 20]. In this study we do not define success in advance (although in our discussions with practitioners many different views of project success were thoroughly explored) but instead allowed the respondents to use their own (implicit) definitions. Only questions broadly related to requirements engineering and requirements management are considered in this paper. Sections of the questionnaire not considered here are discussed elsewhere e.g., [26, 27, 28]. Our perspective could be largely considered a review of RE risk factors in in-house development projects. In-house software development failure is unlikely to receive the same attention as third-party software development failures with their attendant litigation and negative media coverage.

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 co-linearity. Therefore, we chose to conduct an initial exploratory study as a first step in understanding this data. To identify the most critical RE risk factors we develop a prediction equation for software project outcomes from one data set and then test its generalisability, by using it to predict project outcomes, on the other two data sets. This paper contributes to a better understanding of what actually happens in organizations that develop in-house software and which are the most important RE risk factors that are predictors of software project outcomes.

3. Questionnaire Responses

Group 1 respondents are IT developers, project leaders, and project managers employed by a large U.S. financial services organization. These practitioners were based in a number of different U.S. locations and were involved with the development of in-house software. Group 2 respondents had a similar background to Group 1, and were also located in the U.S. However, Group 3 respondents, though working on similar kinds of projects in similar domains, were based in Sydney, Australia, and as such have a somewhat different background. Most Australian respondents were employed in the IT departments of large business concerns, mostly financial and insurance, and were involved in developing in-house software.

We received completed questionnaires from 132 respondents, reporting on 153 distinct projects. We asked the developers of Group 1 to focus on one project they considered a success and one that was less successful. They reported on 55% unsuccessful projects and 45% successful projects. Groups 2 and 3 respondents each completed a single questionnaire related to a recently completed project of their choice. Group 2 reported on 70 projects (83% successful) and Group 3 on 41 projects (88% successful). The projects selected by our respondents varied from small in-house developments with one person that took a few months to complete, to large multiyear in-house projects with over 200 developers. Fourteen percent of the projects were maintenance or enhancement projects. There were no significant differences between the groups for number of personnel employed on the projects, or for the number of maintenance and development projects described.

As noted earlier, Group 1 respondents each filled in two questionnaires, one for a successful project and one for a less successful project. Group 2 and Group 3 respondents, describe significantly more successful projects than Group 1, i.e., when given a choice of what kind of project to describe most respondents preferred to discuss a successful one or, alternately, there
were more successful projects. However, project success rates reported in the Standish 1999 Report [23] and 2001 Report [24] would suggest that our respondents preferred to focus on successful projects. There is no significant difference between Group 2 and Group 3 for the number of successful projects included in their data sets.

Table 1 shows the percentages of “yes” answers to the questions asked. In some cases, for reasons of clarity and space, we have consolidated the answers to those questions originally answered on a five point Likert scale to above average and average/below average. For reader clarity, we have reworded the original question to provide a label that fits this revision. Table 2 shows significant associations with project success (p < 0.05). We have classified our RE questions in the tables as follows: “C” refers to questions that deal with the project sponsor, customers and users; “R” to questions directly related to requirements issues; and “M” to questions related to the management of the development process.

The rest of this section is organized as follows: Section 3.1. discusses responses to the questions; Section 3.2. focuses on development of a prediction equation for project success and failure based on the Group 3 data set and tests its generalisability on the other two data sets.

3.1 Detailed responses
Section 3.1 is divided into three sub-sections 1) sponsor, customer and users, 2) requirements issues, and 3) project manager and project management. Each sub-section includes the questions asked, and a short discussion of the factors most highly associated with project outcomes.

3.1.1. Sponsor, Customer and User Questions
(C1) There was a high level of customer/user involvement, (C2) there was a high level of customer/user confidence in the development team, (C3) there was a low level of Customer/user turnover, (C4) senior level project sponsorship lasted right through the project, and (C5) the project was affected by large numbers of customers and users.

The responses to these questions show the level of support and commitment that the project actually has from customers, users and the project sponsor. Keil et al [13] devote one of four quadrants in a risk categorization framework to the customer mandate. They suggest that risk mitigation strategies that create and maintain good relationships with customers and promote customer commitment to a project cannot be built overnight. Senior management and project managers must establish and maintain long-term relationships with both their customers/users and with company senior management. Most risks in this quadrant cannot be controlled by the project manager but they can be influenced. Risk mitigation strategies involve relationship management, trust building and political skills. Projects in which either top management or user commitment is lacking represent a high-risk proposition. But initial commitment is not enough. Once a project has started project managers must constantly gauge the level of commitment from both top management and the user community to avoid being caught in a situation where support for the project evaporates. Lack of executive support has now replaced user involvement as the number one cause of project failure [13].

Table 1: Percentage “Yes” Responses to Questions
<table>
<thead>
<tr>
<th>ID</th>
<th>Question</th>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
<th>Total1</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>There was a high level of customer/user involvement</td>
<td>60</td>
<td>64</td>
<td>83</td>
<td>68</td>
</tr>
<tr>
<td>C2</td>
<td>There was a high level of customer/user confidence in the development team</td>
<td>41</td>
<td>61</td>
<td>63</td>
<td>55</td>
</tr>
<tr>
<td>C3</td>
<td>There was a low level of customer/user turnover</td>
<td>57</td>
<td>68</td>
<td>49</td>
<td>65</td>
</tr>
<tr>
<td>C4</td>
<td>Senior level project sponsorship lasted right through the project</td>
<td>55</td>
<td>73</td>
<td>73</td>
<td>70</td>
</tr>
<tr>
<td>C5</td>
<td>The project was affected by large numbers of customers/users</td>
<td>24</td>
<td>33</td>
<td>32</td>
<td>30</td>
</tr>
<tr>
<td>R1</td>
<td>Requirements were gathered using a specific method</td>
<td>50</td>
<td>46</td>
<td>59</td>
<td>52</td>
</tr>
<tr>
<td>R2</td>
<td>Requirements complete and accurate at project start</td>
<td>3</td>
<td>51</td>
<td>54</td>
<td>40</td>
</tr>
<tr>
<td>R3</td>
<td>If not complete at start, requirements were completed before project ended</td>
<td>44</td>
<td>78</td>
<td>85</td>
<td>55</td>
</tr>
<tr>
<td>R4</td>
<td>Overall, the requirements good</td>
<td>44</td>
<td>78</td>
<td>86</td>
<td>71</td>
</tr>
<tr>
<td>R5</td>
<td>The project have a well defined scope</td>
<td>42</td>
<td>78</td>
<td>83</td>
<td>70</td>
</tr>
<tr>
<td>R7</td>
<td>Customers/users made adequate time available for requirements gathering</td>
<td>61</td>
<td>70</td>
<td>66</td>
<td>66</td>
</tr>
<tr>
<td>R8</td>
<td>There was a central repository for requirements</td>
<td>56</td>
<td>61</td>
<td>80</td>
<td>64</td>
</tr>
<tr>
<td>R10</td>
<td>The size of the project had a negative impact on requirements</td>
<td>50</td>
<td>26</td>
<td>46</td>
<td>37</td>
</tr>
<tr>
<td>M1</td>
<td>The requirements were managed effectively</td>
<td>39</td>
<td>57</td>
<td>68</td>
<td>56</td>
</tr>
<tr>
<td>M2</td>
<td>The project manager was experienced in the application area</td>
<td>74</td>
<td>58</td>
<td>68</td>
<td>68</td>
</tr>
<tr>
<td>M3</td>
<td>A defined development methodology was used</td>
<td>49</td>
<td>55</td>
<td>81</td>
<td>61</td>
</tr>
<tr>
<td>M4</td>
<td>The methodology was appropriate for the project?</td>
<td>42</td>
<td>56</td>
<td>81</td>
<td>60</td>
</tr>
<tr>
<td>M5</td>
<td>The delivery decision was made with appropriate requirements information</td>
<td>30</td>
<td>64</td>
<td>51</td>
<td>52</td>
</tr>
<tr>
<td>M6</td>
<td>The project manager was able to choose the development methodology</td>
<td>42</td>
<td>33</td>
<td>33</td>
<td>35</td>
</tr>
</tbody>
</table>

Our results exhibit a high level of multi co-linearity. From these results, and the significant relationships shown in the data, we hypothesise that (C1) there was a high level of customer/user involvement, (C2) there was a high level of customer/user confidence in the development team, (C3) a low level of customer/user turnover and (C5) the project was not affected by large numbers of customers/users are the result of (C4) senior level project sponsorship lasted right through the project. While C1, C2, C3 and C4 were significantly associated with the project outcome we found no such relationship between C5 and project outcomes. While C3 and C5 are not factors that commonly appear in the project risk literature our practitioners suggested that they were risk factors likely to be associated with project failure. Our overall data set includes four projects employing more than 80 staff; all were negatively impacted by large numbers of customers/users. This result reinforces the need to find the voice of the customer by identifying different classes of users, selecting and working with individuals who represent each user class and other stakeholder groups and agreeing on who the requirements decision makers are for the project [29]. Our larger projects had significantly more problems with large numbers of customers/users and customer/user turnover. Given that we had a majority of small projects, C3 and C5, and their relationships with other factors warrants further research.

It is clear from the results shown in Table 1 that Group 2 and Group 3 projects have some similarities. This is not surprising given that both groups consist of mainly successful projects.

1 This column represents the percentage of “yes” answers to the questions for all projects.
However, Group 3 projects have a significantly higher level of customer/user involvement and a significantly lower level of customer/user turnover.

### 3.1.2. Requirements Questions

(R1) Were the requirements gathered using a specific method? (R2) Were requirements complete and accurate at the start of the project? (R3) If not complete at the start were requirements completed before project completion? (R4) Overall were the requirements good? (R5) Did the project have a well-defined scope? (R7) Customers/users made adequate time available for requirements gathering? (R8) Was there a central repository for requirements? and (R10) Did the size of the project have an impact on requirements?

One of the first steps in requirements elicitation consists of defining the requirements development process; question R1 addresses this. Definition of a requirements development process will normally include the use of a RE methodology [29]. However, we found that R1, gathering requirements with a specific method, was not significantly associated with project outcome for any of the groups. Overall the groups had fairly similar results for this factor. A further risk factor suggested by our practitioners was the size of the project in terms of functionality. Project size hampers requirements gathering, and leads to unclear, incomplete, and potentially unstable requirements [6]. Our results support the importance of project size as a project risk factor.

Many risks threatening software projects involve the ambiguities and uncertainties in establishing the project’s scope and requirements [13]. Examples of specific risks are misunderstanding requirements and not managing change properly. The project manager has considerable control over these risks but they do require “skillful interfacing” with the user/customer. Hence evolutionary approaches, such as iterative development, are sometimes used [13]. Defining and documenting scope is one of the steps in requirements elicitation that overcomes these difficulties to provide a reference against which to evaluate the proposed requirements [29]. Customers and users need to be educated on the impact of scope changes in terms of both project cost and schedule. Our results support the importance of a well-defined project scope.

While a central repository for requirements is not commonly cited in the literature as a success factor our practitioners suggested that it was important. The relationship of C8, there was a central repository for requirements, with project outcomes suggests that this factor supports effective requirements management. The fact that overall only 64% of the projects used a central repository tends to encourage the view that there is significant room for improvement in requirements management. Group 3 with the highest proportion of successful projects has a central repository for requirements for 80% of projects.

Our results support the importance of good requirements; R3 and R4 are significantly associated with project success for all the groups. It is interesting that nearly 30% of our projects did not have good requirements at any stage and over 75% of these projects were failures. Again we see similarities between Group 2 projects and Group 3 projects with Group 1 projects having significantly lower percentages for most requirements factors. It is interesting to note the 3% of Group 1 projects with complete and accurate requirements at the project start. Even allowing for the few projects using prototyping and JAD, and the 55% unsuccessful projects, this figure is very low.
The data suggests that R10, projects small enough that requirements gathering is not impacted, where R7, customers/users make time for requirements gathering, will have R5, a well defined scope, and will have R4, good requirements, which are R8, stored in a central repository.

**Table 2: Significant Associations with Project Success**

<table>
<thead>
<tr>
<th>ID</th>
<th>Question</th>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>There was a high level of customer/user involvement</td>
<td>0.000</td>
<td></td>
<td></td>
<td>0.000</td>
</tr>
<tr>
<td>C2</td>
<td>There was a high level of customer/user confidence in the development team</td>
<td>0.001</td>
<td>0.040</td>
<td></td>
<td>0.000</td>
</tr>
<tr>
<td>C3</td>
<td>There was a low level of customer/user turnover</td>
<td></td>
<td>0.005</td>
<td></td>
<td>0.019</td>
</tr>
<tr>
<td>C4</td>
<td>Senior level project sponsorship lasted right through the project</td>
<td>0.004</td>
<td>0.005</td>
<td></td>
<td>0.000</td>
</tr>
<tr>
<td>C5</td>
<td>The project was affected by large numbers of customers/users</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>Requirements were gathered using a specific method</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>Requirements complete and accurate at project start</td>
<td></td>
<td></td>
<td></td>
<td>0.006</td>
</tr>
<tr>
<td>R3</td>
<td>If not complete at start, requirements were completed before project ended</td>
<td>0.009</td>
<td>0.001</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>R4</td>
<td>Overall, the requirements were good</td>
<td>0.009</td>
<td>0.000</td>
<td>0.004</td>
<td>0.000</td>
</tr>
<tr>
<td>R5</td>
<td>The project had a well-defined scope</td>
<td></td>
<td>0.001</td>
<td></td>
<td>0.000</td>
</tr>
<tr>
<td>R7</td>
<td>Customers/users made adequate time available for requirements gathering</td>
<td>0.009</td>
<td>0.000</td>
<td></td>
<td>0.000</td>
</tr>
<tr>
<td>R8</td>
<td>There was a central repository for requirements</td>
<td></td>
<td>0.004</td>
<td>0.043</td>
<td>0.000</td>
</tr>
<tr>
<td>R10</td>
<td>The size of the project had an impact on requirements</td>
<td>-0.000</td>
<td>-0.000</td>
<td>-0.000</td>
<td></td>
</tr>
<tr>
<td>M1</td>
<td>The requirements were managed effectively</td>
<td>0.001</td>
<td>0.000</td>
<td></td>
<td>0.000</td>
</tr>
<tr>
<td>M2</td>
<td>The project manager was experienced in the application area</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3</td>
<td>A defined development methodology was used</td>
<td></td>
<td>0.033</td>
<td></td>
<td>0.008</td>
</tr>
<tr>
<td>M4</td>
<td>The methodology was appropriate for the project</td>
<td></td>
<td>0.002</td>
<td>0.007</td>
<td>0.007</td>
</tr>
<tr>
<td>M5</td>
<td>Delivery decision was made with appropriate requirements information</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td>0.000</td>
</tr>
<tr>
<td>M6</td>
<td>The project manager was able to choose the development methodology</td>
<td></td>
<td></td>
<td></td>
<td>0.020</td>
</tr>
</tbody>
</table>

3.1.3. Project Manager and Project Management

(M1) The requirements were managed effectively, (M2) Was the project manager experienced in the application area, (M3) Was a defined development methodology used? (M4) Was the methodology appropriate for the project? (M5) Was the delivery decision made with appropriate requirements information and, (M6) The project manager was able to choose the methodology

Risks associated with the project manager and project management have to do with project execution and include poor project management factors such as lack of an effective development methodology and insufficient staffing [13]. These are the risk factors that are
normally under the direct control of the project manager and hence project managers consider them to be moderate rather than high risk items. A disciplined development methodology, effective processes for managing change and ambiguity are needed and there is a danger that project managers may overestimate their own abilities [13]. This can lead to a failure to develop appropriate risk strategies. Another risk factor associated with project management is effort and schedule estimation. Truly reliable estimates are rare [24] as estimating is hard and estimating without knowing what the system requirements are is even harder [11]. Just over 50% of our projects had estimates made with appropriate requirements information.

Our practitioners suggested that a project was more likely to be successful if the project manager was experienced in the application area. Our data, however, did not support this view. Our results show the importance of a defined development methodology appropriate for the project. Nevertheless, while a defined development methodology should be used for a successful project it should be appropriate. We should not underestimate the importance of requirements management as requirements are never static but in a constant state of change. Our results support the importance of effective requirements management with Group 1 and Group 2 data sets and the overall data with significant associations with project success for this factor. Although our practitioners suggested that the project manager should choose the project’s development methodology our overall results did not support this opinion. While only about one third of project managers were able to choose the development methodology this variable was not significantly associated with project success. In fact, in nearly half the projects where the project manager chose the methodology respondents thought the methodology chosen was inappropriate.

Our results indicate that M2, a project manager experienced in the application area, using M4, a defined methodology appropriate for the project, M5, who makes the delivery decision with appropriate requirements information, and M1, who manages requirements effectively, will be associated with a successful project.

3.2. Predicting success

In order to identify factors that are useful in predicting project outcomes we need to focus on practices that occur early in the software development process. If project managers are able to identify problematic projects early enough, they can take mitigating action. In order to identify the most important factors that differentiate between successful projects and unsuccessful projects we developed a prediction equation based on our data. For a prediction equation to be useful it should be general enough to be used across a number of different environments. We chose to develop a prediction equation from the Group 3 data set as the Australian data set shows a number of differences from the other two groups (see Tables 1 and 2). We then tested the equation’s effectiveness in predicting the outcomes of Groups 1 and 2’s projects.

We use binary logistic regression to develop a prediction equation as our dependent variable, project success, is binary in nature. The best prediction equation developed from the Group 3 data consists of a single variable, R3, requirements were not complete at project start, but they were completed before the project ended. Given that our data set contains only 41 projects we can expect only a limited number of variables to enter into the prediction equation. A larger data set would be more likely to give a prediction equation with more variables. Using the prediction equation to forecast the outcomes of Group 3 projects, we predict 100% failed projects, 87% successful projects and 90% projects correctly overall. However, predicting the
outcome of a data set using an equation developed from the same data set is not really a test of the “goodness” of the prediction equation but rather more a self-fulfilling prophecy. To show the utility of a prediction equation we must test it on independent data sets. A good prediction equation will predict correctly at least 70% of the time [2].

Using our Group 3 prediction equation with Group 1 projects we correctly predict 74% failed projects, 67% successful projects and 71% correctly overall. Using the Group 3 equation to predict the outcomes of Group 2 projects we correctly predict 75% failed projects, 83% successful projects and 81% correctly overall. Overall, our equation is reasonably accurate.

4. Discussion and Further Research

For in-house software development projects our results did not support

- using a specific requirements gathering method
- project managers with experience in the application area
- project manager being able to choose the development methodology

as factors significantly associated with project success.

What is more important than a defined requirements gathering methodology is that the project has a defined software development methodology that is appropriate for the project, as both of these variables were significantly correlated with project success. Any requirements gathering methodology must fit within the software development methodology used.

Overall, our results indicate that the most important factors for the early prediction of project success include:

- a high level of customer/user involvement
- a high level of customer/user confidence in the development team
- customers/users who make adequate time available for requirements gathering
- senior level project sponsorship that lasts right through the project
- good requirements
- completing the requirements at some stage before the project end
- a well-defined project scope
- a central repository for requirements
- project size
- effective requirements management
- a delivery decision made with appropriate requirements information.

For early prediction of a project with problems, a project manager should carefully monitor all the above factors as part of a comprehensive risk management strategy. While our prediction equation indicates that R3, completing the requirements at some stage before the project ends, is a key factor for the prediction of project outcome, there is a high degree of co-linearity in our data. R3 is significantly related to C1, C2, C4, R4, R5, R7, R8, R10(-), M1, M4, and M5. Hence a project manager who completes the requirements as some stage in the development process is likely to have all the other factors in place.
As noted earlier, a larger data set would have resulted in a prediction equation containing more variables. If we had combined Groups 2 and 3 (111 projects) to develop a prediction equation, our equation would have included R8 and M1; combining Groups 1 and 2 (112 projects), or using the overall data set (153 projects) we would have developed prediction equations including M1 and M5. However, because we developed our prediction equation on 41 projects our equation consisted of only one variable.

How generalisable are the factors identified in this study? While we believe that the results of this initial study are significant on their own, we intend to compare against factors important for good requirements in other environments, such as software developed for third parties. This research serves as a starting point for motivating our continuing research into requirements practices in industry.

The relationship between customer/user involvement (C1), with level of confidence in the development team (C2), is interesting and leads us to ask about causal effects. Are customers/users involved because they are confident in the development team or do they become more confident in the development team because of their involvement? Certainly there is likely to be a reciprocal effect: a positive involvement experience is likely to reinforce confidence in developers and vice-versa. What motivates customer involvement with the development team? What instils customers with confidence in the development team?

While customer/user turnover, and large numbers of customers and users, are not factors that commonly appear in the project risk literature our practitioners suggested that they were risk factors. Our larger projects had significantly more problems with large numbers of customers/users and customer/user turnover. Given that we had a majority of small projects the relationships of these two factors with other factors warrants further research. Do we have the same problems with projects with limited functionality and large numbers of customers and users?

References


McConnell Steve, 1989, Rapid Development, Microsoft Press, Redmond, WA.


Verner, J., Overmyer S., and McCain, K., 1999, “In the 25 years since The Mythical Man Month what have we learned about project management?”, Information and Software Technology, December, pp. 1021-1026.


Wiegers, K, 2003, Software Requirements, Microsoft Press, Redmond, WA, USA.