Information and Software Technology 52 (2010) 934–944

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Evaluating logistic regression models to estimate software project outcomes

Narciso Cerpa [a,*], Matthew Bardeen [a], Barbara Kitchenham [b], June Verner [c]

[a] Facultad de Ingeniería, Universidad de Talca, Curicó, Chile
[b] School of Computing and Mathematics, Keele University, Staffordshire ST5 5BG, UK
[c] Computer Science and Engineering, University of New South Wales, Sydney, Australia

ARTICLE INFO

Article history:
Received 23 November 2008
Received in revised form 25 March 2010
Accepted 27 March 2010
Available online 13 April 2010

Keywords:
Project outcome
Tailored cut-off
ROC analysis
Single-company data
Cross-company data
Classifier evaluation

ABSTRACT

Context: Software has been developed since the 1960s but the success rate of software development projects is still low. During the development of software, the probability of success is affected by various practices or aspects. To date, it is not clear which of these aspects are more important in influencing project outcome.
Objective: In this research, we identify aspects which could influence project success, build prediction models based on the aspects using data collected from multiple companies, and then test their performance on data from a single organization.
Method: A survey-based empirical investigation was used to examine variables and factors that contribute to project outcome. Variables that were highly correlated to project success were selected and the set of variables was reduced to three factors by using principal components analysis. A logistic regression model was built for both the set of variables and the set of factors, using heterogeneous data collected from two different countries and a variety of organizations. We tested these models by using a homogeneous hold-out dataset from one organization. We used the receiver operating characteristic (ROC) analysis to compare the performance of the variable and factor-based models when applied to the homogeneous dataset.
Results: We found that using raw variables or factors in the logistic regression models did not make any significant difference in predictive capability. The prediction accuracy of these models is more balanced when the cut-off is set to the ratio of success to failures in the datasets used to build the models. We found that the raw variable and factor-based models predict significantly better than random chance.
Conclusion: We conclude that an organization wishing to estimate whether a project will succeed or fail may use a model created from heterogeneous data derived from multiple organizations.

1. Introduction

Software has been developed since the 1960s but we still have not learned enough to substantially increase the success rate of software development projects. Charette suggests that "Billions of dollars are wasted each year on failed software projects" and that "We have a dismal history of projects that have gone awry" [10]. Developing software systems is an expensive and often difficult process since development projects are affected by problems, such as poor project management, cost and schedule overruns, poor quality software and under-motivated developers [3,8,57]. Charette provides a long list of high-profile failed projects from around the world in his "Hall of Shame" [10]. Software development failures and poor product quality lead to a lack of credibility and communication problems among software developers, senior management, customers, and users, which in turn makes the software development process and product implementation even harder [16,18]. The project manager (PM) and the development team must deal with many pressures from project stakeholders (i.e., upper level management, marketing, accounting, customers, and users) during the software development process that impact both the cost and the quality of the software produced [4]. These pressures may include tight schedules and late changes to requirements caused by a poor requirements elicitation process and/or changing business needs. One study found that 20% of software projects failed, and that 46% experienced cost and schedule overruns or significantly reduced functionality [32], while other studies suggest various failure rates for software development projects up to 85% [22,25,34]. Several researchers have reported on software project failures from the customer/user perspectives (e.g., [15,23,67]), but it is also important to recognize the effects of failures on software development staff. Late projects usually cause developers to suffer long hours of unpaid overtime, loss of

* Corresponding author. Tel.: +56 75 201710; fax: +56 75 201713.
E-mail addresses: ncerpa@utalca.cl (N. Cerpa), mbardeen@utalca.cl (M. Bardeen), b.a.kitchenham@cs.keele.ac.uk (B. Kitchenham), june.verner@gmail.com (J. Verner).

0950-5849/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2010.03.011

motivation, and stress leading to high staff turnover and its associated costs. Although there are many guidelines for successful software development (e.g., [4,33,35,41]), few project post-mortems are published [57], and little understanding is gained from the results of past projects within organizations.

The capacity of the project manager to understand the consequences of actions taken during the software development process and the effect that decisions have on the development outcome are critical to the success of a software project. Key development practices are frequently ignored and early warning signs of project failure are not understood. Some project failures are predictable and avoidable, although it is not always possible to identify which aspects are important to success and failure early enough to take action.

The literature provides many definitions of the success or failure of a software project, as well as describing various aspects influencing project outcome. For example, some of the aspects are: organizational structure, communication with customer/users; user requirements and requirements specification, scheduling and project budget, customer satisfaction; product quality; leadership; upper management support; personality conflicts; software development methodologies; business processes and resources; and the project management process and tracking tools [10,12,15–17,19–25,32,36–45,49,51–58,60–63,66,67]. As can be appreciated, the number of variables that influence these aspects are innumerable. Therefore it firstly seems necessary to identify those aspects most important in ensuring project success, then to identify those variables that have the most influence over the identified aspects. With this information in hand, a project manager can quickly identify problem areas in his project and work to improve them.

Because the software practitioner perspective is extremely valuable to the discipline of software engineering, and in particular to the management of the software development process, this paper describes a survey-based empirical investigation into factors that contribute to project outcome from the practitioner perspective. While we are aware that there are many objective definitions available for whether a project is a success or failure [2,40,45], we choose to focus on the developer's perception as that is the information most readily available to a project manager.

The goal is to produce a tool that may help the project manager to predict project outcome and identify problem areas in the project to efficiently allocate scarce project resources. To achieve this we first need to identify those elements which correlate to project success and failure, reduce the dimensionality of these elements, then to build models to support predictions from these elements, then to test the models for their predictive capacity.

A number of other approaches have been used to develop models to predict if a software project will succeed or fail [1,47]. For this analysis, we focus our attention on prediction using a logistic regression model. In such models adjusting the classification threshold, i.e., the cut-off point, is key to good predictions [7,46]. An analysis of the receiver operating characteristics (ROC) graph is then used to compare the predictive capacity of the various models [5,31].

In effort estimation, a parallel field in software engineering, research studies have suggested that single-company (homogeneous) data sets are necessary to produce more accurate estimations [27,28]. However, generating a single-company data set often requires a prohibitive amount of time to collect enough data from past projects. During this time, the technology or practices of the organization may change and the past project data may no longer be representative [6]. As a result, using cross-company (heterogeneous) data sets to build prediction models is an attractive alternative. In this paper we use cross-company data sets to build models and test them against a holdout data set from a single organization.

Thus, our contribution is to provide project managers with insight into: (a) which practices practitioners believe have the greatest impact on project success and failure, hence (b) where to focus their attention when resources are constrained, and (c) issues affecting the prediction of project outcome.

The research questions addressed in this paper are:

RQ1: Are there a small number of key aspects that impact project success and failure?

RQ2: If we try to predict failure using logistic regression, is it better to use the identified aspects (raw variables) or to reduce the variable set further with principal components factor analysis?

RQ3: Is it possible to use models built on data from other organizations to predict successful and failing projects in a specific company?

2. Related work

Previous research on software project outcome prediction has been based on the identification of software characteristics which can be taken as success factors [1] or threats [48] with a probability of occurrence. Since some project characteristics will lead to specific project outcome, it seems essential to identify the successful project characteristics in order to determine similarities among projects [64]. In order to have reliable predictions of project success we need to have an appropriate method to measure characteristics and identify similar projects.

Learning from past projects is essential to increase the probability of success of future projects. A basic definition of a successful software project is that the associated costs and development time are within the estimates, the functionality is sufficient and the software quality is satisfactory for the client. However, this definition is *post-hoc*; by the time these problems are visible the software project has already failed. Instead researchers have focused on identifying those aspects that serve as predictors or causal agents for project failure or success.

Abe et al. collected 29 metrics classified into categories such as development process, project management, company organization, human factors, and external factors to build a software project prediction model using a Bayesian classifier [1]. In a different scope, Wang used a *k*-means clustering algorithm to predict the success of open source software projects. The clustering was based on those groups of characteristics or success measures that describe open source software outcome, such as project outdegree (interaction between actors), active developer count change trend, bugs fixing speed and project releases [59].

Cheng and Wu proposed an evolutionary support vector machine inference model to dynamically predict project success, which was built based on a hybrid approach that includes a support vector machine (SVM) and a fast messy genetic algorithm (fmGA). This model integrates the process of continuous assessment of project performance to dynamically select continuous factors that influence and have the ability for predicting project success [11].

Some project characteristics or metrics are more important than others in determining project outcome and to identify similarities among projects. Some approaches based on similarity identification have been used in the past (e.g. nearest neighbor, friends, key success driver analysis). The nearest neighbor uses a distance measure, a simple approach used in case-based reasoning to identify similar projects. The friends approach uses all available information to identify a set of similar projects. The third method consists of analyzing and identifying the key success drivers to find similar projects [64]. A comparison of these methods shows that using key drivers of project success gives better software project outcome predictions. Therefore, it is important for software developers to identify the key project characteristics to improve their control over project outcome. Another approach to predict software project success which was mentioned above is the identification of project threats and their probability of occurrences [48]. Two probabilities are usually used, the probability of a threat occurring; and the probability of the threat causing a negative impact or consequence on some project success criteria.

Mirroring the approaches mentioned above and used previously, in this article we analyze software project data collected through a series of surveys given to software developers from the US and Australia with the purpose of identifying the most important variables affecting project outcome. This permits the construction of prediction models using software project key success factors to which the project manager should address the projects' resources.

When building prediction models is essential to evaluate their performance, and one method of doing so is to measure and compare their prediction accuracy. However, Lessmann et al. note that comparing prediction models based on their accuracy at given cut-off points is not the best method of determining their effectiveness. They suggest comparing the areas under the classifiers' ROC curves since these can be "assessed independently from thresholds" [31]. This issue arises because the cut-off point is used as a threshold to classify the output a prediction model (a real number in the range of 0–1 inclusive – corresponding to a probability) into two categories – 0 or 1 (in this case, failure or success respectively). Thus if the logistic regression model returned a value of 0.86 for a given project, and the cut-off point was 0.5, the project would be classified as successful. Comparing models solely based on accuracy could lead to differing conclusions depending on the value of the cut-off point since the chosen cut-off point has a large impact on the overall accuracy of the model.

Another advantage of analyzing the area under the ROC curves is that it allows a relatively straightforward way to determine if the model predicts better than random chance. The diagonal line $y = x$ of an ROC curve represents a classifier that randomly predicts the output [14]. Given an ROC curve in which a prediction model's performance appears slightly better than random, it is logical to ask if this performance is significantly better or it is just better by chance. This has been answered by DeLong et al. who presented a non-parametric approach for comparing the area under the ROC curves [13].

In previous articles we have used subsets of the collected data to address issues such as cost, effort, and schedule estimation [54,55], US project management practices [51,56], predicting good requirements [53], and exploring the application of case based reasoning techniques to projects considered successful by developers and management [62,63]. In [9] and [58] we used only the projects classified as failures from this data and from an additional dataset collected from Chilean software developers to identify factors leading to project failures.

This paper differs in that all the variables from our data sources from the US and Australia are analyzed together with the aim of identifying those key characteristics that contribute to project success, estimating project outcome, and comparing the use of raw variables versus factors in the construction of prediction models. This allows us to identify factors that are important across the different data sets and those that appear to be unimportant in all sources. We also use the analysis of ROC curves to evaluate and compare the predictive capacity of models generated using logistic regression, both over raw variables and over factors obtained by principal components analysis.

We have not previously considered the data in its entirety, nor have we previously addressed the issue of using heterogeneous data to build models to predict the success of software projects within a single company.

In the next section we describe the data we collected including some project demographics. Subsequent sections describe the methodology used and the results of the data analysis. In the final section we provide a discussion of the results and the conclusions.

3. Data

In this section we discuss our survey construction, administration, and responses, project demographics and validity.

3.1. Survey construction

Based on software engineering literature, we identified important issues affecting project outcome and then discussed these issues with over 90 developers of in-house software. These discussions served to refine the issues to those that were most important from a developer point of view based on their past experiences. We then developed a questionnaire to investigate which practices lead to successful project outcomes.

This survey was organized into a number of sections covering the software development process:

1. sponsor/senior management,
2. customer/user,
3. requirements,
4. estimation and scheduling,
5. development process,
6. the project manager and project management,
7. the development team.

We also asked respondents if they considered the project they referenced when answering the survey was a success or a failure. The definition of software project success or failure may be different depending on stakeholders' perceptions [39,60,61], but current studies have shown that developers' perceptions have important commonalities across cultures [38,43,45]. During the discussion about the issues identified in the literature, it was evident that all of the developers had their own definition of what was a successful project and what was a failed project. Thus we did not define project success in the questionnaire as we felt that respondents had enough knowledge to decide whether the project they referenced in the questionnaire was successful or not.

3.2. Survey administration

Our questionnaire was distributed to practitioners from a large US financial institution (data set 1) who each responded by answering it twice, once for a recent project they considered successful and once for a recent project they considered a failure. The questionnaire was later distributed to a second group of practitioners, from various companies in North Eastern USA (data set 2), and a third group of practitioners from a number of different Australian companies (data set 3); each answered the questionnaire once. Our sample is not random but rather a convenience sample.

3.3. Project demographics

In all we received completed questionnaires from 143 respondents, reporting on 164 distinct projects. The majority of our respondents were developers involved with software for use within their own organizations, although a few were involved in outsourced development projects. Projects were classified as a success or not according to the participants' view of the project.

Table 1
Percentage of projects by employee type and number of employees.

# of employees	Company employees (%)	IT contractors (%)	Total staff (%)
None	2.5	38.0	0
1–4	33.1	37.2	27.3
5–9	28.9	14.0	27.3
10–19	14.9	5.0	21.5
20–29	8.3	3.3	6.6
30–49	5.8	1.7	9.9
50–100	6.6	0.8	5.0
>100	0	0	2.5

The first set of 42 questionnaires (data set 1), described 42 separate projects, 21 were regarded as successful and 21 unsuccessful. The second set of responses (data set 2) included descriptions of 80 unique projects with a 71% success rate. While the third set of responses (data set 3) included 42 projects with a 78% success rate.

Overall 64% of projects were regarded as successful and 36% unsuccessful, 86% were development projects (62% successful), and 14% were large maintenance/enhancement projects with significant numbers of developers assigned to them (75% successful). Most project managers (68%) had experience in the application area, though 32% had little or no experience in the application area of the software developed. Additional demographics are given in Table 1. Based on this table we can see that 2.5% of the projects do not have company employees and are handled by IT contractors alone. Thirty-eight percent of the projects do not utilize IT contractors. Obviously, there are no projects without any staff.

3.4. Validity

An analysis of validity assesses how well a research instrument identifies concepts of interest and has implications for the measures used, findings and conclusions. Wohlin et al. [65] discuss validity under the following headings: construct validity, conclusion validity, and internal and external validity. Construct validity is concerned with the relationship between theory and observation. Conclusion validity is concerned with the relationship between the treatment and the outcome; to make sure that there is a statistical relationship i.e., with a given significance. Internal validity is concerned with making sure that a relationship between factors and outcome is not a result of factors over which we have no control or have not measured. External validity provides support for the generalization of the results of the study. Validity supports the repeatability of the study findings. In this study it is important to have a balance between the different types of validity; in particular construct, internal and conclusion validity. We are testing the variables proposed by practitioners and augmented by the literature to construct a theory related to what contributes to successful software development projects.

- *Construct validity*: Our comprehensive review of the literature and discussions with 90 software development practitioners leads us to believe that many of the critical factors that define a successful software development project were identified before developing the questionnaire. The literature is rich in risk management and software development failures [2,9,10,21–23,25,32,38–42,44,45,51,60,66]. The subjects were asked to respond a questionnaire regarding a past project, but they were not informed about the main objectives of the research to avoid hypothesis guessing [65].
- *Conclusion validity*: Factor analysis and logistic regressions are sufficiently powerful to reveal valid patterns in the data. The Kaiser–Meyer–Olkin adequacy test and Bartlett's test of sphericity showed that factor analysis was appropriate for the data [50]. We believe that our measures are reliable since the variables and measurement scales were derived from interviews with 90 experienced software development practitioners and a literature review. All practitioners participating in the study had sufficient experience in a variety of software development projects to respond the questionnaire. Instructions for answering the questions were provided to ensure a reliable and consistent treatment implementation.
- *Internal validity*: Our respondents have experience in software development and, as a result, have opinions based on their professional experience regarding particular aspects of software development they perceive are important contributors to project success. As always, a threat to internal validity is the subconscious desire of survey respondents to overemphasize the positive and under-emphasize the negative. In data set 1, we specifically asked for both a successful and a failed project from each respondent to avoid this bias. In the other two data sets we asked respondents for only one project of their own choosing. The results show that the respondents in data sets 2 and 3 tended to report many more projects as successful. We notice that those that do report negative project outcomes are more critical in their responses [9].
- *External* validity: Our research sample was gathered through a convenience sample; all the US respondents and some of the Australian respondents were involved in courses related to project management. Particular considerations of external validity are a reflection of the practitioners sampled. In this study, respondents reported involvement in professional software developments across several industries, project sizes, and project types including in-house and outsourced development projects. The only incentive they were offered was access to the final results.

4. Methodology

Since the data consists of three sets of surveys collected from different countries, we analyze the overall data set, and each of the three data sets corresponding to those surveys (data sets 1–3). Likewise, as stated in Section 1 we are also interested in whether it is possible to use data collected from the general industry to build a model and use it to predict project outcome for a particular company. Therefore we will also analyze the two heterogeneous data sets (data sets 2 and 3) together.

Another goal is to determine which is the better predictor of project success, logistic regression models developed using (a) factors identified through principal component analysis or (b) subsets of raw variables.

Our process consists of three overarching steps:

1. Identifying the variables that are strongly correlated with project success and failure for the overall data set, and each individual data set. Identifying groups of highly-correlated variables (factors) using principal components analysis.
2. Building the prediction models using logistic regression for both factors and variables over the combined data sets 2 and 3.
3. Testing the predictive capacity of the models by comparing the area under the ROC curves generated by our prediction models, and testing the effect tailoring the cut-off has on predictive accuracy.

4.1. Identify

To answer *RQ1*, we begin by reducing the variable set to remove extraneous data and make the generated models more comprehensible by:

1. Correlating all variables with project outcome, within each data set and overall. Spearman correlation is used as the variables are either categorical or ordinal [30].
2. Selecting variables significantly correlated with project outcome at the 90% level or above for the three data sets and overall.
3. Removing variables with large numbers of missing values.

Following this we use principal components factor analysis on the reduced set of variables to develop a set of factors suitable for predicting project outcome. This set of factors is composed of aggregate sets of original variables. These identified factors represent variables that are highly correlated with each other i.e., groups or clusters of variables [26].

We also apply principal components factor analysis to each of the data subsets separately and the combined data sets 2 and 3. This allows us to investigate the similarity between the factors in the overall data set and those identified from each data subset.

4.2. Build

In this step we develop two prediction models, one based on both the factors obtained in the previous step and the other on a reduced set of raw variables. We will use these models to identify if the created factors are more useful for predicting success than the raw variables. Since our dependent variable is categorical, we use a stepwise binary logistic regression, using a forward conditional approach over data sets 2 and 3 combined to build the predictive models.

4.3. Test

During the testing phase, the predictive models will be tested on data set 1 to generate the ROC curves, which will be subsequently used to answer *RQ2* and *RQ3*. We will use the area under the generated ROC curves to compare the general predictive capacity of the models. If the areas differ significantly, then we can say that the one with the larger area has a higher predictive capability – thus answering *RQ2*. Additionally, we can answer *RQ3* by comparing the areas under the curves generate by the models to that of an ROC curve generated by random chance i.e., 0.5. If they are significantly greater then we can be sure the models have generated better predictions than random chance.

To demonstrate the impact that the cut-off point has on a model's overall accuracy and rate of false positives and false negatives, we will vary the cut-off point of the model.

Koshgoftaar et al. suggest balancing the Type 1 and Type 2 errors by empirical assessment of the most appropriate cut-off level for allocating projects to the success and failure categories [29]. However, an alternative procedure is to use a cut-off probability corresponding to the actual proportion of successful projects.

Since there is a large discrepancy between the number of successful projects and the number of failures in data sets 2 and 3, we will change the standard cut-off point of the logistic regression from *0.5* to one that reflects the actual proportion of successful projects in the data used in building the models (i.e., to match the success rate).

5. Results

In this section we discuss the results from our study, following the steps described in the methodology section above.

5.1. Identify

We obtained a Spearman correlation between each success variable and project outcome for each data set, as well as the overall data set. Table 2 identifies all variables for which significant correlations ($p < 0.1$) were found in every dataset. Inspection of the 12 variables significantly associated with outcome showed that four had a large number of missing values (25–28%) and were hence eliminated from further analysis. Imputation methods were considered but discarded, even though the threshold for missing variables is around 40% [48]. We felt that if it was possible to accurately predict project success or failure without these variables, then the inclusion of the variables at a later point would only strengthen the models. No other variables had more than 5% missing values.

The excluded variables were "Adequate time was allowed for each of the phases", "How well did team members work together", "How high was the motivation of team members" and "How good was the working environment?". Additionally, we did not find that there was any correlation between the variables from the sponsor section of the survey and project outcome.

Note that the variable "Staff were added late to meet an aggressive schedule" is negatively correlated with project success. This is

Table 2
Correlations ($p < 0.1$) with significant project outcome for all individual data sets and the overall data set. None of the variables in the sponsor/senior management section of the survey was correlated with project outcome.

Section	ID	Variable	Overall	Data set 1	Data set 2	Data set 3
Sponsor	1	Nil	Nil	Nil	Nil	Nil
Customers and users	2	Level of confidence the customer has in project manager (PM) and team members	0.418**	0.495**	0.289*	0.367*
		The customers had realistic expectations	0.378**	0.370*	0.271*	0.393*
Requirements	3	The project had good requirements overall	0.498**	0.406**	0.465**	0.447**
Estimation and scheduling	4	How good were the estimates	0.474**	0.548**	0.367**	0.335*
		Staff were added late to meet an aggressive schedule	−0.356**	−0.278	−0.428**	−0.364*
Project manager	5	The project manager communicated well with staff	0.399*	0.335**	0.330**	0.415**
		How good was the project manager	0.382**	0.350*	0.360*	0.366*
		The project manager related well to software development staff	0.350**	0.444**	0.273*	0.358*
Development process	6	Adequate time was allowed for each of the phases	0.407**	0.331	0.409**	0.487**
Development team	7	How well did team members work together	0.557**	0.460**	0.625**	0.582**
		How high was the motivation of team members	0.404**	0.450**	0.422**	0.331*
		How good was the working environment?	0.366**	0.314	0.393**	0.367**

* Corresponds to $p < 0.05$.
** Corresponds to $p < 0.01$.

in agreement with the comments made by Brooks in 1975 [8]. Appendix A identifies variables that were not significantly correlated with project outcome in any of the data sets. That is not to say that a variable such as "There was a project manager" might not be a problem if the answer was "No". However, we found no systematic association between project outcome and these variables.

Another note is that the variable "The project had good requirements overall" is derived from the responses to two questions in the original survey. The requirements were considered to be good overall if they were complete and accurate at the start of the project or if they were completed adequately during the project.

The Kaiser–Meyer–Olkin adequacy test gave a result of 0.835, signifying that factor analysis was very appropriate for the data. The results of the Bartlett's test of Sphericity are $\chi^2 = 588.882$, $df = 66$, $p = 0.000$. These results indicate that (a) the degree of common variance among the variables is sufficient i.e., the factors extracted will account for a substantial amount of the variance and (b) the inter-correlation matrix did not come from a population in which the variables are non-collinear.

We applied factor analysis (principal component analysis with Varimax rotation [26]) to the first eight variables in Table 2 for the overall data set. We selected factors with Eigen values greater than 0.9. Although it is normal to restrict the analysis to factors with Eigen values greater than 1, the second and third Eigen vectors both had meaningful interpretations and together accounted for over 23% of the variation in the correlation matrix. This resulted in three factors which together explained 75% of the variance.

Each factor appeared to measure a distinct concept (see Table 3). Factor 1, "project manager capability" is composed of three variables (project manager communicated well with staff; how good was the project manager; and project manager related well to staff). Factor 2, "realistic project plans" is composed of four variables (level of confidence of customers in project manager and team, customers have realistic expectations, good requirements overall and how good were the estimates). Factor 3, "adding staff to meet schedule" is composed of one variable (staff were added late to achieve aggressive schedule).

Table 3
First three factors based on principal component analysis with Varimax rotation. Bold faced entries represent those variables with a loading of 0.5 or greater. These variables make up the three factors listed in the column heading.

Variables	Factor 1	Factor 2	Factor 3
Level of confidence of customers in PM and team	0.191	**0.747**	−0.160
Customers had realistic expectations	0.215	**0.732**	0.120
The project had good requirements overall	0.251	**0.776**	−0.195
How good were the estimates	0.473	**0.636**	−0.141
Staff were added late to achieve aggressive schedule	−0.148	−0.116	**0.958**
The project manager communicated well with staff	**0.877**	0.301	−0.021
How good was the project manager	**0.892**	0.259	−0.075
Project manager related well to staff	**0.793**	0.252	−0.232
Percentage variation explained	51.7	11.9	11.3
Factor description	Project manager capability	Realistic project plans	Adding staff to meet schedules

When we applied principal component analysis to each of the individual data sets separately, we found that it identified factors that were either identical to or were subsets of factors obtained from the overall data set. Likewise, the principal component analysis of the combined data sets 2 and 3 identified factors identical to the overall data set. Therefore we will only use the factors identified from the overall data set.

5.2. Build

To build the variable-based model, we applied logistic regression on the following variables:

- Level of confidence the customer has in project manager and team.
- The customer had realistic expectations.
- The project had good requirements overall.
- How good were the estimates.
- Staff were added late to meet aggressive schedule.
- The project manager communicated well with staff.
- How good was the project manager.
- The project manager related well to development staff.

We then performed a stepwise binary logistic regression using a forward conditional approach for the variables using data sets 2 and 3 combined. The final model is given in Table 4. When performing the regression, three variables are included in the model. The order in which they are introduced is:

- The project had good requirements.
- How good was the project manager.
- Staff added late to achieve aggressive schedule.

We performed a logistic regression using the factors identified in Section 5.1. The final model is shown in Table 5.

5.3. Test

With the logistic regression models built, we tested them against the homogeneous data from data set 1 to generate the ROC curves shown in Fig. 1. These curves represent the effect that

Table 4
Variable-based model.

Model	*B*	Standard error	Degree of freedom	Significance
The project had good requirements overall	2.106	.713	1	.003
Staff were added late to meet aggressive schedule	−1.372	.557	1	.014
How good was the project manager	.660	.301	1	.028
Constant	−2.395	1.240	1	.053

Table 5
Factor-based model.

Model	*B*	Standard error	Degree of freedom	Significance
Factor 1	.775	.284	1	.006
Factor 2	1.220	.320	1	.000
Factor 3	−.835	.262	1	.001
Constant	1.029	.287	1	.000

Factor 1 is defined by Table 3 as project manager capability, Factor 2 as realistic project plans, and Factor 3 as adding staff to meet schedules.

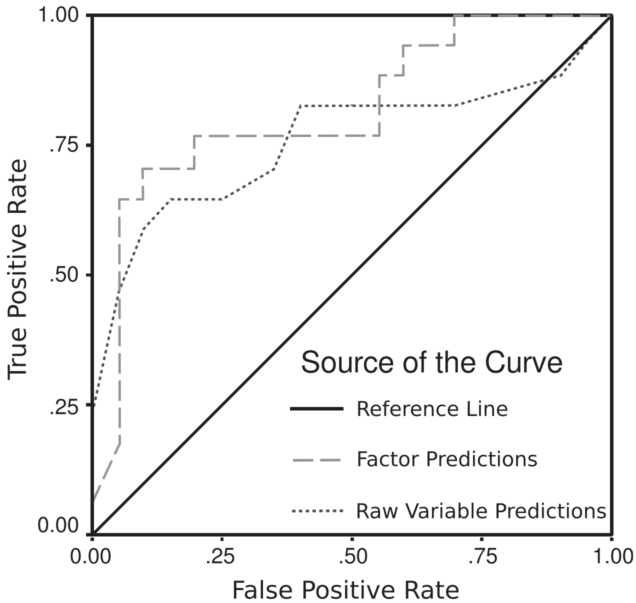


Fig. 1. The ROC curve for predictions of data set 1 using results from data sets 2 and 3.

varying the cut-off point of the models has on the rate of successful projects classified as successful (true positives) to successful projects classified as failures (false positives). An ideal predictor would have a point at (0, 1), i.e., all true positives with no false positives. In the following sections we analyze the area under the ROC curves to answer *RQ2* and *RQ3*.

5.3.1. Factors versus variables

Using DeLong et al.'s non-parametric approach [13] to compare the area under the factor-based and variable-based curves, we find the difference in area is 0.0574, the z-statistic is 1.4, and the p significance level is 0.162. This means while the area under the factor-based ROC curve is slightly greater than that of the raw variable-based model (Table 6), there is no statistical difference between the curves.

5.3.2. Performance versus random chance

In the previous sub-section we determined that there was no significant difference between the areas under the curves generated by our factor-based and variable-based models. However, for completeness we will compare the area under both to that of the curve generated by random chance.

Using DeLong et al.'s approach again [13], we find for the variable-based model that the difference between the areas is 0.257, the z-statistic is 2.904, and the p significance level is 0.0037.

For the factor-based model, the difference between the areas is 0.315, the z-statistic is 4.219, and the p significance level is 0.0001. These p significance levels mean that the areas under the ROC curves generated by both models differ significantly to the area under the random chance curve.

Table 6
Area under the ROC curve.

Test result variable(s)	Area	Std. error (a)	Asymptotic sig. (b)	Asymptotic 95% confidence interval	
				Lower bound	Upper bound
Raw variable-based model	.757	.087	.008	.586	.928
Factor-based model	.818	.073	.001	.675	.960

5.3.3. Standard cut-off versus tailored cut-off

We compared the changes in prediction accuracy of the models using the standard cut-off versus a tailored cut-off for the heterogeneous data (used in building the models) to those in the homogeneous holdout data. The standard cut-off was 0.5 and the tailored cut-off was 0.74 – the ratio of successes to failures in the heterogeneous data (data sets 2 and 3). Fig. 2 shows the results of the testing.

When testing the variable-based model against the build data we found that while we improved the prediction accuracy of failures using the tailored cut-off (from 50% to 73%), we decreased the prediction accuracy for successes from 95.9% to 69.6% (Fig. 2a). Overall, the variable-based model with standard cut-off predicted 83.8% while with the tailored cut-off it only predicted 70.7% of cases correctly.

Fig. 2b shows the results for the factor-based models against the build data. As in the case of the variable-based model, the tailored cut-off point improved the prediction accuracy for failure cases (from 46.2% to 65.4%) and decreased the accuracy for success cases (from 94.5% to 75.3%), resulting in a decrease of overall accuracy from 81.8% of cases predicted correctly to 72.7% of the cases predicted correctly.

Testing against the holdout data revealed some surprises. Changing the cut-off point from 0.5 to 0.74 for the variable-based model resulted in an increase in overall prediction accuracy from 67.6% to 75.7%. Again, failure prediction accuracy increased (from 65% to 85%) and success prediction accuracy decreased (from 70.6% to 64.7%) – Fig. 2c.

The overall accuracy for the factor-based model was the same for both cut-off points (78.4%) but again, failure accuracy increased and success accuracy decreased (from 80% to 90% and from 76.5% to 64.7% respectively) when tailoring the cut-off point. These results can be seen in Fig. 2d.

6. Discussion

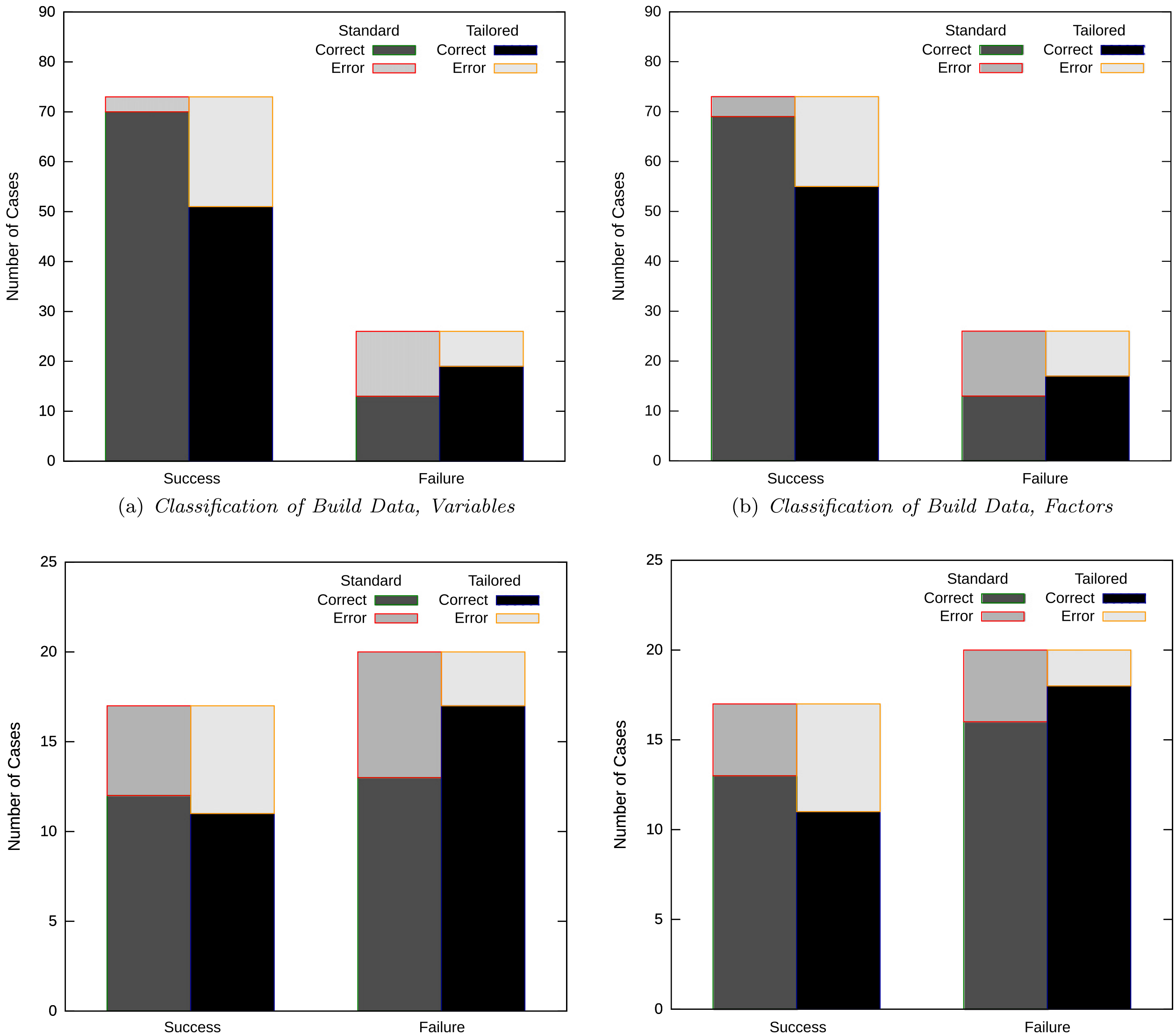
We will discuss our results by first analyzing the identified variables and factors. Then we will analyze the results of the factor and variable-based models. We also discuss the performance of the models in comparison to that of a random classification of project success. Finally we analyze the effects of tailoring the cut-off point and highlight the associated practical issues.

6.1. Identified factors and variables

One of the limitations of *post-hoc* correlation based studies is that the more questions researchers ask, the more likely it is that positive correlations will be obtained by chance. Our questionnaire has 42 questions which means that chance correlations are a potential threat to validity unless the number of responses is large. Pooling the information from three different populations has given us the opportunity to identify a core set of variables that are consistently related to our success variable and are therefore unlikely to have arisen by chance.

We have also identified 29 variables that were not found to be significantly related to our success variable in any of the three studies. This means that these variables can be excluded from any future questionnaire. Removing irrelevant questions decreases the time required to complete it and improves validity by substantially reducing the threat of spurious correlations.

The factor analysis of the eight questions that were consistently related to the success variable and had relatively few missing



(a) *Classification of Build Data, Variables*
(b) *Classification of Build Data, Factors*
(c) *Classification of Holdout Data, Variables*
(d) *Classification of Holdout Data, Factors*

Fig. 2. Logistic regression accuracy – models built with data sets 2 and 3 (build data) and tested against data set 1 (holdout data). Tailored cut-off is 0.74 and standard cut-off is 0.5.

values suggested that they related to three separate dimensions. This was supported by the stepwise logistic regression that identified only three variables from each of the three dimensions. Thus, our results have identified three critical issues:

1. How good are the project requirements.
2. How good is the project manager.
3. Whether staff are added late to meet schedules.

Overall, our results are consistent with other research into project success factors. In particular, adding late staff having a negative impact on project success is consistent with Brooks theory [8]. However, in our opinion, the need to add staff to a project is indicative of an earlier problem, so while it is correlated to project failure, further research is necessary to confirm whether it is a causal agent or not.

6.2. Factors versus variables

We did not find that a model based on the factors with the overall data set had significantly more predictive capacity than a stepwise model based on three of the raw variables. The predictive accuracy of the models are good, reaching as high as 85% accuracy overall on the hold-out data, but the results show that changing the cut-off point changes the predictive accuracy. The generated ROC curves cross over each other – this confirms that accuracy will be affected by changes in cut-off points. A good cut-off point for the variable-based model might not be the same as for the factor-based model.

6.3. Performance versus random chance

We see that both variable-based and factor-based models predict significantly better than chance when testing the model developed with cross-company data against data collected in a single company. This is a positive development, of course, but has stronger implications for the results of the survey. We had identified the potential unconscious bias for respondents to overstate their successes as a threat to the internal validity of our results. The data collection for the single company was controlled to prevent this bias by asking for one success and one failure from respondents, whereas no such precautions were taken for the survey when administered for data sets 2 and 3. Yet the models developed with this data were still able to predict project outcome on the single company data. This implies that this bias is either non-existent or relatively unimportant for the purposes of prediction.

6.4. Effects of tailoring the cut-off point

The effects of tailoring the cut-off point can be seen clearly in our results. With a standard cut-off point failures were poorly predicted, and when we set the cut-off point using *a priori* knowledge of the ratio of success to failures in build data we achieve a better prediction of failure accuracy. When testing on the build data we saw decreases of overall accuracy using a tailored cut-off point, yet when testing over the holdout data the accuracy was the same or actually increased. Thus it is advisable to make this information available when developing prediction models from cross-company data for the use by other organizations.

Likewise, for users of these models it is important to tailor the cut-off point when predicting the success of their own projects. For some organizations it might be wise to more accurately predict failures at the cost of overall accuracy. For example, if the cost of a misclassified failure is very high in relation to the cost of a misclassified success, as would be the case in an outsourcing company, misclassifying a small number of projects as successful could represent significant penalties. This could outweigh the cost of allocating additional resources to any successful projects that were misclassified as failures.

On the other hand when using these models to classify internal software projects where the cost of failure and the cost of success are relatively equal, the cut-off that gave the best overall accuracy might be more important. This could reduce the overall cost to the organization. Accurate predictions would, as well, allow better decisions from senior level management, permitting the cancellation of projects with little hope of success.

7. Conclusions and further work

Although our results are based on a convenience sample, they contribute to assessing the generality of aspects which influence project outcome since the data is drawn from a variety of different sources.

Our analysis has identified a small set of 12 variables that are significantly correlated with project outcome in all three groups of data, although we were only able to further investigate eight of the variables because there were many missing values for the other four. We believe that these represent a key set of variables for predicting project outcome. We have also identified a large number of variables that are not associated with project outcome in any of the data sets. We believe that most of these may be omitted from any general model of project outcome factors.

Referring to our specific research questions:

RQ1: *Are there a small number of key aspects that impact project success and failure?*
Yes, we found that some variables were hallmarks of project success and failure. We found three general areas where a project manager should focus their attention: project manager capability, realistic project plans, and not adding staff late to meet schedules. Within these three general areas there are a number of specific aspects, such as: confidence level in the team on the part of the customer, managing the customer's expectations, the estimation and requirement gathering process, and project manager interactions with the development team.

RQ2: *If we try to predict failure using logistic regression, is it better to use the identified aspects (raw variables) or to reduce the variable set further with principal components factor analysis?*
We did not find any difference in the predictive capability or accuracy between the models built with logistic regression over the raw variables or factors identified by principal components analysis.

RQ3: *Is it possible to use models built on data from other organizations to predict successful and failing projects in a specific company?*
We have shown that it is indeed possible to predict the outcome of projects from a specific organization using models created from cross-company data. Prediction accuracy over the hold-out data reached 85% in some circumstances and was never less than 67%.

7.1. In relation to previous work

These results agree with the relevant literature on software project outcome prediction that it is important to identify the key aspects influencing project success – be it using raw variables or principal components factors. Most of the models in the literature also used classifiers to predict software project success and used the identified key aspects for model building. The aspects used by some of the models are similar to the ones used here. For example: senior management, development process, requirements, customers, estimation, project manager and project management, and human resources. These models represent a project assessment tool to help managers in making decisions in a time efficient manner and permit them to take corrective actions to achieve a successful project.

7.2. Future work

Our logistic analysis omitted four of the key variables because they exhibited a relatively large number of missing values. These variables are:

- Adequate time was allowed for each of the phases
- How well did the team members work together?
- How high was the motivation of the team members?
- How good was the working environment?

Future data collection activities need to ensure that these variables are collected, so we can assess the impact they have on models for predicting project outcome. In addition, we will investigate the best approaches for dealing with missing values. For example, are analysis methods such as Bayesian networks or case-based reasoning better able to deal with missing values than standard logistic regression? We also need to consider improving our questionnaire by the addition of explicit failure factors if we wish to develop better predictive models.

7.3. Summary

We have identified specific aspects of projects where attention should be focused, developed prediction models for use in identifying troubled projects before they fail, demonstrated the importance of choosing the appropriate cut-off point for the prediction model, and finally demonstrated that models built from cross-company data can be used to successfully predict data within a single company.

Acknowledgement

This research was supported by the Chilean Grant FONDECYT 1030785.

Table 7
The variables that are not significantly correlated with project outcomes.

Sponsor	Project manager given full authority The project began with a committed champion The commitment lasted right through the project The sponsor was involved with decisions Other stakeholders committed and involved Senior management impacted the project
Customer and users	Customers/users involved in schedule estimates Problems were caused by large by the number of customers/users involved Were you affected by large numbers of users
Requirements	Requirements were gathered using a particular method The requirements were complete and accurate Did the scope change during the project
Estimation and scheduling	The project had a schedule
Project manager	There was a project manager Was project manager changed during project Was everyone in project treated equally or did project manager play favorites Project manager's background Years of experience of the project manager Was project manager experienced in application area Was project manager able to pitch and help if needed
Development process	How frequently were meetings held on the average The project was over organized Other projects negatively impacted this project What was the effect of the schedule on the development process
Development team	Aggressive schedule affected motivation Number of consultants/contractors The consultants/contractors reported to the project manager The project manager had the authority to choose team members Did all the key people stay throughout the project

Appendix A

See Table 7.

References

[1] S. Abe, O. Mizuno, T. Kikuno, N. Kikuchi, M. Hirayama, Estimation of project success using Bayesian classifier, in: Proceeding of the 28th International Conference on Software Engineering – ICSE '06, vol. 4, 2006, p. 600.
[2] D. Baccarini, The logical framework method for defining project success, Project Management Journal 30 (December) (1999) 25–32.
[3] E.M. Bennatan, On Time, Within Budget, John Wiley & Sons, Inc., New York, NY, USA, 2000.
[4] B.W. Boehm, Software Engineering Economics, Prentice Hall, Englewood Cliffs, 1981.
[5] A. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition 30 (7) (1997) 1145–1159.
[6] L.C. Briand, T. Langley, I. Wieczorek, A replicated assessment and comparison of common software cost modeling techniques, in: Proceedings of the 22nd International Conference on Software Engineering, New York, New York, USA, ACM, New York, NY, USA, 2000, pp. 377–386.
[7] J.C. Brito, E. Crespo, O. Paulo, Modelling wildlife distributions: logistic multiple regression vs. overlap analysis, Ecography 22 (3) (1999) 251–260.
[8] F.P. Brooks, The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley, USA, 1975.
[9] N. Cerpa, J.M. Verner, Why did your project fail?, Communications of the ACM 52 (12) (2009) 130
[10] N. Charette, Why software fails, IEEE Spectrum 42 (9) (2005) 36–43.
[11] M. Cheng, Y. Wu, Dynamic prediction of project success using evolutionary support vector machine inference model, in: Proceedings of the 25th International Symposium on Automation and Robotics in Construction, 2008.
[12] F.D. Davis, Perceived usefulness, perceived ease of use and user acceptance of information technology, MIS Quarterly 13 (1989) 319–339.
[13] E.R. DeLong, D.M. Delong, D.L. Clarke-Pearson, Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach, Biometrics 44 (3) (1988) 837–845.
[14] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (8) (2006) 861–874.
[15] E.J. Garrity, G.L. Saunders, Introduction to information systems success measurement, in: E.J. Garrity, G.L. Saunders (Eds.), Information System Success Measurement, Idea Publishing Group, 1998, pp. 1–12.
[16] D. Gefen, It is not enough to be responsive: the role of cooperative intentions in MRP II adoption, The DATA BASE for Advances in Information Systems 31 (2000) 65–79.
[17] D. Gefen, M. Keil, The impact of developer responsiveness on perceptions of usefulness and ease of use: an extension of the technology acceptance model, The DATA BASE for Advances in Information Systems 29 (1998) 35–49.
[18] D. Gefen, D. Straub, The relative importance of perceived ease-of-use in IS adoption: a study of e-commerce adoption, JAIS 1 (2000) 1–30.
[19] R.L. Glass, Software Runaways: Monumental Software Disasters, Prentice-Hall, 1997.
[20] R.L. Glass, Evolving a New Theory of Project Success, Communications of the ACM 42 (November) (1999).
[21] N. Hagerty, Understanding the link between IT project manager skills and project success, in: Proceedings of SIGCPR Conference, Evanston, IL, 2000, pp. 192–195.
[22] T. Hoffman, Study: 85% of IT departments fail to meet business needs, Computerworld 33 (1999) 24.
[23] M. Ishman, Measuring information systems success at the individual level in cross-cultural environments, in: E.J. Garrity, G.L. Saunders (Eds.), Information System Success Measurement, Idea Publishing Group, 1998.
[24] C. Jones, Patterns of Large Systems Failure and Success, IEE Computer 28 (1995) 86–87.
[25] M. Jørgensen, K. Moløkken-Østvold, How large are software cost overruns? A review of the 1994 CHAOS report, Information and Software Technology 48 (2006) 297–301.
[26] S.K. Kachigan, Statistical Analysis: An Interdisciplinary Introduction to Univariate & Multivariate Methods, Radius Press, New York, 1982.
[27] C.F. Kemerer, An empirical validation of software cost estimation models, Communications of the ACM 30 (5) (1987) 416–429.
[28] B.A. Kitchenham, N.R. Taylor, Software cost models, ICL Technical Journal 4 (1984) 73–102.
[29] T.M. Koshgoftaar, E.B. Allen, J. Deng, Controlling overfitting in software quality models: experiments with regression trees and classification, in: 7th International Software Metrics Symposium, 2001, pp. 190–198.
[30] E.L. Lehmann, H.J. M D'Abrera, Nonparametrics: Statistical Methods based on Ranks, Prentice-Hall, 1998.
[31] S. Lessmann, B. Baesens, C. Mues, S. Pietsch, Benchmarking classification models for software defect prediction: a proposed framework and novel findings, IEEE Transactions on Software Engineering 34 (4) (2008) 485–496.
[32] K.R. Linberg, Software developer perceptions about software project failure: a case study, Journal of Systems and Software 49 (1999) 177–192.
[33] S. McConnell, Rapid Development, Microsoft Press, Redmond, Washington, 1996.
[34] K. Moløkken-Østvold, M. Jørgensen, S.S. Tanilkan, H. Gallis, A. Lien, S. Hove, A survey on software estimation in the Norwegian industry, in: Proceedings 10th International Symposium on Software Metrics, IEEE Computer Society, 2004, pp. 208–219.
[35] M. Paulk, B. Curtis, M. Chrissis, C. Webster, Capability Maturity Model for Software, Technical Report, Software Engineering Institute, Carnegie Mellon, 1993.
[36] J. Pereira, N. Cerpa, M. Rivas, Factores de éxito en proyectos de desarrollo de software: análisis de la industria chilena del software, in: Workshop de Ingeniería de Software, Arica, 2004.
[37] J. Pereira, N. Cerpa, M. Rivas, Risk factors in software development projects: analysis of the Chilean software industry, in: 1st Experimental Software Engineering Latin American Workshop, Brasilia, October 2004.
[38] J. Pereira, N. Cerpa, J.M. Verner, M. Rivas, J.D. Procaccino, What do software practitioners really think about project success: a cross-cultural comparison, Journal of Systems and Software 81 (2008) 897–907.
[39] J.K. Pinto, S.J. Mandel, The causes of project failure, IEEE Transactions on Engineering Management 34 (November) (1990).
[40] J.K. Pinto, D.P. Slevin, Project success: definitions and measurement techniques, Project Management Journal 19 (1988) 67–72.
[41] R. Pressman, Fear of trying: the plight of rookie project managers, IEEE Software 15 (1998) 50–51, 54.
[42] J.D. Procaccino, J.M. Verner, Early risk factors for software development, in: Proceedings of the 12th European Software Control and Metrics Conference, April 2001, pp. 107–116.
[43] J.D. Procaccino, J.M. Verner, Software practicioner's perception of project success: a pilot study, International Journal of Computers, the Internet, & Management 10 (2002).
[44] J.D. Procaccino, J.M. Verner, S.P. Overmyer, M.E. Darter, Case study: factors for early prediction of software development success, Information and Software Technology 44 (2001) 53–62.
[45] J.D. Procaccino, J.M. Verner, K.M. Shelfer, D. Gefen, What do software practitioners really think about project success: an exploratory study, Journal of Systems and Software 78 (2005) 194–203.
[46] L. Schoonhoven, D.E. Grobbee, M.T. Bousema, E. Buskens, Predicting pressure ulcers: cases missed using a new clinical prediction rule, Journal of Advanced Nursing 49 (1) (2005) 16–22.

[47] D. Smite, Project outcome predictions: risk barometer based on historical data, in: International Conference on Global Software Engineering (Icgse) (ICGSE 2007), August 2007, pp. 103–112.
[48] Q. Song, M. Sheppard, Missing data imputation techniques, International Journal of Business Intelligence and Data Mining 2 (2007) 261–291.
[49] The Standish Group, CHAOS, Technical Report, 1994.
[50] S. Tobias, J. Carlson, Brief report: Bartlett's test of sphericity and chance findings in factor analysis, Multivariate Behavioral Research 4 (3) (1969) 375–377.
[51] J.M. Verner, J. Sampson, N. Cerpa, What factors lead to software project failure?, in: Research Challenges in Information Science, Marrakech, Morroco, 2008, pp. 71–80.
[52] J.M. Verner, N. Cerpa, Australian Software Development: What Software Project Management Practices Lead to Success?, in: IEEE Proceedings of ASWEC, Brisbane, Australia, 2005, pp. 70–77.
[53] J.M. Verner, K. Cox, S. Bleistein, Predicting good requirements for in-house development projects, in: ACM International Symposium on Empirical Software Engineering, Rio de Janeiro, Brazil, 2006, pp. 154–163.
[54] J.M. Verner, K. Cox, S. Bleistein, N. Cerpa, Requirements engineering and software project success: an industrial survey in Australia and the US, Australasian Journal of Information Systems 13 (2005) 225–238.
[55] J.M. Verner, W.M. Evanco, The effect of cost and effort estimation on project success, in: Proceedings of ICSSEA 2003, vol. 4, 2003, pp. 1–10.
[56] J.M. Verner, W.M. Evanco, In-house software development: what software project management practices lead to success?, IEEE Software 22 (2005) 86–93
[57] J.M. Verner, W.M. Evanco, N. Cerpa, State of the practice: How important is effort estimation to software development success?, Information and Software Technology 49 (2007) 181–193
[58] J.M. Verner, S.P. Overmyer, K.W. McCain, In the 25 years since The Mythical Man-Month what have we learned about project management?, Information and Software Technology 41 (1999) 1021–1026
[59] Y. Wang, Prediction of Success in Open Source Software Development, Master's Thesis, University of California, Davis, California, 2007.
[60] J. Wateridge, IT projects: a basis for success, International Journal of Project Management 13 (1995).
[61] J. Wateridge, How can IS/IT projects be measured for success?, International Journal of Project Management 16 (1998)
[62] R. Weber, W.M. Evanco, M. Waller, J.M. Verner, Identifying critical factors in case-based prediction, in: V. Barr, Z. Markov (Eds.), Proceedings of the 17th Annual Conference of the International Florida Artificial Intelligence Research Society, AAAI Press, 2004, pp. 207–212.
[63] R. Weber, M. Waller, J.M. Verner, W.M. Evanco, Predicting software development project outcomes, in: D. Bridge, K. Ashley (Eds.), Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR), Springer-Verlag, 2003, pp. 595–609.
[64] C. Wohlin, A.A. Andrews, Evaluation of three methods to predict project success: a case study, in: Proceedings of International Conference on Product Focused Software Process Improvement, Springer, 2005.
[65] C. Wohlin, P. Runeson, M. Host, M. Ohlsson, B. Regnell, A. Wesslen, Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers, London, 2000.
[66] C. Wohlin, A. von Mayrhauser, Assessing project success using subjective evaluation factors, Journal of Software Quality (August) (2000).
[67] J. Woodroof, G.M. Kasper, A conceptual development of process and outcome user satisfaction, in: E.J. Garrity, G.L. Saunders (Eds.), Information System Success Measurement, Idea Publishing Group, 1998, pp. 122–132.