The Journal of Systems and Software 112 (2016) 48–64

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Evaluating different families of prediction methods for estimating software project outcomes

Narciso Cerpa[a,*], Matthew Bardeen[a], César A. Astudillo[a], June Verner[b]

[a] Universidad de Talca, Talca, Chile
[b] University of Oulu, Finland

ARTICLE INFO

Article history:
Received 12 May 2014
Revised 7 August 2015
Accepted 3 October 2015
Available online 23 October 2015

Keywords:
Project outcome predictors
Classification techniques
AUC analysis

ABSTRACT

Software has been developed since the 1960s but the success rate of development projects is still low. Classification models have been used to predict defects and effort estimation, but little work has been done to predict the outcome of these projects. Previous research shows that it is possible to predict outcome using classifiers based on key variables during development, but it is not clear which techniques provide more accurate predictions. We benchmark classifiers from different families to determine the outcome of a software project and identify variables that influence it. A survey-based empirical investigation was used to examine variables contributing to project outcome. Classification models were built and tested to identify the best classifiers for this data by comparing their AUC values. We reduce the dimensionality of the data with Information Gain and build models with the same techniques. We use Information Gain and classification techniques to identify key attributes and their relative importance. We find that four classification techniques provide good results for survey data, regardless of dimensionality reduction. We conclude that Random Forest is the most appropriate technique for predicting project outcome. We identified key attributes which are related to communication, estimation, and process review.

1. Introduction

The development of software is a complex and expensive process that has been researched and implemented since the early 60s, but we still have not learned enough to substantially increase its success rate (Cerpa and Verner, 2009). Charette estimated that software project failures cost between $25 and $75 billion to the U.S.A.'s economy. This cost excludes projects exceeding their budget or finishing late, even though this is usual in most cases (Charette, 2005). Previous studies suggest various failure rates for software development projects reaching up to 85% (Glass, 1997; Hoffman, 1999; Jørgensen and Moløkken-Østvold, 2006). The literature also suggests that software projects are usually affected by many problems during their development such as; poor project management, cost and schedule overruns, poor quality software and under-motivated developers (Bennatan, 2000; Brooks, 1975; El Emam and Koru, 2008; Verner et al., 2007). However, some of these problems are post-process – by the time they are found the software project has already finished and is already a failure. Current research has placed emphasis on identifying those attributes or variables that may serve as predictors or causal agents for software project outcome (Cerpa et al., 2010; Reyes et al., 2011). Failures in software development and poor software quality lead to a lack of credibility and communication problems among developers, senior management, customers, and users, making the software development process and software implementation even more difficult (Gefen, 2000; Gefen and Straub, 2000). During the development process, the project manager (PM) and the development team must usually deal with various pressures from project stakeholders (i.e., upper level management, marketing, accounting, customers, and users) that impact both the cost and the quality of the software produced (Boehm, 1981). Some of these pressures are tight schedules and late changes to requirements usually caused by a poor elicitation process for requirements and/or changing business needs.

Some researchers have reported on software project development failures from the customer/user perspectives (e.g., Garrity and Saunders, 1998; Ishman, 1998; Woodroof and Kasper, 1998), however, it is also important to identify and recognize the effects of failures on the software development staff. Developers usually suffer of long hours of unpaid overtime, loss of motivation, and stress due to late projects. This leads to high staff turnover and its associated costs. Despite of having many guidelines for successful software development

* Corresponding author. Tel.: +56 75 2201710; fax: +56 75 2201713.
E-mail addresses: ncerpa@utalca.cl (N. Cerpa), mbardeen@utalca.cl (M. Bardeen), castudillo@utalca.cl (C.A. Astudillo), june.verner@gmail.com (J. Verner).

http://dx.doi.org/10.1016/j.jss.2015.10.011
0164-1212/

(e.g., Boehm, 1981; McConnell, 1996; Paulk et al., 1993; Pressman, 1998), few project post-mortems are published and shared (Verner et al., 1999), and little understanding is gained from the results of past projects within and between organizations. It is critical to the success of a software project to have a project manager with the capacity to understand the consequences of actions taken during the software development process and the effect that decisions have on the development outcome. Project managers frequently ignore some key development practices and do not understand early warning signs of project failure. Some project failures are predictable and avoidable, although it may not be always possible for developers and managers to identify which variables are more important to success and failure to take action in time.

The software engineering literature provides several definitions of software project success or failure, as well as definitions and descriptions of various factors influencing software project outcome. Some of these factors are, for example: 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 (Charette, 2005; Davis, 1989; Garrity and Saunders, 1998; Gefen and Keil, 1998; Jørgensen and Moløkken-Østvold, 2006; Linberg, 1999; Pereira et al., 2004; Procaccino et al., 2005; Standish Group, 1994; Verner et al., 2008, 1999; Wateridge, 1995; Weber et al., 2003; Wohlin and von Mayrhauser, 2000; Woodroof and Kasper, 1998). There are innumerable variables or attributes that influence these factors.

Therefore, it is necessary to first identify the factors that are more important in ensuring project success, and then to identify the variables that have the most influence over the identified factors. Project managers that have this information in hand can quickly identify problem areas in their project and work to solve them. Determining which of these variables are more important has demonstrated to be a difficult task. Fortunately, current developments in the field of Statistics and Machine Learning have facilitated this process.

Since the software practitioner perspective is extremely valuable to the discipline of software engineering, and in particular to the management of the software development process, in this paper we describe a survey-based empirical investigation into factors that contribute to project outcome from the practitioner perspective. We are aware that there are many objective definitions available for whether a project is a success or failure (Baccarini, 1999; Pinto and Slevin, 1988; Procaccino and Verner, 2002). The traditional definition of success includes project management success (i.e., cost, time, and quality), and project product success (Baccarini, 1999; Pinto and Slevin, 1988). Other definitions include issues of importance to the stakeholder. For example, users measures of software project success are: meeting user requirements, user satisfaction and meeting budget (Wateridge, 1995). Project managers perceived a successful project as one in which they have met the user requirements, provide a sense of quality and personal achievement (Procaccino and Verner, 2006). For top management a successful software project is the one that supports strategic objectives (Bleistein et al., 2005). However for this study, we choose to focus on the perception of the developers since that is the information most readily available to a project manager during a project.

A number of different approaches have been used to develop models to predict the outcome of software projects (Abe et al., 2006; Cerpa et al., 2010; Cheng and Wu, 2008; Mizuno et al., 2004; Reyes et al., 2011; Smite, 2007; Takagi et al., 2005; Wang, 2007). For this analysis, we focus our attention on different families of classifiers for building prediction models. An analysis of the receiver operating characteristics (ROC) graph is used to compare the predictive capacity of the various models (Bradley, 1997; Lessmann et al., 2008). The best suited classifier will be considered to provide information to project managers regarding the chances their software project has for success or failure.

Therefore, our research questions are:

- *RQ1:* What is/are the most appropriate classifier(s) for estimating software project outcome with this collection of data?
- *RQ2:* What are the key attributes or variables that impact software project success?

We feel this work is important for both project managers and researchers. For project managers, it will provide a valuable tool for identifying the potential result of a project before it is too late, allowing them to focus their efforts to ensure a successful project. For researchers, it may enable them to target specific factors of the software development process which should be considered in their future investigations.

To provide a brief overview of the organization of the paper: in the next section we discuss the we discuss the previous work in software project success prediction and in Section 3 we describe the data was used to build the models including some project demographics. In subsequent sections describe the methodology used and show the results of the data analysis. In the final sections we provide a discussion of the results and the conclusions.

2. Related work

Most previous research on the prediction of software project outcome has focused on the identification of the software development characteristics or practices which contribute to project success (Abe et al., 2006; Cerpa et al., 2010) or in the probability of occurrence of threats (Smite, 2007). There are characteristics or practices that should be used by all software projects in order to be successful, but others may have different effects depending on the context of the project (Egorova et al., 2010). Some researchers suggest that it is essential to identify the characteristics of successful projects in order to determine similarities (Cerpa et al., 2010; Procaccino et al., 2001; Wohlin and Andrews, 2005). Hence, in order to have reliable predictions of software project outcome we need to have an appropriate method to evaluate characteristics and identify those projects which are similar to each other. To increase the probability of success in future software development projects, we must learn from past project experiences. A basic and traditional definition of software project success is related to having development costs and time within the estimates, sufficient functionality and satisfactory software quality for the client. However, this definition is post-hoc; by the time these problems are known the software project has already finished and it is considered a failure. Instead current research has focused on identifying those attributes that serve as predictors or causal agents for project failure or success (Cerpa et al., 2010; Reyes et al., 2011).

Researchers have used these attributes in two different ways to predict software project outcome. Initially, they had a list of criteria for measuring success. This list of criteria was based on characteristics of past successful projects and each criterion has some value associated with it, called success points. Those criteria that apply to the project are selected and all their success points are summed up. If the total is greater than a predefined threshold, the project is predicted as successful (Standish Group, 1994; McConnell, 1996). More recently, machine learning techniques have been used to learn from past projects and build prediction models (Abe et al., 2006; Mizuno et al., 2004; Reyes et al., 2011; Wang, 2007; Cheng and Wu, 2008; Smite, 2007; Takagi et al., 2005).

Abe et al. collected 29 metrics classified into several categories such as development process, project management, company organization, human factors, and external factors to build a model using a Bayesian classifier to predict software project outcome (Abe et al., 2006). Mizuno et al. also used a Bayesian classifier to predict runaway