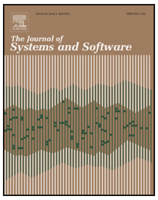
The Journal of Systems and Software 92 (2014) 157–169

Contents lists available at ScienceDirect

The Journal of Systems and Software

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



Failure factors of small software projects at a global outsourcing marketplace

Magne Jørgensen*

Simula Research Laboratory and University of Oslo, P.O. Box 134, NO-1325 Lysaker, Norway

ARTICLE INFO

Article history:
Received 2 July 2013
Received in revised form 22 January 2014
Accepted 23 January 2014
Available online 3 February 2014

Keywords:
Outsourcing
Project failures
Risk management

ABSTRACT

The presented study aims at a better understanding of when and why small-scale software projects at a global outsourcing marketplace fail. The analysis is based on a data set of 785,325 projects/tasks completed at vWorker.com. A binary logistic regression model relying solely on information known at the time of a project's start-up correctly predicted 74% of the project failures and 67% of the non-failures. The model-predicted failure probability corresponded well with the actual frequencies of failures for most levels of failure risk. The model suggests that the factors connected to the strongest reduction in the risk of failure are related to previous collaboration between the client and the provider and a low failure rate of previous projects completed by the provider. We found the characteristics of the client to be almost as important as those of the provider in explaining project failures and that the risk of project failure increased with an increased client emphasis on low price and with an increased project size. The identified relationships seem to be reasonable stable across the studied project size categories.

1. Introduction

A great deal of resources are spent on software projects that fail to deliver useful functionality. For example, the proportion of started and then cancelled projects, sometimes termed "aborted" or "abandoned" projects, is reported to be 9% (Sauer et al., 2007), 11% (Tichy and Bascom, 2008), and 11.5% (El Emam and Koru, 2008). Several non-peer reviewed reports claim a much higher proportion of cancelled software projects, but may be less reliable or less representative of the population of software projects. The frequently cited Standish Group Chaos Report (1995), for example, claims that as many as 31% of all software projects get cancelled. The low reliability of that report is discussed in (Jørgensen and Moløkken-Østvold, 2006; Eveleens and Verhoef, 2010). While the cancellation rates described in the Standish Group Chaos Reports and similar non-peer reviewed surveys are likely to be exaggerated, there is no doubt that the proportion of cancelled projects is substantial.

The definition of a failed project in software surveys typically includes both cancelled projects and projects completed with a very poor product or process quality. Consequently, the reported failure rates appear higher than the corresponding cancellation rates. Exactly how much higher depends on the failure criteria used. For example, El Emam and Koru (2008) categorized a project as having failed if it received a score of "poor" or "fair" in four out of five of the following performance criteria: user satisfaction, ability to meet budget targets, ability to meet schedule targets, product quality and staff productivity. This definition led to a failure rate of more than twice the cancellation rate for the same set of projects, i.e., a failure rate of 26% for the data set reporting a cancellation rate of 11.5%. Defining every project that does not deliver the specified product, is over budget, or is not on time as a failure, as is the case in several reports, typically amounts to 50–80% of all software projects being failures. For an overview of software failure surveys see (Hashmi and Stevrin, 2009).

The challenge of defining project failures meaningfully is further illustrated in (Boehm, 2000), where Barry Boehm makes the reasonable claim that not all cancellations should be considered to be failures. There may, for example, be good reasons for cancelling a well-managed project if the project's original assumptions of usefulness are no longer valid. In that case, the failure would clearly be to continue a project that is no longer needed instead of cancelling it. A similar problem may occur when a project is interpreted as a failure because it delivers something other than what was originally specified or expected. There are development processes, e.g., agile methods, in which requirements are meant to evolve as part of the learning process and, clearly, it would be meaningless to define the learning process leading to change in requirements as indicating a failure. It may also be important to separate a project failure from a product failure, see for example (Baccarini, 1999). Finally, there may be differences in the failure perspectives of different project stakeholders, which also lead to different interpretations of whether a project has failed or not (Agarwal and Rathod, 2006).

* Tel.: +47 924 333 55.
E-mail address: magnej@simula.no

0164-1212/$ – see front matter © 2014 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2014.01.034

In spite of the problems with providing a commonly accepted definition of project failures, there is little doubt that there are many situations where resources can be saved by reducing the number of projects that do not deliver anything, deliver a product much later than expected, or deliver a product that is not useful at all for the client. Not only is the direct waste of project resources likely to be substantial, but also the indirect waste such as lost business opportunities.

The importance of reducing the waste of resources on project failures motivates the high number of studies concerning the reasons for project failures, and methods to reduce failure rates. This includes several studies surveying what the stakeholders, such as the software developers, project managers, clients and users, perceive are the main failure and success factors of software projects. For lists of such perceived factors see for example (Linberg, 1999; Schmidt et al., 2001; Charette, 2005; Fabriek et al., 2008; Verner et al., 2008; Al-Ahmad et al., 2009).

The study presented in this paper differs, to our knowledge, from previous studies on project failures on the following characteristics: (i) It focuses on the effects of the potential failure factors known at the time of the project start-up, (ii) It uses only observational data and not perceived project failures, and (iii) It focuses on the, typically very small, projects conducted at a global outsourcing marketplace. While there are other studies that use project data to predict failures based on project characteristics (Wohlin and Andrews, 2001; Mizuno et al., 2004; Cerpa et al., 2010; Egorova et al., 2010), these studies do not restrict the prediction of the risk of project failure to observable variables known at the project start-up. The use of the variables known at start-up makes the model more useful for practical settings. By the time it is possible to know that the project plan is poor or the problem complexity is higher than expected, it may already be too late to take the proper action to avoid project problems and reduce the risk of failure. Prediction models using variables known at the project start-up may, amongst others, allow a client to get input related to an expected increase in the failure risk when emphasizing a lower price rather than higher skills when selecting a provider for the project. On the other hand, restricting a prediction model to variables observed at the project start-up means that the model will not be as accurate as models that include project failure variables known at much later stages in the development process.

The goal of our study is not only to predict the risk of project failure, in the context of a global outsourcing marketplace, at the time of project start-up, but also to better understand when we have a context that is more likely to result in project failure. This improved understanding may be used to avoid situations with a high risk of failure, or if that is not possible, to give risk reducing actions a high priority in high-risk situations.

The remaining part of this paper is organized as follows: Section 2 describes the project database we used to build the model of project failure. Section 3 defines the model variables, describes the process of building the model, evaluates the accuracy of the prediction model and discusses the factors connected with higher or lower risks of project failure. Section 4 contains a discussion of the limitations of the analyses. Section 5 concludes.

2. The project database

The dataset we used to develop the prediction model consists of 785,325 small-scale software projects. The clients and providers of these projects have been using the services of vWorker.com (now merged with freelancer.com), which is a web-based global marketplace that connects software clients and providers. The providers are typically single software developers or smaller outsourcing companies located in low-cost countries, but include developers and companies from high-cost countries as well. There are also a few larger companies that use this marketplace, in addition to other channels, to find work for their employees. The services offered by the vWorker.com marketplace include:

- The means for clients to search for and invite project bids from providers with appropriate skills, e.g., Java, php, and SQL.
- Support for providers to place bids on a project.
- Arrangements that ensure that the provider is paid when the work is completed, and that the client does not have to pay if the work is of too low quality.
- Processes for managing disagreements/negotiations between the clients and the providers regarding payments or quality of work (arbitration processes).
- Skill tests of the providers.
- Evaluations of provider performances from previous projects through the presentation of aggregated and project specific information about client satisfaction and project cancellations.
- Evaluations of client performances from previous projects through the presentation of aggregated and project specific information about providers' satisfaction and project cancellations.

Typically, the providers and the clients never physically meet and conduct all of their communication through the functionality provided by vWorker.com, or other internet-based communication.

The characteristics of the data set of the vWorker.com projects that we used in this study include:

- Project data registered between May 2001 and October 2012.
- Number of projects: 785,325.
- Number of bids placed: 4,791,067.
- Mean number of bids per project: 6.1.
- Proportion of cancelled projects: 11.1%.
- Proportion of projects that were either cancelled or the provider received a client satisfaction rating of "poor" or worse: 14.0% (these are the projects we categorized as failed, see Section 3). Notice that a cancelled project either had no satisfaction score or a satisfaction score of "poor" or worse.
- Average provider pass rate on skill tests: 64%.
- Number of different provider nationalities: 187.
- The ten largest provider countries (sorted by decreasing number of projects): India, US, Romania, Pakistan, UK, Russia, Ukraine, Canada, Bangladesh, and the Philippines. Some of the providers located in high-cost countries seemed to use developers from low-cost countries, i.e., the provider country is not always as it seems from the vWorker.com presentation of a country.
- Number of different client nationalities: 177.
- The ten largest client countries (sorted by decreasing number of projects): US (with more than 50% of the projects), UK, Australia, Canada, India, Germany, the Netherlands, Israel, Sweden, and France.
- Proportion of projects where the client and provider are located in different countries: 90%.
- Proportion of projects where the client and provider have collaborated previously using vWorker.com: 43%.
- Price range of projects: 1–30,000 USD, with a mean of 146 USD, i.e., most projects are very small. Nearly all projects are based on a fixed-price contract between the client and the provider.

As can be seen from the above information, this data set includes a high number of projects, but the projects are, on average, very small. Based on personal experience with several of the companies on vWorker.com, we experience that a typical rate per work-hour is 10–20 USD, which means that the average project or task effort is likely to be only about 10 work-hours. Where there is, we believe,