The Journal of Systems and Software 83 (2010) 2175–2187

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Software engineering projects may fail before they are started: Post-mortem analysis of five cancelled projects

Jarmo J. Ahonen[a,*], Paula Savolainen[a,b]

[a] School of Computing, University of Eastern Finland, P.O. Box 1627, 70211 Kuopio, Finland
[b] Lero — The Irish Software Engineering Research Centre, University of Limerick, Ireland

ARTICLE INFO

Article history:
Received 17 November 2009
Received in revised form 11 June 2010
Accepted 13 June 2010
Available online 19 June 2010

Keywords:
Software engineering
Project cancellation
Project failure
Post-mortem analysis
Customer
Supplier

ABSTRACT

Context: Software project cancellations are often caused by mistakes made during the project, and such cancellations make a strong economic impact. We analyzed five cancelled software engineering projects. One case was an internal product development project of a company that sells products to its customers. The other four cases were different software engineering projects, and outcomes of these projects were planned to be delivered to external customers.
Objective: This study reports a post-mortem analysis of five software engineering projects with the aim of providing more knowledge about the reasons for cancellation decisions and the causes behind those reasons.
Methods: The research method is case study. A method for a document-based post-mortem analysis was developed and post-mortem analysis was performed. All project documentation was available for analysis.
Results: The reasons for the cancellation decisions were well-known ones. In four cases of five, the outcome of the project was to be delivered to an external customer, but in these cases the causes of the cancellation reasons were not found from the normal project documentation. In these cases the cause of the cancellation originated in a phase before the start of the project and therefore the project was doomed before it was started.
Conclusion: It is reasonable to suggest that a remarkable portion of project cancellations are due to mistakes made before the project is started in the case of contract-based software engineering projects.

1. Introduction

A cancelled software project is usually an unwanted situation which means loss of economic resources, despair, and embarrassment. A large cancelled software project may ruin careers and even exterminate companies. Software development history has numerous examples of project cancellations as well as consequences of software project cancellations. Unfortunately, it is likely that there are no easy means to avoid or reduce software project cancellations.

The economic impact of project cancellations is difficult to measure in any meaningful way, and even the percentage of cancelled projects is not clear (Glass, 2005). A project cancellation, sometimes called an abandonment, is a situation in which practically nothing, or even nothing at all, is salvaged from the project. A project cancellation is something that nobody wants to flaunt, and therefore getting an even reasonably accurate estimate of how many projects are cancelled is next to impossible. Some estimates have, however, been presented. For example, Charette (2005) estimated that 5–15% of all large-scale software projects are cancelled in the USA, and that the total yearly cost of cancellations may be as much as US$75 billion.

Assuming that Charette's estimates are correct, the number of cancellations is daunting and their economic impact significant. The motivation of this paper is to investigate ways to reduce the number of cancellations. The basic approach to achieve that goal is obvious: if we do not conduct post-mortem reviews, we are unlikely to understand why our projects fail (Cerpa and Verner, 2009). Analysis of cancelled projects enables us to modify and improve the software development process (Reel, 1999) and to identify critical decision points before and during the project execution.

The way to avoid past mistakes is by understanding what went wrong and how it could have been avoided. Good answers to these questions regarding cancelled software projects are not, however, generally available. This makes general advancement of our soft-

* Corresponding author. Tel.: +358 405678018.
E-mail addresses: jarmo.ahonen@uef.fi (J.J. Ahonen), paula.m.savolainen@uef.fi (P. Savolainen).

0164-1212/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2010.06.023

ware engineering project knowledge much more difficult. There are at least two reasons for the unavailability: the small percentage of projects that go through a post-mortem analysis (Glass, 2002), and the general unavailability of knowledge of cancelled software projects.

The first problem, the small percentage of projects analyzed, is not restricted to the software engineering field — some surveys have revealed that 80% of all R&D projects are not reviewed at all after completion (von Zedtwitz, 2002). In that sense, software engineers, whether practitioners or researchers, are in the same situation as other professions. The fact that the situation is not very good in other professions does not, however, give us any excuse not to perform proper post-mortem analysis.

The small percentage of analyzed projects can be improved by analysing more projects, but the second problem, the general unavailability of knowledge of cancelled projects, is a much more difficult issue to solve. It is safe to assume that the names of cancelled projects that are repeated in many articles include those cases that have been either too massive to be hidden or that have been public in some legal sense. A good example of research that uses well-known cancelled projects, some of which have been discussed in Glass (1999), is the one performed by Chua (2009). Most of the new cases that appear in newspapers or in scientific journals seem to fall into the same category of massive or public cancellations. Other cancellations are concealed inside the organizations, which is very understandable because neither the organizations nor the individuals involved want the details of those projects to appear in any media. The tendency to hide cancelled projects is intensified in those cases in which the supplier and the customer are separate companies.

Although some projects may be cancelled for reasons related to changes in the business environment or some other outside reasons, many cancelled projects would have succeeded if mistakes had not been made before or during the project execution. Those projects are very interesting because understanding why the cancellation took place would help us to avoid similar mistakes in the future. That understanding is especially important in order to reduce the unnecessary waste of resources.

It should be noted, however, that we do not assume that no project should fail. Failure is an essential part of high-risk projects, especially in the case of R&D projects. Although some types of projects are much more likely to fail than other types, unnecessary failures should be avoided if possible.

In order to achieve better understanding of the mistakes that caused project cancellations, we analyzed five cancelled software projects which should have succeeded. One case was an internal product development project and in the other cases the customer and the supplier were separate companies. In those cases the supplier had made an agreement with the customer for a specific project and agreed to deliver the project outcome to the customer. The aim of the study was to find out why these five projects suffered cancellation. This knowledge will help us to understand software projects better and relieve the impact of project cancellations.

The study reported in this paper was possible because of the unusually rich sets of project data that each case provided for research purposes. Each case allowed us to cut into the body of the deceased project, the body being the paperwork that includes all types of official and unofficial documents related to the project. In the analysis we looked into what happened in the projects, and especially into the actual problems encountered during the projects. The analyzed cases are described in Section 3 and the analysis methodology is presented in Section 4.

In Section 5 we discuss the findings of the post-mortem analysis for each case. The reasons for the cancellations did not provide any real surprises. However, it was surprising that in four cases we were not able to find the cause of the cancellation reason from the documentation which is normally regarded as project documentation. Our inability to find the causes from the project documentation itself led us to extend the analysis to all available documentation. The results of that analysis are presented in Section 6. They show that four of the projects were doomed even before they were started, and that even the most valiant efforts of all stakeholders might not have been able to salvage the project.

Section 7 presents a brief discussion of the possibility of avoiding project cancellations in the analyzed cases. The validity of results is discussed in Section 8: these threats to validity are noteworthy but not serious regarding the results of the study.

The final section, Section 9, ends the article with the conclusion that in many of the cancelled projects the actual cause of the cancellation reason may be hidden somewhere in the actions that took place before the project started.

2. Related research and post-mortem methods

This section discusses related research, both research on post-mortem analysis methods and on the use of post-mortem analysis in order to learn from previous experiences of cancelled projects. However, such learning is quite difficult because very few results of post-mortems of cancelled projects have been reported.

The concept of a post-mortem analysis is not very straightforward because it seems to be a fairly versatile tool. It can be used for the analysis of the end product of the project, the program, as in the research reported by Zhang and Iyer (2007), or for helping software process simulation in order to improve software project estimation (Aguilar-Ruiz et al., 2001). In this article, post-mortem analysis means an analysis performed in order to achieve understanding of a project that has already ended.

There seems to be a general agreement about the necessity of post-mortems (Reel, 1999; Glass, 2001; Birk et al., 2002; Ewusi-Mensah, 2003; Verner and Evanco, 2005), but still they are quite seldom performed (Verner and Evanco, 2005). One of the reasons for this may be that learning from past projects is important but it is not that easy to learn the "hard", non-intuitive lessons (Williams, 2004). Moreover, concern about frank analysis especially of failure creates a natural disincentive within the organization to conduct a post-mortem; it also creates apprehension in the individual preparing to take part in ones that are held (Collier et al., 1996). But post-mortems are especially important if we are to learn from problems encountered during a project (Williams, 2004; Verner and Evanco, 2005). If one does not take time to find out what happened during a failed project, for example, then one is doomed to repeat the same mistakes (Reel, 1999; Ewusi-Mensah, 2003; Verner and Evanco, 2005). It is, however, the case that most cancelled projects are not analyzed at all (von Zedtwitz, 2002).

In order to make people more willing to perform post-mortems, the post-mortem process should be well defined (Collier et al., 1996). Fortunately, there are some reasonably detailed descriptions of post-mortem process in e.g. Tiedeman (1990); Whitten (1995); Collier et al. (1996); Collison and Parcell (2001) and Birk et al. (2002). All these processes are somewhat different but they have the same general structure, which can be simplified into four phases:

1. Data collection, of which there are two basic variations. In both, data are collected from team members, and the variation lies in the utilization of project documentation. Some processes use it, whereas others do not. The data collection can be performed by interviews and questionnaires, or a combination of the two.
2. A workshop meeting, in which at least some the people who participated in the project are present. It can consist of different types of discussion or more formal analysis methods. During the