Available online at www.sciencedirect.com

ScienceDirect

International Journal of Project Management 32 (2014) 1073 – 1082

International Journal of Project Management

www.elsevier.com/locate/ijproman

Quality vs risk: An investigation of their relationship in software development projects

Lazaros Sarigiannidis [a,*], Prodromos D. Chatzoglou [b,1]

[a] Business Administration Dept., Kavala Institute of Technology, Agios Loukas, 65404 Kavala, Greece
[b] Production & Management Engineering Dept., Democritus University of Thrace, Library Building, Kimmeria, 67100 Xanthi, Greece

Received 23 December 2012; received in revised form 3 October 2013; accepted 12 November 2013
Available online 28 November 2013

Abstract

Quality, risk and successful software development projects are three concepts which appear to be indisputably intertwined with one another. The purpose of the present study is to investigate the relationship between people quality, process quality and risk in the context of software development projects of Greek companies. Project team members with different characteristics were used as key respondents. The final sample consisted of 112 projects from 63 companies. Empirical data were analysed using the structural equation modelling technique. The main results indicate a negative effect of people quality on project risk level. On the contrary, process quality appears to have a slightly limited effect, defining only the risk level associated with the project team. The results contribute in the existing literature underlining the importance of quality on the reduction of the project risk level, thus, creating the necessary background for new similar research attempts in the future.

Keywords: Software development; Project quality; Project risk; SEM

1. Introduction

Between now and 2025, the ability of companies and their products, systems and services to compete, adapt and survive will depend to a great and continually increasing extent on the software they will be using (Boehm, 2006). This software could provide them with competitive differentiation and quick adaptation to competitive changes for today's products and services. However, software success depends on software quality (Gorla and Lin, 2010), effectiveness and completeness (Nienaber and Cloete, 2003). Software quality is frequently determined by the software development process quality (Schwalbe, 2000), which is defined through specially developed control metrics (Jorgensen, 1999; Sahraoui et al., 2010; Subramanian et al., 2007).

Software development projects and the effects determining their outcomes have been the subjects of a continued research effort over the last thirty-five years (McLeod and MacDonell, 2011). Rothenberger et al. (2010) stated that software development process has no differences with other engineering disciplines: a managed approach is expected to lead to better quality of the product. Regardless of quality issues, software development projects are particularly difficult to be managed and this is because they incorporate a group of complex characteristics, which are not found in any other type of projects in the engineering or manufacturing field (areas which software development is mainly compared with). Human interaction, high level of complexity and product versatility are few of these characteristics (Jorgensen, 1999; Subramanian et al., 2007). Consequently, despite the fact that software has been effectively applied in a large range of areas, software development projects have a reputation for failure (Savolainen et al., 2012). Naturally, therefore, software development projects have been considered for many years as the ones with the highest risk.

* Corresponding author. Tel.: +30 2513 006162.
E-mail addresses: lsarigia@pme.duth.gr (L. Sarigiannidis), pchatzog@pme.duth.gr (P.D. Chatzoglou).
[1] Tel./fax: +30 25410 79344.

0263-7863/$36.00 © 2013 Elsevier Ltd. APM and IPMA. All rights reserved.
http://dx.doi.org/10.1016/j.ijproman.2013.11.001

The overcoming of the chronic problems that are found in project development, like budget overruns, delays in its completion and weakness to manage and respond to user requirements (McLeod and MacDonell, 2011), are not just desired, but a basic priority for an economy as well (Yang et al., 2009). Subsequently, the role of risk management in software development is considered especially important. Not surprisingly researchers, as well as practitioners, have expressed their interest in managing risk in software development projects (Barki et al., 1993; Boehm, 1991; de Bakker et al., 2010; McLeod and MacDonell, 2011; Nakatsu and Iacovou, 2009). Risk management is a forecasting, complete, systematic and based on the archetypes approach (Hall, 1998), which attempts to standardise the relationships involving risk that threatens the (usually subjective) success of the development process (Lee et al., 2009), with the use of an applicable sum of principles and practices (Addison and Vallabh, 2002).

The technological gap in Greece compared with other European countries (as measured by NRI) and the reluctance of several Greek companies-mainly SMEs to adopt new technologies, limit the potential market growth. Additionally, the software development industry in Greece does not seem to remain unaffected by the global economic crisis. Both companies and households are not willing to invest heavily in new technologies, before the end of the crisis. The impact on the software development market has been visible for about four years, as there is a direct dependence with the budgets of client companies. Thus, the current economic crisis in the EU has driven down budgets and formulates the growth rates at much lower levels. Furthermore, the highest percentage of family SMEs in the Greek economy, which does not have high requirements in software, and they are not aware of the prospects and the necessity of computer systems, lead to a restriction in the size of investment and the scope of new projects. The Greek software development projects seem to be very limited (in terms of budget, duration or employees) compared to those of the technologically developed countries (Greek National SME Observatory, 2011). Also the competition has been intensified with the entry of multinational companies in the Greek market in the last decade, which have higher funds, high costs for R & D and access to specialized workers, offering high quality products, and forming relatively quickly a wide deposit of important customers. Generally, the Greek software packages, although holding significant market shares, face stiff competition from the respective programs of foreign software houses. The latter have considerably strengthened their market shares, creating trends of market concentration, while several Greek SMEs, which constituted a large part of the industry, faced serious problems of competition.

In order to create a more clear and integrated picture about the software development process in Greece in relation to other countries it can be used some well-known methodologies such as the capability maturity model integration (CMMI) which classifies software organizations into five levels based on the sophistication of their engineering and management practices (Na et al., 2007). Unfortunately Greek organizations are not listed in the aforementioned taxonomy and this kind of comparison can't be used.

The aim of this study is the creation of a new research framework, which will arise through a thorough examination of the international literature. This framework will also be tested, using empirical data, in order to measure the significance of the conceptual factors related to project quality and risk.

2. Literature review

2.1. Risk dimensions

Barki et al. (1993) support that software project risks consist of interrelated dimensions and their measurement should not be done with the use of a one-dimensional scale, but, on the contrary, every dimension must be separately, theoretically and practically defined.

Despite the importance of studying risk through its dimensions, only a few researches employing them have been conducted. McFarlan (1981) found three main risk dimensions of the software development process and, more precisely, the size, the technological experience and the structure of the project. He proposed that project managers should develop a complete and integrated risk profile for a software project. Later, Boehm (1991) proposed a risk management framework where he included the factors of risk estimation and control, while he also made a list of the ten most important risks, based on his personal professional experience. In spite of all these, Boehm's list lacks some theoretical foundation (Huang and Han, 2008) and also, due to complexity and other factors (invisibility, harmonisation, flexibility), which characterise software projects, it ceased to have any diachronic value as years pasted (since it was conducted in 1991) (Hughes and Cotterell, 2006).

Barki et al. (1993) conducted an extensive review of studies (120 projects that had been carried out by 75 organizations) concerning software development risk, and proposed 35 parameters for assessing risks, which, in turn, were categorised in five dimensions: technological innovation, application size, expertise, application complexity and organizational environment. However, despite the fact that their study provided a quite useful and understandable tool for risk measuring, it also proved that the estimation scale was extremely complicated (Huang and Han, 2008; Wallace et al., 2004b).

Further, Heemstra and Kusters (1996), based on earlier studies and their professional experience, composed a list of 36 software risks, which later were grouped into 9 categories. Moynihan (1997), in cooperation with 14 Irish software application design specialists, developed a group of 21 risk related points. Later, Longstaff et al. (2000) proposed a framework, named hierarchically holographic modelling (HHM), and distinguished seven dimensions of system integration that involved 32 types of risk. Also, Cule et al. (2000) identified 4 risk dimensions according to their source of origin (labour, customer, environment, individuality), which involved 55 software risks. Then, they proposed a primary risk management strategy for each dimension separately.

Based on Boehm's (1991) risk list, Ropponen and Lyytinen (2000) developed a questionnaire which contained 6 risk dimensions for a research where 83 managers from 1100 software projects participated in. Sumner (2000), through structural