The Journal of Systems and Software 92 (2014) 115–127

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Factors that motivate software engineering teams: A four country empirical study

J.M. Verner[a,b], M.A. Babar[c], N. Cerpa[d,*], T. Hall[e], S. Beecham[f]

[a] University of Oulu, Finland
[b] Keele University, UK
[c] School of Computer Science, University of Adelaide, Australia
[d] Universidad de Talca, Chile
[e] Brunel University, UK
[f] LERO – The Irish Software Engineering Research Centre, Univ of Limerick, Ireland

ARTICLE INFO

Article history:
Received 22 November 2012
Received in revised form 6 December 2013
Accepted 3 January 2014
Available online 25 January 2014

Keywords:
Software development team motivation
Project success
Motivational factors

ABSTRACT

Motivation, although difficult to quantify, is considered to be the single largest factor in developer productivity; there are also suggestions that low motivation is an important factor in software development project failure. We investigate factors that motivate software engineering teams using survey data collected from software engineering practitioners based in Australia, Chile, USA and Vietnam. We also investigate the relationship between team motivation and project outcome, identifying whether the country in which software engineering practitioners are based affects this relationship. Analysis of 333 questionnaires indicates that failed projects are associated with low team motivation. We found a set of six common team motivational factors that appear to be culturally independent (project manager has good communication with project staff, project risks reassessed, controlled and managed during the project, customer has confidence in the project manager and the development team, the working environment is good, the team works well together, and the software engineer had a pleasant experience). We also found unique groupings of team motivational factors for each of the countries investigated. This indicates that there are cultural differences that project managers need to consider when working in a global environment.

1. Introduction

Motivation is the software engineering factor reported to have the single largest impact on practitioner[1] productivity (Boehm, 1981), and software quality management (McConnell, 1996). However, motivation continues to be 'undetermined' and problematic to manage (Procaccino et al., 2005) with no clear definition of what motivates software engineers. There is a clear need for a comprehensive model of motivation in software engineering (Beecham et al., 2007; França et al., 2012). Beecham et al. (2007) comment that "We also need a better way to measure motivation, as basing it on turnover only reflects whether an engineer is motivated to stay in an organization. It does not shed light on what motivates an individual to stay in the software engineering profession, to produce better quality software, increase productivity, and use and share skills".

The most critical component of any software development project is people. In DeMarco and Lister's survey (DeMarco and Lister, 1999) low motivation was found to be one of the most frequently cited causes of software development project failure. Motivation, however, often takes a back seat to other project factors that might be less important; perhaps this is because motivation is extremely difficult to quantify. As McConnell (1996) notes, "Every organisation knows that motivation is important, but only a few organizations do anything about it. Many common management practices are pennywise and pound-foolish, trading huge losses in motivation and morale for minor methodology improvements or dubious budget savings."

Our previous study found that culture, in terms of software engineers operating in different countries, is one factor which affects motivation. In particular, culture can affect the profile of characteristics associated with an individual software engineer (Beecham et al., 2008). Much of the previous research into software engineers'

* Corresponding author. Tel.: +56 75 2201710; fax: +56 75 2201713.
E-mail address: ncerpa@utalca.cl (N. Cerpa).

[1] The way a Software Engineer (as a practitioner) and Software Engineering (as a field) have been referenced has evolved significantly. IT, IS, SE, analysts, developers, programmers are examples of some of the terms used for the practitioner role/field. We use the term 'Software Engineer' (SE) or software engineering practitioner to refer to any of these roles and Software Engineering to refer to the field. However, when quoting or referring to a particular paper, we use the term used in the study.

0164-1212/$ – see front matter © 2014 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2014.01.008

motivation provides a western view of motivation. However, given the increasing importance of global software engineering (GSE), the increasing use of offshore developers, and the effects of cultural differences on GSE projects (Prikladnicki et al., 2003; Carmel, 1999), it is important to have a more complete understanding of motivation.

In this study we investigate team motivation, the impact of culture on team motivation, and the impact of team motivation on project success. We answer the research questions listed below through an analysis of responses to questionnaires from software engineers in Australia, Chile, Vietnam, and USA. The questionnaire addresses a number of issues including those related to project success (Verner et al., 2007; Verner and Evanco, 2005). The research questions we address are:

- RQ1 What is the team motivation for the projects; is team motivation the same for all countries?
- RQ2 What is the relationship between team motivation, and project outcome; is this the same for all countries?
- RQ3 What factors are related to team motivation? Are software engineers in Australia, Chile, Vietnam and the USA motivated in the same way?

Using data from Australia and the USA, we previously found that high staff turnover in the USA was significantly related to team motivation, and that the higher the turnover, the lower the team motivation (Hall et al., 2008). We now explore the relationship between team motivation and team motivational factors for software engineers from four different countries.

This paper is organized as follows: in the next section we provide some background to our study. In Section 3 we discuss our research methodology; then in Section 4 we present our results; in Section 5 we discuss the limitations to our research, and in Section 6 we present some conclusions and identify further research.

2. Background

In this section we present a brief review of the literature related to software project outcome, software engineering culture and motivation.

2.1. Project outcome

Software has been developed since the 1960s but still little is known about how to ensure that software projects are successful (Cerpa and Verner, 2009). Charette (2005) 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". Charette also provides a long list of high profile failed projects from around the world in his "Hall of Shame" and suggests that 5–15% of projects will be abandoned before or shortly after delivery as hopelessly inadequate. Widely publicized disasters include the temporary collapse of the Tokyo stock exchange (Langley, 2005), the power outage in north-east USA (Poulsen, 2004), and the Therac-25 radiation therapy machine (Collins et al., 1994).

Studies suggest failure rates for software development projects are up to 85% (Jørgensen and Moløkken-østvold, 2006). The Standish Group reports that 35% of software projects commenced in 2006 were successful (Rubenstein, 2007) compared with only 16.2% in their corresponding 1994 report (Standish Group International, 1994); however, the 6th CHAOS report still identifies 46% (52.7% in 1994) of software projects as challenged (having cost or time overruns or not fully meeting user's requirements) and 19% (31.1% in 1994) as outright failures (Rubenstein, 2007).

More recent research in this area is that of El Emam and Koru (2008), who suggest that, "although the overall project failure rate is high, word of a software crisis is exaggerated". Their project failure rate for delivered projects (from a survey conducted in 2007), was between 16% and 22%. However, recent reports such as those from the Audit Office of the UK Government (2011), Verner and Abdullah (2012), and newspaper reports (e.g., Vinegar, 2010) continue to describe serious IT project failures.

Software development projects are affected by a series of problems, such as low organizational maturity, lack of senior management involvement, poor project management, budget shortages, unrealistic project plans, poor requirements, changes in requirements and scope, adding staff late to meet schedules, lack of confidence in the team, unrealistic customer's expectations, poor quality software and under-motivated developers (Bennatan, 2000; Brooks, 1975; Cerpa et al., 2010; El Emam and Koru, 2008). Although there is some literature on the topic of project success, only a few studies have identified those characteristics that determine project outcome and define how to measure success (Thomas and Fernandez, 2008). Some researchers have developed models based on data from past projects to predict the outcome of new software development projects, and identify those aspects of the process that are more influential in determining project outcome (e.g., Cerpa et al., 2010; Reyes et al., 2011). Other studies suggest that project success and failure is a matter of perception and that it may vary from project to project (Pinto and Mantel, 1990; Shokri-Ghasabeh and Kavoousi-Chabok, 2009; Wateridge, 1995, 1998). A project may be viewed as a success or as a failure depending on the point of view of different groups of stakeholders (Bennatan, 2000). For example, in a previous study some practitioners defined a project as the most successful they ever worked on, while others stakeholders declared the same project to be a failure (Linberg, 1999). The project they refer to, was over budget by 419%, over schedule by 193% and it size was under-estimated by 130%. Based on these measures, one would say that this project was quite troubled, if not a failure. Practitioners mentioned schedule pressure, poor schedule estimates, poor understanding of resources required, poor understanding of the problem to be solved, as factors that lead to project failure (Linberg, 1999).

Another view of project success is one that considers both project management success and product success. Project management success covers meeting time, cost and quality objectives, while product success refers to the ability of the final product to meet the strategic organizational objectives of the owner of the project and satisfaction of users' and stakeholders' needs where they relate to the product (Baccarini, 1999). A study by (Shokri-Ghasabeh and Kavoousi-Chabok, 2009) reveals that 43% of practitioners surveyed believed that project success was indeed project management success; although 46% of the respondents indicated that these two kinds of success are totally different.

A project should not be considered successful only because it meets the desired time, cost and quality criteria (Shokri-Ghasabeh and Kavoousi-Chabok, 2009), and time, cost and quality are not the only project success criteria. Project managers should be educated to consider criteria other than these three (Collins and Baccarini, 2004). Success is perceived differently by different stakeholders, and researchers should not generalize the definition of project success (Shokri-Ghasabeh and Kavoousi-Chabok, 2009).

Another view of project success is the personal experience of developers. Developers consider things that may impact on them and their ability to perform well in their job; for example, a sense of achievement felt by a developer when doing a good job in a project (Procaccino and Verner, 2002). This view may also have to do with learning and acquiring new skills that may be used later on other projects (Glass, 1999). Developers may learn new skills from different projects, including those that have been cancelled. Having a sense of achievement, delivering quality, and providing a challenging and creative project environment for both managers