Available online at www.sciencedirect.com

SciVerse ScienceDirect

International Journal of Project Management 30 (2012) 407–417

International Journal of Project Management

www.elsevier.com/locate/ijproman

The impact of project capabilities on project performance: Case of open source software projects

Amir Hossein Ghapanchi [a,*], Aybuke Aurum [b]

[a] School of Information and Communication Technology, Griffith University, Queensland, Australia
[b] School of Information Systems, Technology and Management, The University of New South Wales, Australia

Received 31 May 2011; received in revised form 14 September 2011; accepted 20 October 2011

Abstract

Open Source Software (OSS) projects are contingent on volunteer developers and voluntary contributions from users, accordingly, traditional performance measures used for software projects, like being on time and budget, and satisfying specifications, may not be relevant for such projects. Although researchers have studied some predictors of OSS project performance, there is a lack of empirical work that studies the impact of project capabilities on the performance of OSS projects. Therefore, this paper studies project capabilities as potential predictors of OSS project performance. The data, gathered from 607 OSS projects over time (in two snap-shots), were consistent with Dynamic Capability Theory (DCT). In other words, the results showed that in order to achieve higher project performance, OSS projects need to have strong capabilities in terms of proactive and efficient defect-removal as well as proactive and efficient functionality-enhancement. The implications of these results for the OSS research community are discussed, followed by recommendations for OSS practitioners.

Keywords: Project performance; Open Source Software project; Project capability; Dynamic capability theory

1. Introduction

Open Source Software (OSS) is software which is released under a licence that allows the general public to use the software, as well as to examine its source code. The idea of 'opening' the code up to others enables the project team to make the software publicly available on the Internet so that users from all around the world can provide feedback on, and even contribute to, the project.

OSS development has come to be known as a reliable alternative to proprietary software that has the ability to produce cheaper, high quality software (Paulson et al., 2004). This has led numerous commercial and government organisations to adopt OSS products (Sen, 2007). A Forrester Consulting Study showed that more than 50% of North American and European companies use OSS products for their crucial applications (Gold, 2007). Furthermore, over 50% of American government organisations have adopted OSS (Gross, 2007).

The Mozilla web browser, the Apache web server, the Linux operating system, and MySQL database management system are instances of highly successful OSS projects. While many such OSS projects have achieved phenomenal success, almost 80% of OSS projects experience failure (Colazo and Fang, 2009), because they cannot maintain a high level of performance and activity, and because they become ineffective and inefficient (Fang and Neufeld, 2009). This problem has become more critical with the growing dependence of organisations on OSS products (Sen, 2007). This issue begs the question of what factors might contribute to OSS project performance.

Various factors have been reported in the literature as antecedents to OSS project performance, including quality and quantity of knowledge sharing (Long, 2006); product characteristics,

* Corresponding author.
E-mail address: amir.ghapanchi@gmail.com (A.H. Ghapanchi).

0263-7863/$ - see front matter © 2011 Elsevier Ltd. APM and IPMA. All rights reserved.
doi:10.1016/j.ijproman.2011.10.002

team structure and the number of developers (Liu, 2008); communication quality and team effort (Stewart and Gosain, 2006a); OSS project team participation (Capra et al., 2011) and developer experience and release speed (Hahn and Zhang, 2005). Prior studies on OSS projects have also implied the importance of the project capabilities in respect of defect-removal and functionality-enhancement in influencing OSS project performance (Crowston et al., 2003, 2006; Garousi, 2009a). However, to the best of our knowledge, no study has empirically investigated the impact that such capabilities might have on the performance of OSS projects. Thus, this study focuses on project capabilities as potential predictors of OSS project performance.

Our study contributes to the existing literature by: (1) revealing the significant impact that two of the most critical capabilities of OSS projects, namely defect-removal and functionality-enhancement, have on the performance of OSS projects, (2) extending our theoretical understanding of why defect-removal and functionality-enhancement might lead to a higher level of project performance through the theoretical lens of dynamic capabilities theory, (3) operationalising OSS projects' capabilities in respect of defect-removal and functionality-enhancement, and (4) studying a considerable number of OSS projects over time (in two snap-shots), instead of conducting a cross-sectional study, to enable dynamic investigation of the phenomenon under study.

The remainder of the paper is structured as follows. Section 2 provides some background information from the literature. The research model and theoretical justifications for the relationships proposed are presented in Section 3. In Section 4, the research methodology is described. In Section 5, the data analysis and the results of the study are presented. Section 6 derives and discusses the implications for both research and practice, followed by research limitations.

2. Literature review

OSS projects have idiosyncratic peculiarities which make it difficult to extrapolate results from traditional Closed-Source Software (CSS) development to OSS projects. For example, OSS projects are not typically driven by direct monetary profits (Lakhani and Wolf, 2003) and do not offer developers monetary incentives to participate in their development. Therefore the development of OSS applications is contingent on volunteer developers and voluntary contributions by users. In a typical OSS project, software source code is publicly available to everyone, in addition to other information such as communications history, defect-removal history and functionality-enhancement history. Given such points of difference, the traditional performance measures used for CSS projects, like being on time and budget, and satisfying specifications, may not be relevant for OSS projects (Stewart et al., 2006). Thus, the existing studies on OSS success mainly focus on alternative measures, such as the extent to which the OSS community has adopted an OSS project, to evaluate OSS project performance.

Project performance is one of the interesting research streams in OSS literature (Aksulu and Wade, 2010; Crowston et al., in press). Studies in this area mainly focus on different measures to evaluate project outcomes "during development". Various papers in this category have operationalised project performance in different ways. According to project management literature, project performance is composed of project effectiveness and project efficiency (Crawford and Bryce, 2003). Therefore, a comprehensive study of OSS project performance should incorporate both project efficiency and effectiveness. Efficiency simply refers to the extent to which output is created out of a particular amount of input ($\text{Efficiency} = \frac{\text{Output}}{\text{Input}}$). In other words, efficiency means doing things in the most economical way (Midha, 2007). Effectiveness, on the other hand, means the capability of producing an effect. The number of papers which take both OSS project effectiveness and efficiency into account is quite limited. Most studies either look at project effectiveness (Stewart and Gosain, 2006a,b; Subramanian and Soh, 2006) or project efficiency (Hahn and Zhang, 2005; Koch, 2004; Wray and Mathieu, 2008). These studies are discussed in more detail below.

One of the most cited papers on OSS project effectiveness is that of Stewart and Gosain (2006a). Their results showed (among other things) that communication quality and team effort (the number of work weeks) affect task completion. Stewart and Gosain (2006b) also looked at OSS project effectiveness as perceived by OSS project administrators. Their findings showed that the impact of developer team size on perceived effectiveness is more positive in the early project development stages than in later stages. Additionally, they found that the influence of task completion on perceived effectiveness was more positive in later project development stages than in early stages. OSS project efficiency has also been widely studied in the OSS literature. Wray and Mathieu (2008) ranked a set of 34 OSS projects based on their efficiencies. Koch (2009) found that adoption of Sourceforge.net tracker system and forum list as well as subversion and total tool adoption impact project efficiency. Moreover, Hahn and Zhang (2005) identified that developer experience, user list, news list, the tracker used and release speed significantly and positively affect efficiency of developer-targeted projects.

In the OSS literature, there are a number of papers that have studied OSS *project performance*. Hahn and Zhang (2005), for example, found that project management practices like Human Resource (HR) staffing, release management, communication and coordination, and compensation management all impact project performance. Long (2006) found that quality and quantity of knowledge sharing affects project performance. Liu (2008) demonstrated that product characteristics, team structure, the number of developers, developers' years of experience and targeting developers all influence project performance.

However, the literature lacks research that examines the impact of project capabilities on OSS project performance. Therefore, this paper seeks to fill this gap by examining the impact of project capabilities on the performance of OSS projects as defined by project effectiveness and efficiency.

3. Hypotheses and theoretical justifications

The Resource-Based View (RBV) has been widely employed in prior research (Jugdev et al., 2007) to underpin the potential