Global software testing under deadline pressure: Vendor-side experiences ☆

Hina Shah [a,*], Mary Jean Harrold [a], Saurabh Sinha [b]

[a] College of Computing, Georgia Institute of Technology, Atlanta, GA, USA
[b] IBM Research, India

ABSTRACT

Context: In the era of globally-distributed software engineering, the practice of global software testing (GST) has witnessed increasing adoption. Although there have been ethnographic studies of the development aspects of global software engineering, there have been fewer studies of GST, which, to succeed, can require dealing with unique challenges.
Objective: To address this limitation of existing studies, we conducted, and in this paper, report the findings of, a study of a vendor organization involved in one kind of GST practice: outsourced, offshored software testing.
Method: We conducted an ethnographically-informed study of three vendor-side testing teams over a period of 2 months. We used methods, such as interviews and participant observations, to collect the data and the thematic-analysis approach to analyze the data.
Findings: Our findings describe how the participant test engineers perceive software testing and deadline pressures, the challenges that they encounter, and the strategies that they use for coping with the challenges. The findings reveal several interesting insights. First, motivation and appreciation play an important role for our participants in ensuring that high-quality testing is performed. Second, intermediate onshore teams increase the degree of pressure experienced by the participant test engineers. Third, vendor team participants perceive productivity differently from their client teams, which results in unproductive-productivity experiences. Lastly, participants encounter quality-dilemma situations for various reasons.
Conclusion: The study findings suggest the need for (1) appreciating test engineers' efforts, (2) investigating the team structure's influence on pressure and the GST practice, (3) understanding culture's influence on other aspects of GST, and (4) identifying and addressing quality-dilemma situations.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Large software projects that operate with teams working at globally-distributed locations are developed using a *global software-engineering* process. Many industry reports, research papers, and other resources discuss the advantages and disadvantages of adopting global software-engineering practices, such as outsourcing and offshoring [2–6]. Despite the perceived disadvantages, global software engineering has grown into a large industry [7] and it is expected to continue to grow [8,9]. In 2006, ACM[1] conducted a study to provide an objective perspective on "the current and future trends in the globalization of the software industry" [7]. The study report lists the benefits, such as access to skilled resources, cheap labor, and economic benefits, that global software engineering provides. However, the report also acknowledges that there are challenges, such as cultural differences, information-security threats, and lack of appropriate educational policies, that need to be addressed to improve the practice.

To improve the global software-engineering practice, researchers have performed studies to understand the challenges that are introduced or amplified in this practice because (1) the practice has become important and is increasingly being adopted, and (2) less is known about real-world global software-engineering practice than non-global software-engineering practice [10]. In a systematic literature review conducted in 2010, Šmite and colleagues report that "global software engineering is still immature" [10] and one of the research questions in their study concentrated on understanding the state-of-the-art in empirical studies of global software engineering. Based on the study analysis, the researchers

---

reported the need to conduct more empirical studies to understand the global software-engineering practice so that appropriate solutions to address its challenges can be developed.

1.1. Existing research

Within global software engineering, global software testing (GST) (e.g., outsourced, offshored software testing), in particular, is growing rapidly and is expected to grow at an annual rate of over 20% [11]. Rooksby and colleagues argue that "if testing in industry is to be improved, then some appreciation of the circumstances in which testing is done can reveal challenges that new methods and technologies must overcome" [12]. However, little in known about how (collocated and global) software-testing activities are conducted in practice because software testing has received little attention [13]. Thus, there is a need to understand the software-testing practice, particularly the global software-testing practice, because it is being adopted rapidly.

Studies have been conducted to understand software-testing practice in general. Taipale and Smolander conducted a qualitative study to understand current software-testing practice, and suggest process-improvement propositions based on the study findings [14]. They found that the business orientation of an organization affects many aspects of software testing, such as testing schedules, test planning, and complexity of testing. Their process-improvement propositions include adjusting testing to business demands, improving development and testing team communications, and using risk-based testing. Martin and colleagues [15], through their ethnographic study, found that non-technical factors such as "organizational realities and constraints" play a significant role in the way software testing is practiced. They also highlight the gap between the software-testing research and practice, and emphasize the need to conduct organizational studies of real-world software-testing practice.

Similarly, Ahonen and colleagues discuss the impact of organizational models on software testing [16]. They compared three models and conclude that the resource-pool model, in which testing resources are placed in a pool and allocated based on the project needs and resource's expertise, is the most effective of those they studied. Moreover, they suggest the need to explore other hybrid organizational models that can better benefit the testing process. Rooksby and colleagues [12], conducted an ethnographic study in which they studied software-testing practice in agile, open source, and in-house settings. The results of their study demonstrate the cooperative and socially-organized nature of software-testing activities in real-time, real-world situations. Rooksby and colleagues report that studying only the technical aspects of testing is not sufficient because "the determinants of successful software testing have little to do with purely technical considerations."

These studies show that non-technical factors, such as socio-technical and organizational factors, have a significant influence on the way software testing is performed in an organization. Given the characteristics of GST—involving multiple organizations across geographic locations—the role of non-technical factors, such as socio-cultural factors, assumes an even greater significance. In the GST context, Hackett performed a study to understand the distributed and outsourced software-testing practice in an agile setting [17]. However, there is only one systematic study that concentrates on understanding how software testing can be conducted effectively in a globally-distributed software-testing setting [13]. Casey used an inductive case-study approach to identify key factors that hinder and facilitate effective execution of globally distributed software-testing activities. Some of his key findings include the influence of fear on offshore work practice, the impact of fear on communication across teams, and the difficulties involved in offshoring software-testing activities, which does not necessarily make it an easy-to-offshore activity. The results of these studies show that there is clearly a need to conduct further studies in the GST to better understand this practice.

Rooksby and colleagues show that time is of great importance in software testing [12]. Although not specific to software-testing activities, research has attempted to understand the role of time/pressure in software engineering. For example, Austin studied the effect of time pressure on quality in software-development related activities [18]. He discusses the "short-cuts" approach adopted by software developers to meet deadlines, which sometimes results in compromises in quality. Based on the analysis, he derives recommendations for estimation and deadline setting. For another example, Costello discusses how deadline pressures work against practicing effective software-engineering activities and suggests a resource-allocation approach to address the problem [19]. However, time pressure in the GST setting has yet to be studied in depth.

1.2. Our research

Thus, to enhance the GST practice, it is important to understand the practice by studying it in a real-world setting for several reasons. First, software testing in general, and global software-testing practice in particular, are relatively less studied areas of research [12,13]. Thus, there is a need to conduct empirical studies to better understand this practice and its inherent challenges. Second, previous research shows that non-technical factors, such as socio-cultural and organizational factors, play a significant role in software-testing activities. In global settings, such as the outsourced, offshored setting, multiple organizations and people from different cultures are involved. Thus, it is worth investigating how these factors affect the global software-testing activities. Third, it is assumed that global software practices, such as outsourcing and offshoring software-testing related activities, are "smooth and easy." However, recent studies have reported the difficulties and challenges involved in offshoring these activities [13]. Hence, our research goal is to investigate this GST practice in detail. Our work complements Casey's study that also adopted an inductive approach to studying GST teams [13]; our study can be viewed as a similar study performed in a different context and setting. However, in contrast to Casey's study, our study emphasized the testing activities and experiences as perceived by the test engineers instead of testing-related project management.

As the first step toward gaining an understanding of the GST practice in the real-world setting, we conducted an ethnographically-informed study at a vendor organization that provides outsourced, offshored software-testing services. Because the duration of our visit to the organization was limited to 2 months, a longitudinal ethnographic study was infeasible, so we narrowed the scope of our study. We decided to concentrate on understanding how test engineers perform their tasks under pressure situations because time plays a crucial role in software testing and because this was of interest to the vendor organization in which we conducted the study. Thus, we investigated the following research question

> How do vendor-side test engineers in the GST setting accomplish their tasks under deadline pressure situations?

We studied three teams who were involved in providing both manual and automated testing services to clients. We collected the data through face-to-face interviews, observations, and informal chats, and performed the data analysis using the thematic-analysis framework [20].

Section 2 presents the details of our study design. Section 3 reports the findings of the study: how software test engineers per-