The Impact of Defect Resolution on Project Activity in Open Source Projects: Moderating Role of Project Category

Abstract:

The importance of defect resolution on project success in open source projects has been highlighted by prior research. However, there is a lack of empirical work examining the impact of defect fixing on project activity. This paper aims to better understand the relationship between defect fixing and project activity in an open source environment. Data collected from 240 open source software (OSS) projects confirmed that defect fixing positively and strongly impacts on project activity. In addition, the magnitude of this impact was found to be higher for projects with a broader range of use (e.g. ‘Communication’ programs) than those with a lower range of use (e.g. ‘Engineering’ programs).

Key words: Project activity, Open source software, OSS success, Defect fixing process.

1. Introduction

Many OSS projects fail in the early stages of their life (Chengalur-Smith and Sidorova, 2003). According to Krishnamurthy (2002), 63% of OSS projects on Sourceforge.net, the largest OSS host in the world, fail due to their inability to attract the attention of developers and users in the community.

Various factors have been reported in prior studies as antecedents to OSS project success. Certain decisions that project managers have to make before launching a project can influence its success, for

1 Corresponding author
example project license (Subramaniam et al., 2009), Sponsorship (Stewart et al., 2006), and project audience and project topic (Crowston and Scozzi, 2002).

In order to attract higher user interest, an OSS project team has to care about users’ needs. One of the most critical areas in OSS projects that generate high user demand is defect removal. Defect-fixing is one of the most important processes in community-based OSS development. Thus defect fixing might have an influence on project success. Although prior research on OSS projects has implied that defect-fixing has an important impact on OSS project success (Crowston et al., 2003; Garousi, 2009; Crowston et al., 2006), there is a lack of empirical work examining this phenomenon. Hence the focus of our study is to examine the interacting relationships among defect fixing, project activity, and software project category.

The remainder of this paper is structured as follows. The next section reviews the related literature. The research methodology employed in this research is presented in Section 3. Section 4 represents data analysis and the findings, followed by discussion and concluding remarks in Section 5.

2. Research Background

2.1 OSS Success

Many OSS projects cannot grow, remaining in their immature stages or even ‘dying’ for different reasons. On the other hand, there are also many examples of successful open source projects that have achieved huge success in the market. Mozilla Firefox, Apache, Open office, and the Linux operating system are examples of such projects.

In simple terms, success means achieving something desired (Midha, 2007). Measuring success is difficult because it is subjective. Crowston et al. (2006) says “… these measures [OSS success measures] are hard to define for regular I/S [information system] projects and doubly hard for FLOSS [Free/Libre OSS] projects, because of the problems defining the intended user base and expected outcomes”. This has led to the multitude of perspectives in the literature on OSS success.

Current literature has studied various aspects of OSS success such as project activity (e.g. Liu, 2008; Stewart et al., 2006; Colazo, 2007; Long, 2006a; Grewal et al., 2006), user interest (e.g. Subramaniam et al., 2009; Stewart et al., 2006; Crowston and Scozzi, 2002; Midha, 2007), and project performance (e.g. Liu, 2008; Long, 2006a; Guiri et al., 2004; Hahn and Zhang, 2005). This research focuses on ‘project activity’ because high project activity can impact community interest in an OSS project (Subramaniam et al., 2009).
Although several studies have reported antecedents of OSS success, there is a lack of research that has taken defect fixing into account.

### 2.2 Defect Fixing

Defect fixing has been studied by researchers for almost three decades. However, there is little insight into this phenomenon in an open source environment. The primary objective of the defect-fixing process is to detect defects and then fix them.

Several researchers have examined the defect fixing process in OSS projects. For instance, Stewart and Gosain (2006a) used the percentage of bug reports completed as an indicator of OSS project effectiveness. Herbsleb and Mockus (2003) state that the progress in fixing bugs reported and responding to user interests reflect the outcomes of an OSS project. Stewart and Gosain (2006b) also proved that task completion (including the percentage of bug reports completed) impacts perceived effectiveness of OSS projects. Furthermore, through three case studies, Garousi (2009) found that various OSS projects exhibit different levels of responsiveness to the defect-fixing process due to factors like activity level.

One of the most critical areas in OSS projects that users demand very much is defect removal. Thus, defect fixing might have an influence on project success. Literature on OSS projects has also implied the importance of defect-fixing in impacting OSS project success (Crowston et al., 2003; Garousi, 2009; Crowston et al., 2006).

### 2.3 Research Model

The primary objective of this research is to explore the impact of defect removal on project activity. As an OSS project grows and more users become aware of its existence, it receives increasing levels of feedback from the community, including defect reports (Wynn, 2004). The more defects a project resolves and the more effective the defect-fixing process operates, the earlier the next version can be released (Hahn and Zhang, 2005). Therefore, we argue that defect-fixing might impact project activity. Hence, we pose hypothesis 1 below:

**H1.** Defect-fixing has a positive impact on project activity in an OSS environment.

Stewart et al. (2006) considered ‘project category’ as a moderator in the relationship between OSS success and its determinants. The importance of defect-fixing might be greater for open source software with a broader range of uses (such as communication) than for projects that have a more limited scope of use (like engineering software, which are normally used for limited purposes). This is because software with a higher range of uses potentially have a higher number of users. Hence
fixing more defects should result in more users becoming involved in development activity that can lead to higher project activity. For this reason, and in line with Stewart et al.’s (2006) study, we argue that the impact of defect fixing on project activity over time will be stronger for OSS projects that have a broader range of potential uses than for those with a narrower range of potential uses. Hence, we pose hypothesis 2 as below:

**H2.** The magnitude of the impact of defect-fixing on project activity is associated with the project’s category.

Based on the hypotheses posed, we developed a model of relationships (See Figure 1) to be tested. The only dependent variable of the model is project activity. There is one independent variable: defect-fixing. There is also one moderator in the model: software category. As Figure 1 shows, there are 2 hypotheses underlying our research model.

![Research Model](Fig 1. Research Model)

### 3. Research Method

The setting for this research is the largest OSS repository, Sourceforge.net. As of Feb 2009, Sourceforge has 230,000 registered open source software projects, and it had more than 2 million registered members (Source: www.sourceforge.net).

#### 3.1 Data Collection

Sourceforge divides OSS projects into various categories including: communication, software development, scientific/engineering, database, desktop, education, formats and protocols, games and entertainment etc. In order to limit our sample, in line with prior research (Stewart et al., 2006) we will focus on three categories: *Communication*, *Software Development*, and *Scientific/Engineering*. 
The Scientific/Engineering and Communication categories were chosen because they show different extremes in terms of the way that people generally use software: Scientific/Engineering software, for example, is generally used for a relatively narrow set of engineering and scientific related purposes, whereas communication software programs are more commonly used for a wide range of purposes. The Software Development category, on the other hand, sits somewhere between these extremes. Projects in this category are typically programming tools for developers.

<table>
<thead>
<tr>
<th>Project category</th>
<th>Frequency</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>80</td>
<td>33.3%</td>
</tr>
<tr>
<td>Software development</td>
<td>80</td>
<td>33.3%</td>
</tr>
<tr>
<td>Scientific/Engineering</td>
<td>80</td>
<td>33.3%</td>
</tr>
<tr>
<td>Total</td>
<td>240</td>
<td>100%</td>
</tr>
</tbody>
</table>

In order to narrow our sample space, we decided to exclude projects that had not had any releases within the last 2 years. We also excluded projects whose development status was ‘planning’, ‘pre-alpha’, or ‘alpha’ (because these projects are immature and normally don’t have any software releases) as well as those with a ‘mature’ development status (because these projects do not normally have much activity and are already mature in terms of the resolution of defects, resulting in less project activity). Finally we focus only on those projects that have had at least 5 records in their defect-tracking system.

This research applies a random sampling method which is said to be appropriate for studying OSS projects. Random sampling was used to select OSS projects from Sourceforge. 240 projects were sampled from the projects that met the criteria mentioned above (80 projects from the Communication category, 80 projects from the Software Development category, and 80 projects from the Scientific/Engineering category). We collected data on all of the 240 projects. Tables 1 and 2 show some demographic information on the projects in our sample space.

<table>
<thead>
<tr>
<th>Number of developers</th>
<th>Frequency</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>104</td>
<td>43%</td>
</tr>
<tr>
<td>4-10</td>
<td>78</td>
<td>33%</td>
</tr>
<tr>
<td>11-20</td>
<td>37</td>
<td>15%</td>
</tr>
<tr>
<td>&gt;20</td>
<td>21</td>
<td>9%</td>
</tr>
<tr>
<td>Total</td>
<td>240</td>
<td>100%</td>
</tr>
</tbody>
</table>
3.2 Measuring the Variables

In order to measure the variables in the research model, we selected measure(s) for each variable. We used the ‘number of defects fixed’ to measure ‘defect fixing’ (Stewart and Gosain, 2006a; Herbsleb and Mockus, 2003; Stewart and Gosain, 2006b). The ‘number of files released’ was used to measure ‘project activity’ (Subramaniam et al., 2009). Finally, ‘Project category’ was measured using a categorical variable i.e. 1= the project was in the Communication category, 2= the project was listed in the Software Development category, and 3= the project was listed in Scientific/Engineering category (Stewart et al., 2006).

4. Data Analysis and Results

4.1 Direct Effect

Data analysis was undertaken through partial least squares (PLS) analyses using PLS-Graph 3.0. PLS doesn’t need normal distribution for the variables and is also flexible in terms of sample size. Before testing the moderating effect of software category, the direct effect of user defect fixing on project activity was first tested (Hypothesis 1). A significant and positive direct effect (coefficient = 0.387) was found between defect fixing and project activity. This relationship was also significant (t-value = 2.8) at the 0.05 level. Therefore Hypothesis 1 is accepted.

4.2 Moderation Effect

Moderated multiple regression analysis (MMR) was then applied to test the moderating effect of project category. A three-step approach was adopted to test the moderating effects using MMR with Smart-PLS. First, defect fixing was regressed with project activity and project category (See model 1 in Table 3). Second, the interaction term was added between defect fixing and project activity (See model 2 in Table 3). Third, the difference in $R^2$ between model 1 and model 2 was calculated. Then Cohen’s F-square was calculated. The result of the test is shown in Table 3.

The path coefficient for the moderating effect was -0.169. However Cohen’s F-square of 0.031 showed a small to medium effect. Moderation effects with Cohen’s f-square of 0.02, 0.15, and 0.35 are considered small, moderate, and large effects, respectively (Cohen 1988). Hence the magnitude of the impact of defect fixing on project activity was associated with project category, supporting Hypothesis 2.
### Table 3. Interaction effect

<table>
<thead>
<tr>
<th>Independent variable</th>
<th>Direct effect</th>
<th>Moderating effect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Model 0</td>
<td>Model 1</td>
</tr>
<tr>
<td>Defect-fixing (DF)</td>
<td>0.387</td>
<td>0.39</td>
</tr>
<tr>
<td>Project category (PC)</td>
<td></td>
<td>0.049</td>
</tr>
<tr>
<td>DF*PC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R²</td>
<td>R₀²=0.15</td>
<td>R₁²=0.152</td>
</tr>
</tbody>
</table>

### 5. Conclusion and Discussion

The focus of our study was to examine the relationships among defect fixing, project activity, and software project category. Specifically, we argued that defect fixing had a positive impact on project activity; however the magnitude of this impact was influenced by project category. The data collected on 240 OSS projects confirmed the hypothesis that the magnitude of the impact of defect fixing on project activity was higher for projects with a higher range of use (such as communication software), was lower for projects with a lower range of use (like science and engineering software) and was medium for projects with an in between range of use (e.g. Projects in the Software Development category). This is because projects with a higher range of use potentially have a higher number of users. Hence fixing more defects is likely to result in more users being involved in development activity, which can lead to higher project activity.

This study has valuable implications for practice. Firstly, according to the results from statistical analysis, OSS project teams that pay more attention to defect removal and fix a higher number of defects are more likely to undergo more development activity and frequent software releases. As a result such projects are more likely to attract higher interest from the community. Secondly, projects listed in categories that normally have a higher range of uses (like communication and security) need to take defect fixing more seriously since the impact of defect fixing on project activity in these projects is higher than its impact on projects in categories with lower range of uses (e.g. science and engineering).

### References:


Hahn, J., and Zhang, C. (2005). An exploratory study of open source projects from a project management perspective. paper presented at MIS Research Workshop, Purdue University, West Lafayette, IN.


