

# **Resources Contributing to Gaining Competitive Advantage for Open Source Software Projects: An Application of Resource-Based Theory**

## **Abstract**

Open Source Software (OSS) is an important asset in today's software-intensive society. The success of OSS projects is highly dependent on a number of factors. These factors must be understood and managed as an OSS project progresses. Thus, project management of an OSS project has a decisive role in ensuring the success of its software. The objective of the research is to increase the understanding of the resources affecting the competitiveness of OSS projects. Herewith, the responsiveness of OSS projects to users' needs is assessed via an investigation of the defect-fixing process. A Resource-Based View of the firm (RBV) is used to build theoretical justifications for a set of hypotheses proposed in this study. Data gathered from 427 OSS projects confirmed that developers' interest in and users' contribution to the project as well as frequently updating and releasing the software affect the project's ability to gain competitive advantage through effective defect-fixing. It is also shown that OSS projects that are more popular and have a higher level of organizational communication than others are more likely to gain competitive advantage through effective defect-fixing. Finally, implications of the results for practitioners and the research community are presented.

## **1. Introduction**

Open source software (OSS) has changed the way that software is developed, deployed, and perceived. The spread of OSS was facilitated by commercial and government organizations, which adopted OSS widely (Carillo & Okoli 2008). Nowadays, influential organizations of all sizes have

adopted OSS products (Sen 2007a) because OSS has been recognized as a cheap and reliable alternative to proprietary software (Paulson et al. 2004).

However, despite the increasing adoption of open source software, many OSS projects fail in the early stages of development (Chengalur-Smith & Sidorova 2003; Aksulu & Wade 2010) for various reasons such as inability to attract volunteer developers to join their development team or inability to attract voluntary contribution from user community (Subramaniam et al. 2009). Whereas Linux and Apache have been highly successful instances of OSS (Nelson et al. 2006; Sen 2007b), the majority of OSS projects fail (Colazo & Fang 2009) because of low responsiveness to user needs (Golden 2004). According to Krishnamurthy (2002), 63% of OSS projects on Sourceforge.net, the world's largest OSS host, fail. This might be because a large majority of OSS projects cannot attract the interest of the user community (Stewart & Gosain 2006a) for a number of reasons, such as low responsiveness to user needs (Golden 2004) (e.g., in terms of fixing software defects).

Herein, the responsiveness of OSS projects to user needs is studied via an examination of the defect-fixing process. Previous research on OSS projects has suggested that effective defect-fixing is important for the success of OSS projects (Crowston et al. 2003; Crowston et al. 2006; Garousi 2009). Effective defect-fixing is tied to users' perceptions of the quality, value, and project development activity (e.g., number of line of codes) (Mockus & Weiss 2008; Midha et al. 2010). In light of this, the objective of the study reported herein was to investigate the factors that drive the effectiveness of the defect-fixing process, because defect-fixing may well be a strong source of competitive advantage for OSS projects.

To achieve this objective, we developed and assessed a set of hypotheses regarding the influence of project resources on the effectiveness of the defect-fixing process. Previous studies have proposed a number of factors that might be useful when predicting OSS projects' positive outcomes. For example, certain decisions that project managers have to make before launching the project can influence success. Examples of this include decisions on programming language (Chandrasekar Subramaniam et al. 2009), sponsorship (Stewart et al. 2006), project audience, and project topic

(Crowston & Scozzi 2002). Our study complements this earlier work by focusing on a further factor, *project resources*, because identifying the project resources that have the potential to affect defect-fixing effectiveness might provide organizations that are interested in adopting OSS products with certain criteria for selecting OSS projects. Herein, defect-fixing effectiveness is defined as the extent to which the OSS project accomplishes to remove the defects existing in the software. On the assumption that a project's resources act as a source of added value for users by influencing the effectiveness of the defect-fixing process, we decided to address the following research question:

**RQ:** What are OSS project resources that have the potential to affect defect-fixing effectiveness and, therefore, might provide organizations that are interested in adopting OSS products with certain criteria for selecting OSS projects?

The answer to this question has both theoretical and practical implications. In order to answer the research question, we used Resource-Based View (RBV) of the firm, according to which, in order to attain competitive advantage, projects need to have strategic resources that are valuable, rare, non-imitable, and non-substitutable (Jugdev et al. 2007).

Our study contributes to the existing literature by: (1) extracting the critical resources of OSS projects which are important for their defect-fixing process; (2) revealing the significant impact that the identified OSS projects' resources have on the effectiveness of their defect-fixing process; (3) applying the resource-based view of the firm to the context of OSS projects; and (4) studying a considerable number of OSS projects.

The remainder of the paper is structured as follows. In Section 2, the research background is discussed. The research model and theoretical justifications for the relationships proposed are presented in Section 3. In Section 4, the design of the study is described. In Section 5, the data analysis and the results of the study are presented. The implications for both research and practice are derived and discussed in Section 6. In Section 7, the limitations of the study are noted followed by concluding remarks in Section 8.

## 2. Research Background

OSS projects seem to involve a never-ending process of defect fixing. During the defect-fixing process, defects that are observed in the software are handled and resolved to improve the quality of the software. However, this process should not be characterized as simple maintenance, because the software always evolves to fulfil new user requirements (Ghapanchi & Aurum, 2012). That is why previous research considers continuous defect fixing to be one of the processes that characterize OSS projects (Crowston et al. 2006).

The defect-fixing process has been addressed in only a few studies on OSS. Herbsleb & Mockus (2003) found that the progress in fixing defects influences the positive outcomes in OSS projects. Stewart & Gosain (2006a) used the percentage of defect reports completed as an indicator of the effectiveness of OSS projects and found that the quality of communication among team members and team effort affect the quantity of defect reports completed. They suggested that the success of an OSS project is a function of the extent to which a project receives input from the community, and the extent to which it creates an observable output, such as a defect fixed. Stewart & Gosain (2006b) also provided strong evidence that the percentage of defect reports completed affects the perceived effectiveness of OSS projects, where 'perceived effectiveness' is defined by how well an OSS project succeeds in accomplishing its goals.

An OSS project can be launched through the project website or using an existing portal that offers open source hosting services. Examples of the former are Open Office and Firefox. Examples of portals are Sourceforge.net, Freshmeat.net, OSDir.com, and BerliOS.com. Since 1996, a number of web portals have been launched that are dedicated to serving open source projects, providing a wide range of tools for such matters as development, defect fixing, enhancement, software release, communication, and coordination (Capiluppi et al. 2003). These portals typically provide a defect-tracking system for the purposes of providing quality assurance.

The study reported herein examined Sourceforge.net. Sourceforge does not clearly specify the process by which defects are fixed through its defect-tracking system, but defines the different statuses of a

defect (See Figure 1). When a defect is first reported, it is assigned the status Open. Subsequently, someone (e.g., a project administrator) either assigns the defect to a developer to be fixed, rejected if it is duplicate, out of date or not legitimate; the "pending" status is also used when the defect is legitimate but it is deemed better to be fixed. When a defect is resolved, the defect is assigned the status Fixed. Finally, when the defect report is completed, the status is changed to Closed.

**Figure 1. Defect statuses on Sourceforge.net**

### **3. Research Model**

#### **3.1 Resource-based View of the Firm**

According to the resource-based view, competitive advantage and firm performance are influenced strongly by the firm's resources (Jugdev et al. 2007; Barney and Clark 2007). Resources that are valuable and scarce, and are able to benefit the firm, can bring a temporary competitive advantage to the firm (Hulland et al. 2007). By protecting its resources against imitation and substitution, the firm can sustain the temporary competitive advantage (Wade & Hulland 2004; Lavie 2006; Ray et al. 2003; Barney and Clark 2007). The RBV distinguishes between different types of competitive advantage (temporary competitive advantage V.s. sustained competitive advantage) (Barney 2002). It should be noted that this paper looks into a temporary competitive advantage for OSS projects because of the two following reasons: (1) due to the openness of the OSS projects, their resources are normally imitable; (2) most OSS projects are abandoned after a while (Colazo & Fang 2009).

Barney (1995) introduced the VRIO framework that considers strategic resources as those which are valuable, rare, non-imitable and involve organizational support. If a resource is valuable, rare, hard to imitate and involves it being exploited by the company then it is able to provide the company with a sustained competitive advantage (Barney 1995).

We herein use RBV as a theoretical perspective from which to examine defect fixing for several reasons. Firstly, RBV provides an appropriate theoretical framework to explain the effect of OSS

project resources on defect-fixing effectiveness and underpins the set of hypotheses which will be proposed in this research. Secondly, RBV has been applied successfully in the context of Information Systems (IS) in general (Tarafdar & Gordon 2007) and software projects in particular (Scacchi 2007). Thirdly, RBV can be applied at the project level (Gefen & Straub 2005; Song & Montoya-Weiss 2001). Fourthly, the theoretical claims made by the RBV have received empirical support from the results of a number of studies (e.g., Hulland et al. 2007; Jugdev et al. 2007; Srivastava et al. 2001).

Drawing on RBV, given that OSS projects may provide products and services, and entry into new markets, they will need resources if they are to attain competitive advantage, over both its proprietary competitors and its OSS competitors. Now, responsiveness to customer needs (e.g., in terms of fixing software defects) is one of the most frequently cited concerns of IT practitioners who adopt OSS (Golden 2004). Moreover, an effective defect-fixing process increases users' sense of the software's quality and value (Mockus & Weiss 2008). Therefore, an effective defect-fixing may be considered as a source of competitive advantage for OSS projects.

Firm resources are those that a firm can use in order to fulfil its strategies. According to Daft (1983), firm resources include all assets, firm attributes, organizational processes, capabilities, information, and knowledge. Barney (1991) suggests three types of resource when introducing RBV: human capital, organizational capital, and physical capital. Human capital resources comprise the expertise, knowledge, intelligence, and experience of the firm's employees. Organizational capital resources are an organization's planning, reporting, coordinating, and controlling mechanisms, and relations between people within the firm and between the firm and its environment (Barney 1991); they might also include company reputation, brand image, and product quality (Bharadwaj 2000). Physical capital resources comprise plants, equipment, and technology, and raw material as input to organizations' production processes (Hofer & Schendel 1978). In the context of IS research, the term 'technical resources' has been frequently used as physical resources (Tarafdar & Gordon 2007). Technical resources include hardware, software, networks, applications and databases (Weill & Broadbent 1998; Bharadwaj 2000; Lopes & Galletta 1997). We use Barney's (1991) classification

(but use the term ‘technical resources’ because our physical resources are all technical) and apply it to the defect-fixing process in OSS projects.

### **3.2 Extracting Resources - Interview**

In order to extract OSS project resources required for an effective defect-fixing process, we used the interviews taken place by Sourceforge.net with 22 project administrators of projects of the month (10 projects in 2008, and 12 projects in 2009). Interview transcripts were extracted from <http://sourceforge.net/blog/potm>. Ethics approval was not required for this study as all the interview transcripts were publicly available on sourceforge.net.

We conducted content analysis on 112 pages of interview scripts. Using the software NVivo, we used established principles of coding and analyzing the data, namely thematic analysis (Ezzy 2002), to identify project resources. The code names themselves were based on terms discovered during the literature review of OSS. Codes were created organically when reading the interview transcripts and were assigned until no further codes were discovered and thus saturation was achieved. The collection and analysis of data has been an iterative and interpretative process. Based on going analysis of the data we identified a number of key themes (i.e. nodes) as we read through the transcripts. The themes were actually potential project resources. The themes which achieved a frequency higher than 10 were then extracted and the resources were identified. As a result, we came up with 5 project resources namely: developer interest, user contribution, release frequency, project popularity, and organisational communication. The following bulleted lists show examples of the quotes from the interviews that led us to extract each resource category. The different resource categories are further discussed in Sections 3.3-3.8.

Several interviewees highlighted the need to attract developer interest:

- “Our real challenge is to scale up the team and recruit new people who share the same passion” [Interviewee #11].

- “We have a small team of great hardworking people, but we would love to have a great medium sized team to help [our project] grow and expand further” [Interviewee #13].
- “And of course, we warmly welcome any interested developers on board” [Interviewee #4].

Another important project resource is the user contribution, which was highlighted by several interviewees:

- “External contributors submit patches that a team member will commit after review” [Interviewee #11].
- “We have described each of these in the “How to Contribute” section, everyone can make a difference” [Interviewee #8].
- “Given the wide latitude of possibilities that [our project] offers, we’d like to see more community contributions ...” [Interviewee #17].

Technical resources are also important and release frequency was mentioned by several of the interviewees:

- “[if we could change something about the project, we would] release new versions as soon as possible” [Interviewee #2].
- “We have always had a high quality of code, few bugs, and a fairly decent release rate” [Interviewee #5].
- “Once the releases started to flow, the user base took off, and the torrent just keeps growing” [Interviewee #8].

Understanding project popularity is crucial for open source software projects and it is closely related project resources. Interviewees commented that:

- “We've see[n] a huge amount of adoption beginning last July with the release of our platform and it's just continued to grow as the word has spread” [Interviewee #17].
- “It took quite some time before the word got out about our product” [Interviewee #6].



- The community is our strength and we want [our project] to be the web analytics framework of choice” [Interviewee #13].

Communication was also seen an important aspect in OSS project as the developers and user community do not operate in close proximity. Examples of comments from the interviewees included:

- “There is generally a lot of interest in [our project], via the forums and by email [mailing lists]” [Interviewee #9].
- “... there is a forum where everyone can post questions and offer help to others” [Interviewee #9].
- “We're constantly being amazed by the stories users share with us in our forums” [Interviewee #21].

### **3.3 Developer Interest – Human Capital Resource**

After extracting the key resources required for effective defect-fixing, we tried to fit them within the resource framework proposed by Barney (1991). The following sentences describe this correspondence. As a result of this correspondence, we relate human resources to developer interest and user contribution to the project. The frequency with which software releases are made will be viewed as a technical resource. The term ‘organizational resource’ is operationalized by project popularity and organizational communication. In what follows, we describe each resource and formulate an associated hypothesis.

The first resource that is of critical importance for an OSS project in order to run an effective defect-fixing process is the developers. The skills and expertise of individual developers involved in an OSS project can be viewed as a valuable resource for the project which is also non-substitutable because developers are the ones who generate the software code. Previous research has found that having highly skilled developers is a factor for the success of OSS projects (Colazo 2007; Giuri et al. 2004; Stewart & Gosain 2006b). This prompted us to consider whether the interest that developers have in an OSS project might constitute a valuable resource for the project to gain competitive advantage. From this perspective, the greater the interest that developers have in the project, the greater will be

their contribution to a more effective defect-fixing process (See Figure 2). Developers typically join an OSS project whose product fulfils their personal need or holds some interest for them (Raymond 1999), thus they do not normally join two projects that are working on the same kind of software application. That leads us to posit that developers' interest might be also a scarce and non-imitable resource for the defect-fixing process. On the basis of the foregoing, we formulate the following hypothesis:

**H1.** Developer interest has a positive effect on the effectiveness of the defect-fixing process.

### **Figure 2. Research Model**

#### **3.4 User Contribution – Human Capital Resource**

'User contribution' is the contribution that the community makes to aspects of an OSS project, such as fixing a defect (Kavanagh 2004). As mentioned above, OSS projects are commonly run on a voluntary basis; hence, attracting contributions from the user community is a valuable and non-substitutable resource for their success. Such user behaviours might also have stronger connections to performance outcome (Sundaram et al. 2007). Drawing on RBV, Bogers et al. (2010) showed that user innovations and contributions are a key capability for the organization. Hence, it is likely that OSS projects that receive a greater contribution than other projects from their users in terms of resolving defects will have a more effective defect-fixing process. Further, users typically contribute to an OSS project whose software addresses their needs (Raymond 1999), hence they do not typically use and contribute to two projects that are developing the same kind of software. The fact that this is so indicates that in addition to be valuable and non-substitutable, user contribution might be a rare and non-imitable source for the defect-fixing process. On the basis of the foregoing, we formulate the following hypothesis:

**H2.** The contribution that users make to fixing defects has a positive effect on the effectiveness of the defect-fixing process.

### 3.5 Release Frequency – Technical Resource

As mentioned above, technical resources can include hardware, software, networks, applications, and databases (Weill & Broadbent 1998; Bharadwaj 2000; Lopes & Galletta 1997).. Such resources shape the platform on which the software project can operate. OSS projects are typically run virtually and by volunteer developers from all around the world. The developers of these projects typically use their own personal computer to take part in the project as well as their own access to the Internet (Scacchi 2007). In addition, these developers bring their own choice of tools (e.g., compilers of source code), programming methods, and development techniques to the project (Scacchi 2007). As a result, “There are few shared computing resources beyond the project’s Web site ... (Scacchi 2007), p. 11).” Given that a typical OSS project does not provide the developers with networks and hardware, project software release would seem to constitute a shared computing resource for OSS projects.

Release frequency means how often the core developers update an OSS project to release a newer version of the software (J. Long 2004). Release management is one of the most important project management issues in OSS development (Hahn & Zhang 2005). Crowston et al. (2006) state that release management is one of the most critical continuing processes that characterizes OSS projects. Each version of the software incorporates a number of defects, so resolving them will lead to better-quality software. Every time an OSS project releases a new version of the software, the users try to download, use, and see the new updates in the latest version of the software. This promotes the use of the software in the community and as a result increases the chance that a particular defect can be identified. The more often a project team releases a new version of the software, the greater the chance that a user/developer in the community will find a defect in the software. This, in turn, provides more input for the defect-fixing process. On this basis, we formulate the following hypothesis:

**H3.** The frequency with which new versions of the software are released has a positive effect on the effectiveness of the defect-fixing process.

### **3.6 Project Popularity – Organizational Capital Resource**

Company popularity and reputation have been regarded as organizational capital resources (Bharadwaj 2000). Firm popularity has been often mentioned in the literature as a valuable, rare, non-substitutable, and non-imitable resource that is critical for achieving competitive advantage (Bharadwaj 2000; Vergin & Qoronfleh 1998). In the OSS context, project popularity is an organizational attribute that can affect the number of users who adopt an OSS product and as a result influence the number of potential defect detectors for the project (Chandrasekar Subramaniam et al. 2009). Researchers have often used the number of times an OSS product has been downloaded as a proxy for the project's popularity (Midha 2007). The number of downloads shows the size of the user community of a given OSS product. A larger user community leads to more defects being identified because, as Raymond (Raymond 1999) says, "given enough eyeballs, all bugs are shallow". Further, the identification of defects provides input for future uses of the defect-fixing process, and the more defects are identified, the greater the input. This leads us to postulate that the effectiveness of defect-fixing process might be greater for OSS projects that are more popular. Accordingly, we formulate the following hypothesis:

**H4.** A project's being popular has a positive effect on the effectiveness of defect-fixing process.

### **3.7 Organizational Communication – Organizational Capital Resource**

Barney (1991) defines organizational capital resources as an organization's planning, reporting, coordinating, and controlling mechanisms. Organizational communication is a source of planning, coordinating, and knowledge-sharing in the project. It has been suggested in the literature that organizational communication affects the project's competitive advantage (Daft 1983). A project's mailing lists and forums are media that users and developers of an OSS project can use to communicate each other for planning, coordinating and knowledge-sharing. Users and developers in an OSS project normally post messages on the project mailing lists or project forums to share knowledge, coordinate or plan activities, ask for help, report a suggestion, express thank or interest, or

even complain about the project. Effective communication between the participants in an OSS project can lead to a more effective defect-fixing process because effective communication results in the creation of common knowledge among the participants in this process, as well as the sharing of knowledge among them (Subramanian & Soh 2006). The capability of a project to produce and communicate knowledge has been reported as a scarce, valuable, scarce, non-imitable and non-substitutable resource that can lead to strategic competitive advantage (Peteraf 1993; Tucker et al. 1996). On the basis of the foregoing, we formulate the following hypothesis:

**H5.** Effective communication has a positive effect on the effectiveness of the defect-fixing process.

Figure 3 illustrates the interaction between the three categories of resource that are involved in the defect-fixing process of OSS projects. As Figure 3 shows, team developers and user contributors are two human resources that work on software releases (as a technical resource) to identify and fix the potential defects. Such human resources normally communicate with each other (as an organizational resource) through project mailing lists and forums to share knowledge, plan, or coordinate. Project popularity is another organizational resource, which represents size of the user community for an OSS project. Community users use the software releases and, as a result, they might identify and submit defects. They can also become contributors over time.

**Figure 3. Resources involved in the defect-fixing process of OSS projects**

#### **4. Study Design**

The present paper seeks to discover relationships between OSS project outcome in terms of defect-fixing effectiveness and its proposed antecedents (project resources) by careful empirical observations, accurately quantifying the concepts and analyzing the measures. This is consistent with positivist epistemology, where the ultimate purpose is to explain relationships between concepts and objects (Neuman 2006). Several researchers have attempted to objectively represent concepts such as OSS project outcome and its determinants, and have calculated their numerical value by proposed

measures. As a result, adapting the same approach makes it possible for this study to apply the previously used measures directly.

#### **4.1 Sampling**

Researchers working on OSS development operate in a virtual environment, collecting their data from electronic records and less often from the actual people who generate them. As mentioned previously, the setting chosen for this research is the largest OSS repository in the world, Sourceforge.net.

As of May 2011, Sourceforge has 260,000 registered projects and more than 2.7 million registered developers (Source: [www.sourceforge.net/about](http://www.sourceforge.net/about)). Sourceforge divides OSS projects into various categories, such as communication, desktop, scientific/engineering, and software development. In order to increase the generalizability of the results, we took our sample from a number of different categories. This was in accordance with previous research that studied projects hosted on OSS repositories, including Sourceforge (e.g., Stewart et al. 2006; Long 2006a; Stewart et al. 2005). Collecting data from projects in all categories was not possible, given the available time and resources; hence, we focused on five categories: communication, security, software development, scientific/engineering, and games and entertainment. *Scientific/Engineering* and *Game* on the one hand, and *Communication* and *Security* on the other, were chosen because they show different extremes in terms of the range of users. For example, Scientific/Engineering software is generally used by engineers and scientists for a relatively narrow set of engineering and scientific purposes, whereas Communication software programs might be used by a broader range of users (anyone, including engineers and scientists) for a wide range of purposes. The category Software Development sits somewhere between the two extremes. Projects in this category are typically programming tools for developers. Given the range of use and users of projects in these categories, selecting from them should allow limited generalization of the results.

In order to narrow our sample, we then imposed restrictions as follows: we excluded projects that had not had any releases within the last 2 years, to discard inactive projects; we excluded projects whose development status was planning, pre-alpha, or alpha, because such projects do not normally have any

software releases; we excluded projects whose development status was mature, because such projects normally have fewer defects and hence have less activity; we focused on those projects that had had at least five records in their defect-tracking system; we focused only on projects that had adopted Sourceforge tools for communication (e.g., forums and mailing lists) and excluded projects that used external forums or mailing lists; and we excluded projects that had not assigned their defect reports for any reason like having only one developer in the development team. Moreover, we excluded projects smaller than 1 Megabytes and larger than 10 Megabytes to control for project size<sup>1</sup>.

Our sampling criteria were satisfied by a total of 427 projects in these five categories. Table 1 shows some demographic information for the projects. We collected data on all the variables in the research model, including control factors from the data publicly available on the projects' website.

**Table 1. Distribution of the projects in terms of number of downloads**

## **4.2 Measurement**

Herein, we measured defect-fixing effectiveness by percentage of the defects that have been fixed (Garousi 2009). In order to compute this percentage, we had to collect data on total number of defects submitted as well as the number of defects resolved. Developer interest was measured by the number of developers registered on the project profile as team members (Chandrasekar Subramaniam et al. 2009). User contribution to the defect-fixing process was operationalized by the number of defects that has been assigned to and resolved by users. In order to measure the release frequency, we first calculated the average number of files that a given project had released per month. Borrowed from Long (2006a), we then divided the average release frequency by the maximum release frequency of the 427 projects to get a standardized value between 0 and 1. The number of times an OSS product

---

<sup>1</sup> Larger OSS applications require more coding activities and therefore may need more developers. It could be possible that a small project consists of a small number of developers while a large project may have more developers. Moreover, as larger software is more likely to have greater functionality, it is more likely to attract a higher user interest. Therefore, in this paper, we focused on projects of similar size. According to Koch (2009), there are two ways to measure product size. First, by calculating the size in bytes of software offered, and, second, by counting the number of lines-of-code from the project's source code. Given the large number of projects in my sample, the latter approach was beyond the available time and resources; thus, we decided to take the first approach. However, because product size in bytes is to a large extent affected by documentation, graphics and similar kinds of files, we excluded the bytes related to such files (Koch 2009).

had been downloaded was used as a proxy to the project's popularity (Midha 2007). Finally, the sum of the number of messages posted on a project's mailing lists and the messages posted on the project's forums was calculated and used to measure the organizational communication (Subramanian & Soh 2006). Table 2 summarizes the indicators used to operationalize the variables under study.

**Table 2. Measures for the model variables**

According to previous research, three of the variables in the research model (project license (Stewart et al. 2006), project audience (Crowston & Scozzi 2002), and programming language (Crowston & Scozzi 2002) promote OSS positive project outcomes, such as project success; hence, they might affect defect-fixing effectiveness. However, we were not interested in their effect, so we controlled for them. A categorical variable was used for *licence restrictiveness* (Stewart et al. 2006): 0 = restrictive licence (GNU GPL), 1= non-restrictive licences (anything except GNU GPL). *Project audience* is a dummy variable that is set to 1 if the project is targeted towards developers and is set to zero otherwise (Crowston & Scozzi 2002). *Programming language* is also a dummy variable that is set to 1 if the project uses a popular programming language (e.g., C, C++, C#, and Java), and zero otherwise (Chandrasekar Subramaniam et al. 2009).

### **4.3 Data Collection**

Data on the measures of defect-fixing effectiveness (dependent variable) and user contribution (an independent variable) were manually extracted via an advanced search on the projects' defect-tracking systems by using the Sourceforge search facility (See Appendix A). For example, in order to measure the number of the 'defects resolved by users', as the indicator of the contribution by users, we undertook the following process. First, the number of the defects resolved by team members was calculated. To do so, we selected 'Bug' as 'tracker', "Closed" as the bug status and highlighted all team members as "assignee". Then, the advanced search retrieved the defect reports that had been resolved by team members. Second, we selected "Bug" as tracker and "Closed" as the bug status and retrieved the total number of defects resolved. Third, we subtracted the number of the 'defects



resolved by team' from the total number of defects that had been resolved to calculate the number of the 'defects resolved by users'. As indicated in Table 2, data on the other measures (e.g., number of developers and project license) were extracted directly from the project profile page on Sourceforge. This was in line with the recommendation of Howison and Crowsto (2004) on avoiding to use web-spiders or web-parsers to collect data from Sourceforge.

## **5. Data Analysis and Result**

We selected multiple regression analysis as an appropriate method for data analysis. SPSS (version 17) were used for data analysis. In this research, because our variables were highly left skewed, we normalized them by computing their natural log (Crowston & Scozzi 2002). All other assumptions required to conduct regression analysis were met.

The model explained 43.4% ( $R^2 = 0.434$ ) of the variance in the measure of defect-fixing effectiveness. As indicated in Figure 4, all five hypotheses were supported. These results are consistent with expectations from the RBV theory. A significant ( $p < 0.001$ ) and positive direct effect (coefficient = 0.176) was found between developer interest and defect-fixing effectiveness (H1). This supports that the skills and expertise of individual developers within an OSS project can be viewed as a valuable resource, and that the interest that developers have in an OSS project can constitute a valuable resource for the project to gain competitive advantage. Moreover, user contribution significantly ( $p < 0.001$ ) affected (coefficient = 0.261) defect-fixing effectiveness (H2). This supports that innovations and contributions by user community with regards to an OSS project's defect-resolution process can constitute a valuable resource for the project to gain competitive advantage.

### **Figure 4. The results of the research model**

The analysis indicated that greater release frequency was positively and significantly ( $p < 0.001$ ) associated with defect-fixing effectiveness (coefficient = 0.115). Thus, H3 was supported.

As predicted in H4, project popularity was positively and significantly ( $p < 0.001$ ) related to defect-fixing effectiveness (coefficient = 0.307). This indicates that OSS projects that are more popular than others are more likely to have effective defect-fixing activities. H5 predicted that organizational communication would contribute to a more effective defect-fixing process in the project. A significant effect ( $p < 0.001$ ) for organizational communication on defect-fixing effectiveness supported H5 (coefficient = 0.175).

As for control variables, a latent construct was formed with control variables as its formative measures. The relationship between that latent variable and the dependent variable was insignificant at a 0.05 level (coefficient = -0.025, t-value = 1.11).

Finally, the moderating effect of project category on the five hypotheses was examined to determine whether the results are significantly different across the five project categories. None of the moderating effects was significant ( $p < 0.05$ ), which means that the results can be generalized across the five project categories.

## **6. Discussion**

### **6.1 Contributions to Research and Implications for Theory**

In light of the insights from this research, we would like to raise a number of implications for the research community. Firstly, previous researchers have used project activity, user interest in an OSS, and developer interest as indicators for positive outcomes of OSS projects (Subramaniam et al. 2009; Stewart et al. 2006; Crowston & Scozzi 2002). Our study adds process effectiveness (i.e. the effectiveness of the defect-fixing process) to the list of indicators for positive outcomes in an OSS environment. The results of our study thus indicate that it may be beneficial to be more innovative in identifying measures of positive outcomes for OSS projects (e.g., defect-fixing effectiveness), in addition to using the traditional measures, such as the number of downloads or project activity.

Secondly, we used the resource classification proposed by Barney (1991) (human capital resources, organizational capital resources, and physical capital resources (technical resource) to conceptualize potential determinants of defect-fixing effectiveness. A further contribution of the research reported herein is therefore the operationalization of these three resource categories for the defect-fixing process of OSS projects. Developers' interest in and users' contribution to the defect-fixing process were used to operationalize human capital resources. Technical/physical capital resources were operationalized by release frequency, and organizational capital resources was operationalized by project popularity and organizational communication. A promising avenue of research may be to use other classifications of resources that have been proposed in the literature to examine the relationship between defect-fixing effectiveness and its antecedents in an OSS environment.

Thirdly, we used the RBV to postulate the relationship between the effectiveness of the defect-fixing process and its antecedents in OSS projects. The results showed that all of the five antecedents that were proposed in this study: (developer interest, user contribution, release frequency, project popularity, and organizational communication) affect defect-fixing effectiveness significantly and positively. Specifically, the results suggest that in order for OSS projects to have a more effective defect-fixing process, they need to:

- attract a higher level of developer interest (examples of the strategies to achieve this includes increasing software modularity, decreasing software complexity, using open source operating systems, using popular programming languages (e.g. C, C++ and C#));
- attract a higher level of user contribution (examples of the strategies to achieve this includes choosing less restrictive licenses, having sponsors, having frequent releases);
- update and release project files more frequently (examples of the strategies to achieve this includes using a copy-left license, and having more effective project management practices like HR staffing and compensation management);
- become more popular with the software community (examples of the strategies to achieve this includes finding sponsors, frequent release and announcements, increasing project activity);

- communicate more effectively (examples of the strategies to achieve this includes responsiveness to messages received on the project's mailing lists and forums).

## **6.2 Lessons for Practitioners**

The dimension of defect-fixing effectiveness corresponds nicely to the practitioners' concerns regarding OSS, in that responsiveness to customer needs is one of the most frequently cited concerns of IT practitioners who adopt OSS (Golden 2004). Thus, the results of our study have several implications for organizations that are interested in adopting OSS products. Firstly, human capital resources (developer interest and user contribution) were found to contribute significantly to defect-fixing effectiveness. In light of this, organizations would be well-advised to adopt software from OSS projects that have a strong base of registered developers and those that receive strong contributions from users for resolving their defects.

Secondly, the technical capital resource (the release frequency) was also found to promote effective defect-fixing. Hence, release frequency can constitute another selection criterion for organizations that are considering using an OSS. In this study, release frequency was examined in part because of its practical relevance. Release management is under the control of project team. Taking into account the results of this study, we advise OSS project managers that updating and releasing project files more frequently can significantly contribute to the success of their defect-fixing activities.

Thirdly, we found that organizational capital resources (project popularity and organizational communication) make a strong contribution to defect-fixing effectiveness. The results showed that the more popular an OSS project, the more effective the defect-fixing process. One reason for that could be the fact that projects that are more popular (i.e. have been downloaded more) can attract a larger pool of community users to identify and report the product defects. Effective communication between the participants in an OSS project was also found to contribute to more effective defect-fixing. In light of these results, prospective OSS adopters would be well-advised to adopt software from OSS projects with a higher level of popularity (measured by download rate) and those that have a larger amount of organizational communication through their mailing lists and forums.

Fourthly, the results of this study show that among the project resources proposed, project popularity has the strongest predictive value in explaining defect-fixing effectiveness. This stresses the important role of project popularity in impacting defect-fixing effectiveness. User contribution was found the second strongest predictor of defect-fixing effectiveness. This highlights the important role of user community for OSS project's defect-fixing process. Developer interest and organizational communication were found to have almost equal power to explain defect-fixing effectiveness, while release frequency was found to be the weakest predictor of defect-fixing effectiveness. The implication of these results for OSS project teams is that they should give priority to strategies that contributes to those factors that have higher predictive value in explaining defect-fixing effectiveness.

Fifthly, nowadays, many multinational companies such as IBM actively take part in OSS projects. The results of this study have implications for executives of companies faced with the decision of which OSS project to sponsor – either by funding or by allocating the company's developers, or both. The executives of such companies certainly seek to sponsor an OSS project that has a greater chance of project effectiveness (Grewal et al. 2006). To achieve this, our results advise these companies to sponsor OSS projects that: (1) attract a higher level of developer interest, (2) attract a higher level of user contribution, (3) update and release project files more frequently, (4) become more popular with the software community, and (5) communicate more effectively.

Finally, the dependent construct of this research (i.e. defect-fixing effectiveness) provides organizational users who want to evaluate and compare OSS projects with a criterion on which to base their analysis. One such criterion shows the extent to which an OSS project's team values quality; so the measurement model proposed in this study can be used to generate and compare overall defect-fixing effectiveness for different projects.

## **7. Limitations**

Sourceforge offers ample accessible data from OSS projects, making it an attractive data set for information systems researchers. There are limitations to the use of this information though; for instance the measures do not always match perfectly the concepts that are used in the hypotheses

proposed by researchers (Crowston & Scozzi 2002). However, instead of capturing people's perceptions of the concepts, such data represents real practice, "while this process [using data on Sourceforge to measure the concepts] adds noise, making it harder to find reliable relationships, it should not add bias, meaning that the relationships we [the researchers] do find are 'true' and not artefacts of the testing process." (Crowston & Scozzi 2002) (p.7). There might be other variables, such as project developers' skills or the quality of organizational communication, that cannot be captured by the data that Sourceforge provides for researchers, but that into which in-depth qualitative research might provide insight. We also acknowledge that the list of antecedents to defect-fixing effectiveness that we used in the study is by no means exhaustive.

A further limitation of the study is that the sample might be biased. Sourceforge started in 1999, but there are many older, well-established and well-known OSS projects. These projects are important for study purposes, but they are not hosted on Sourceforge. For instance, the main Linux development team has not opted to use Sourceforge, even though some smaller supporting projects have. There are also many successful OSS projects that are hosted on Sourceforge but use their own defect-tracking system rather than the tracking system that Sourceforge has provided for them. These projects are also not included in our sample decreasing the generalizability of the study.

Every OSS project has limited resources in terms of developers' time and effort, thus project developers are not normally able to address all the requested defects. That is where defect selection and prioritization comes to the play and shows its significance. Therefore, taking task prioritization into account is important when studying the defect-fixing process of OSS projects. This research initially sought to address this, however, we noticed that the majority of the OSS projects hosted on Sourceforge fail to properly adopt the feature that Sourceforege offers to them to manage their task prioritization. On the other hand, excluding those projects which have not used this feature would have left us with a handful of projects which would not be enough for the purpose and statistical analysis of this study. We acknowledge this as one of the limitations of this paper, and call for future qualitative research that conducts multiple case studies and deeply study few OSS projects taking into account of task prioritization for their defect-fixing.

External validity means “to what populations, settings, treatment variables, and measurement variables can an effect be generalised [p.5]” (Campbell & Stanley 1963). In this research, we selected our sample to projects in five categories of communication, security, software development, scientific/engineering, and games and entertainment. This would allow us to generalize our results to projects in these five categories. Although the fact that projects in the five categories from which we sampled show different extremes in terms of the range of users and this might allow limited generalization of the results, the findings might not be generalizable across all project categories. Further research might contribute to increasing the generalizability of our results by sampling from more project categories.

## **8. Conclusions**

We have reported on a study of the effectiveness of the defect-fixing process as a source of competitive advantage for OSS projects. Using RBV as a theoretical perspective, we examined the relationships between the project resources required for the defect-fixing process and its effectiveness in an OSS environment. An analysis of data collected from 427 OSS projects confirmed that attracting a higher level of developer interest in and user contribution to the defect-fixing process has a significant and positive effect on the effectiveness of the defect-fixing process. The frequent updating and releasing of the project files, effective organizational communication, and strong project popularity were also found to promote effective defect fixing.

## **References**

- Aksulu, A., & Wade, M., 2010. A comprehensive review and synthesis of open source research. *Journal of the Association for Information Systems*, 11 (11).
- Barney, J., 1991. Firm Resources and Sustained Competitive Advantage. *Journal of Management*, 17(1), p.99.
- Barney, J., 1995. Looking inside for competitive advantage. *Academy of Management Executive*, 9(4), pp. 49-61.
- Barney, J.B., 2002. *Gaining and sustaining competitive advantage*: 2nd ed. Upper Saddle River, NJ:

Prentice-Hall.

- Barney, J.B., & Clark, D.N., 2007. *Resource-based theory: Creating and sustaining competitive advantage*. Oxford, UK: Oxford University Press.
- Bharadwaj, A.S., 2000. A Resource-Based Perspective on Information Technology Capability and Firm Performance: An Empirical Investigation. *MIS Quarterly*, 24(1), pp.169-196.
- Bogers, M., Afuah, A. & Bastian, B., 2010. Users as Innovators: A Review, Critique, and Future Research Directions. *Journal of Management*, 36(4), pp.857 -875.
- Campbell, D.T. & Stanley, J.C., 1963. *Experimental and Quasi-Experimental Designs for Research*. Rand McNally College Publishing Company, Chicago
- Capiluppi, A., Lago, P. & Morisio, M., 2003. Characteristics of open source projects. In the 7th European Conference on Software Maintenance and Reengineering. pp. 317-327.
- Carillo, K. & Okoli, C., 2008. The Open Source Movement: A Revolution In Software Development. *Journal of Computer Information Systems*, 49(2), pp.1-9.
- Chengalur-Smith, S. & Sidorova, A., 2003. Survival of open-source projects: A population ecology perspective. In the 24th International Conference of Information Systems. Atlanta, GA.
- Colazo, J., 2007. *Innovation success: An empirical study of software development projects in the context of the open source paradigm*. Canada: The University of Western Ontario (Canada).
- Colazo, J. & Fang, Y., 2009. Impact of license choice on Open Source Software development activity. *Journal of the American Society for Information Science and Technology*, 60(5), pp.997-1011.
- Conley, C., 2008. *Design for quality: The case of open source software development*. United States -- New York: New York University, Graduate School of Business Administration.
- Crowston, K., Annabi, H. & Howison, J., 2003. Defining open source software project success. In the 24th International Conference on Information Systems. Seattle, WA.
- Crowston, K., Howison, J. & Annabi, H., 2006. Information systems success in free and open source software development: theory and measures. *Software Process: Improvement and Practice*, 11(2), pp.123-148.
- Crowston, K. & Scozzi, B., 2008. Bug Fixing Practices within Free/Libre Open Source Software Development Teams. *Journal of Database Management*, 19(2), pp.1-30.
- Crowston, K. & Scozzi, B., 2002. Open source software projects as virtual organisations: competency rallying for software development. *IEE Software Proceedings*, 149(1), pp.3-17.
- Daft, L.R., 1983. *Organizational Theory and Designs*, St. Paul: West Pub. Co.
- Ezzy, D., 2002. *Qualitative Analysis: Practise and Innovation*, Routledge, London, UK.
- Garousi, V., 2009. Evidence-Based Insights about Issue Management Processes: An Exploratory Study. In the International Conference on Software Process. Vancouver, Canada.
- Gefen, D. & Straub, D., 2005. A Practical Guide To Factorial Validity Using PLS Graph: Tutorial And Annotated Example. *Communications of the Association for Information Systems*, 6,



pp.91-109.

- Ghapanchi, A.H. & Aurum, A., 2011. Measuring the Effectiveness of the Defect-Fixing Process in Open Source Software Projects. In The 44th Hawaii International Conference on System Sciences. Hawaii.
- Ghapanchi, A., & Aurum, A., 2012. Determinants of Project Performance: Case of Open Source Software Projects. *International Journal of Project Management*, 30(4), pp. 407-417.
- Giuri, M. et al., 2004. Skills and openness of OSS projects: Implications for performance (working paper). Available at: [http://kb.cospa-project.org/retrieve/1454/giuri\\_et\\_al.pdf](http://kb.cospa-project.org/retrieve/1454/giuri_et_al.pdf).
- Golden, B., 2004. *Succeeding with open source*, Addison-Wesley.
- Grewal, R., Lilien, G.L. & Mallapragada, G., 2006. Location, location, location: how network embeddedness affects project success in open source systems. *Management Science*, 52, pp. 1043-1056.
- Hahn, J. & Zhang, C., 2005. An exploratory study of open source projects from a project management perspective. Available at: [http://www.krannert.purdue.edu/academics/mis/workshop/hz\\_110405.pdf](http://www.krannert.purdue.edu/academics/mis/workshop/hz_110405.pdf) [Accessed October 22, 2010].
- Hair, J. et al., 2006. *Multivariate Data Analysis*, New Jersey: Pearson Education Inc.
- Herbsleb, J.D. & Mockus, A., 2003. An empirical study of speed and communication in globally distributed software development. *Software Engineering, IEEE Transactions on*, 29(6), pp.481-494.
- Hofer, C. & Schendel, D., 1978. *Strategy Formulation: Analysis and Concepts*, St. Paul: MN: West Publishing.
- Howison, J., & Crowston, K., 2004. The perils and pitfalls of mining SourceForge Mining Software Repositories Workshop at the International Conference on Software Engineering (ICSE). Edinburgh, Scotland.
- Hulland, J., Wade, M. & Antia, K., 2007. The Impact of Capabilities and Prior Investments on Online Channel Commitment and Performance. *Journal of Management Information Systems*, 23(4), pp.109-142.
- Jugdev, K., Mathur, G., & Fung, T.S. 2007. Project management assets and their relationship with the project management capability of the firm. *International Journal of Project Management*, 25 (6), pp. 560–568
- Kavanagh, J.F., 2004. RESISTANCE AS MOTIVATION FOR INNOVATION: OPEN SOURCE SOFTWARE. *Communications of AIS*, 2004(13), pp.615-628.
- Krishnamurthy, S., 2002. Cave or Community?: An Empirical Examination of 100 Mature Open Source Projects. *SSRN eLibrary*. Available at: [http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=667402](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=667402) [Accessed October 22, 2010].
- Lavie, D., 2006. The competitive advantage of interconnected firms: an extension of the resource-based view. *Academy of Management Review*, 31(3), pp. 638-658.
- Long, J., 2004. *Understanding the creation and adoption of information technology innovations: The*

- case of Open Source software development and the diffusion of mobile commerce*. Texas: The University of Texas at Austin.
- Long, Y., 2006. *Social structure, knowledge sharing, and project performance in open source software development*. Nebraska: The University of Nebraska - Lincoln.
- Lopes, A.B. & Galletta, D., 1997. Resource-based theory and a structural perspective of strategy applied to the provision of internet services. In the Third Americas Conference on Information Systems. Indianapolis, IN.
- Midha, V., 2007. *Antecedent to the success of open source software*. North Carolina: The University of North Carolina at Greensboro.
- Midha, V. et al., 2010. Improving Open Source Software Maintenance. *Journal of Computer Information Systems*, 50(3), pp.81-90.
- Mockus, A. & Weiss, D., 2008. Interval quality: relating customer-perceived quality to process quality. In *the 30th international conference on Software engineering*. Leipzig, Germany: ACM, pp. 723-732.
- Nelson, M.L., Sen, R. & Subramaniam, C., 2006. UNDERSTANDING OPEN SOURCE SOFTWARE: A RESEARCH CLASSIFICATION FRAMEWORK. *Communications of AIS*, 2006(17), pp.2-37.
- Neuman, W.L., 2006. *Social Research Methods 6th ed.*, Pearson international edition.
- Paulson, J.W., Succi, G. & Eberlein, A., 2004. An Empirical Study of Open-Source and Closed-Source Software Products. *IEEE Transactions on Software Engineering*, 30(4), pp.246-256.
- Peteraf, M.A., 1993. The cornerstones of competitive advantage: A resource-based view. *Strategic Management Journal*, 14(3), pp.179-191.
- Ray, G., Barney J.B. & Muhanna W.A., 2003. Capabilities, business processes, and competitive advantage: choosing the dependent variable in empirical tests of the resource-based view. *Strategic Management Journal*, 25(1), pp. 23-37.
- Raymond, E., 1999. *The Cathedral and the Bazaar*, Sebastopol: CA: O'Reilly & Associates.
- Scacchi, W., 2007. Understanding the Development of Free ECommerce/ E-Business Software: A Resource-Based View. In *Emerging Free/Open Source Software Practices*. Hershey, PA: IDEA Group Publishing, pp. 170-190.
- Sen, R., 2007a. A Strategic Analysis of Competition Between Open Source and Proprietary Software. *Journal of Management Information Systems*, 24(1), pp.233-257.
- Sen, R., 2007b. WAITING FOR USABLE OPEN SOURCE SOFTWARE? DON'T HOLD YOUR BREATH! *Communications of AIS*, 2007(20), pp.382-392.
- Song, M. & Montoya-Weiss, M.M., 2001. The Effect of Perceived Technological Uncertainty on Japanese New Product Development. *The Academy of Management Journal*, 44(1), pp.61-80.
- Srivastava, R.K., Fahey, L. & Christensen, H.K., 2001. The resource-based view and marketing: The role of market-based assets in gaining competitive advantage. *Journal of Management*, 27(6), pp.777 -802.

- Stewart, K.J., Ammeter, A.P. & Maruping, L.M., 2005. A Preliminary Analysis of the Influences of Licensing and Organizational Sponsorship on Success in Open Source Projects. In the 38th Annual Hawaii International Conference on System Sciences.
- Stewart, K.J., Ammeter, A.P. & Maruping, L.M., 2006. Impacts of License Choice and Organizational Sponsorship on User Interest and Development Activity in Open Source Software Projects. *Information Systems Research*, 17(2), pp.126-144.
- Stewart, K. & Gosain, S., 2006a. The impact of ideology on effectiveness in open source software development teams. *MIS Quarterly*, 30(2), pp.291-314.
- Stewart, K. & Gosain, S., 2006b. The moderating role of development stage in free/open source software project performance. *Software Process: Improvement and Practice*, 11(2), pp.177-191.
- Straub, D., Boudreau, M.C. & Gefen, D., 2004. Validation Guidelines For IS Positivist Research. *Communications of the Association for Information Systems*, 13, pp.380-427.
- Subramaniam, C., Sen, R. & Nelson, M.L., 2009. Determinants of open source software project success: A longitudinal study. *Decision Support Systems*, 46(2), pp.576-585.
- Subramaniam, A.M. & Soh, P.H., 2006. Knowledge Integration and Effectiveness of Open Source Software Development Projects. In The Tenth Pacific Asia Conference on Information Systems. pp. 914-925.
- Sundaram, S. et al., 2007. Technology use on the front line: how information technology enhances individual performance. *Journal of the Academy of Marketing Science*, 35(1), pp.101-112.
- Tarafdar, M. & Gordon, S.R., 2007. Understanding the influence of information systems competencies on process innovation: A resource-based view. *The Journal of Strategic Information Systems*, 16(4), pp.353-392.
- Tucker, M.L., Meyer, G.D. & Westerman, J.W., 1996. Organizational Communication: Development of Internal Strategic Competitive Advantage. *Journal of Business Communication*, 33(1), pp.51 -69.
- Vergin, R.C. & Qoronfleh, M.W., 1998. Corporate Reputation and the Stock Market. *Business Horizons*, 41(1), pp.19- 26.
- Wade, M. & Hulland, J., 2004. Review: The Resource-Based View and Information Systems Research: Review, Extension, and Suggestions for Future Research. *MIS Quarterly*, 28(1), pp.107-142.
- Weill, P. & Broadbent, M., 1998. *Leveraging the New Infrastructure: How Market Leaders Capitalize on Information Technology*, Boston, MA: Harvard Business School Press.