Information Systems Management, 32:285–298, 2015

ISSN: 1058-0530 print / 1934-8703 online
DOI: 10.1080/10580530.2015.1079999

Taylor & Francis
Taylor & Francis Group

A Longitudinal Study of the Impact of Open Source Software Project Characteristics on Positive Outcomes

Amir Hossein Ghapanchi[1] and Madjid Tavana[2,3]

[1]*School of Information and Communication Technology, Griffith University, Queensland, Australia*
[2]*Business Systems and Analytics Department, La Salle University, Philadelphia, Pennsylvania, USA*
[3]*Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, Paderborn, Germany*

This article formulates and tests a set of hypotheses about the success of open source software projects with respect to *market penetration* and *human resource attraction*. The authors collected data from 1025 open source software projects in a longitudinal study. The cross-sectional results show that the extent of a project's operating systems, the range of translated languages, programming languages, and project age positively impact OSS projects' positive outcomes with respect to market penetration and human resource attraction.

Keywords human resource attraction; longitudinal study; market penetration; open source software

INTRODUCTION

A wide range of organizations from private to public have adopted open source software (OSS) products (Carillo & Okoli, 2008; Lundell, Lings, & Syberfeldt, 2011; Spinellis & Giannikas, 2012). Although OSS development has become a reliable alternative to proprietary software (Paulson, Succi, & Eberlein, 2004), Fuggetta (2003) argues that it is not proven that open source uniquely and necessarily causes software to be better, more reliable, or cheaper to develop. The failure of so many OSS projects has also become a matter of concern (Qua, Yangb, & Wanga, 2011; Wangba, Hub, & Shankerc, 2012). In fact, despite the increasing adoption of OSS, many OSS projects fail in the early stages of development (Aksulu & Wade, 2010). This failure is caused by various factors such as the inability to attract volunteer developers to join their development team or the inability to attract voluntary contribution from the user community (Subramaniam, Sen, & Nelson, 2009).

Address correspondence to Madjid Tavana, Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, 1900 W. Olney Ave., Philadelphia, PA 19141, USA. E-mail: tavana@lasalle.edu

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uism.

OSS projects' ability to produce cheaper, high quality software (Paulson et al., 2004) has led numerous commercial and government organizations 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 organizations have adopted OSS (Gold, 2007).

Various factors have been reported in the literature as antecedents to OSS project success including project audience and project topic (Crowston & Scozzi, 2002), sponsorship (Stewart, Ammeter, & Maruping, 2006), project licensing (Subramaniam et al., 2009), software complexity (Midha, Singh, Palvia, & Kshetri, 2010), the reputation of participants (Crowston & Scozzi, 2002), and project popularity and developer activity (Midha & Palvia, 2012). Although the impact of some project characteristics on OSS success has been investigated in the current literature, to the best of the authors' knowledge, no study has investigated the impact of such relationships over time. In response to this gap, the authors focus on the following questions: "What project characteristics influence the success of OSS projects with respect to market penetration and human resource attraction? And, do these impacts change over time?"

This article seeks to significantly contribute to the body of knowledge on OSS projects by revealing what impact the identified OSS projects' characteristics have on projects' market penetration and human resource attraction. This study also contributes to the OSS research by (1) examining the potential relationships between the three indicators of success that have been identified in the current OSS literature (i.e., market penetration, human resource attraction, and project activity), and (2) studying a considerable number of OSS projects over time, rather than conducting a cross-sectional study.

The remainder of this article is structured as follows. The next section provides some background information, and reviews the literature on OSS success. Then, the research model and theoretical justifications for the relationships proposed are

presented. Next, the research design is described. Following that, the data analysis and the results of the study are presented. Subsequently, the results are discussed, implications for research and practice are derived, and research limitations and future works are presented.

LITERATURE REVIEW

Success models for traditional software development projects frequently focus on such indicators of success as information quality, the quality of system and service, individual and organizational impacts, user satisfaction, and use (DeLone & McLean, 2003). These indicators are strongly related to the "use environment" of the software. In contrast, studies on success of OSS projects generally put a greater focus on the "development environment" (Ghapanchi, 2013, 2015; Ghapanchi & Aurum, 2011a). One reason for this difference in focus is that unlike traditional closed-source software development (CSSD), OSS projects rely mostly on volunteer efforts. Another reason is because there is a difference in the availability of information in the two cases. For CSSD projects, information about the development environment is not publicly available, whereas the "use environment" is less difficult to study. In contrast, for OSS projects, the development environment is publicly visible, whereas the "use environment" is difficult to study or even to identify (Ghapanchi & Aurum, 2012a, 2012b; Ghapanchi, Aurum, & Daneshgar, 2012; Ghapanchi, Aurum, & Low, 2011). In light of the foregoing, OSS researchers have focused mostly on measures of success that relate to the development environment, such as project team size, project activity, and task completion (Crowston, Howison, & Annabi, 2006; Stewart & Gosain, 2006). This article employs two success measures of market penetration and human resource attraction as described next.

Market Penetration

"Market penetration" is defined as the ability of an OSS project to attract community users to adopt the project's product (Midha, 2007). The interest displayed by users is critical for OSS projects because of a "direct network effect." Software falls within this category of products and services that show a direct network effect (Subramaniam et al., 2009). A direct network effect occurs when a consumer receives increasing value from using a product as the number of product consumers rises (Shy, 1996). Market penetration has been found to be a key characteristic of competence for organizations (Lettl, 2007) as well as projects (Ghanbarzadeh et al., 2014). High market penetration can impact project outcomes such as project vitality and activity (Stewart et al., 2005). Several researchers have worked on OSS projects' "market penetration" (e.g., Feller & Fitzgerald, 2002; Grewal, Lilien, & Mallapragada, 2006; Liu, 2008; Midha & Palvia, 2012; Stewart et al., 2006; Subramaniam et al., 2009).

Human Resource Attraction

OSS projects rely mostly on volunteers spending time and energy on the project to further the development. Human resource attraction means the ability of an OSS project to attract community developers to join the project's core development team. Since the development of OSS applications is contingent on volunteer developers, attracting developer interest is crucial for OSS projects. That is why human resource attraction has been frequently used as a very important aspect of OSS project success. Many OSS projects become inactive and die because they cannot attract volunteer developers to further their development activities. Several researchers have worked on OSS projects' "human resource attraction" (e.g., Chengalur-Smith et al., 2010; Stewart & Gosain, 2006; Subramaniam et al., 2009).

Literature Gap

The existing literature has investigated three broad categories of antecedents to OSS success. One category includes characteristics such as project topic, programming language, project audience, and license (Crowston & Scozzi, 2002; Stewart et al., 2006; Subramaniam et al., 2009). Another category includes characteristics such as software modularity, complexity, degree of decomposition, degree of coupling, and software quality (Liu, 2008; Midha, 2007; Purarjomandlangrudi et al., 2014). The third includes factors that are related to network structure, such as the structure within which core developers collaborate, core/periphery fitness, and network embeddedness (Grewal et al., 2006; Long, 2006).

The following gaps in the literature regarding the impact of project characteristics on project success are presented in Table 1. There is a lack of longitudinal studies in this research area. Although the effect of making projects compatible with specific operating systems (e.g., Linux) has been examined (e.g., Subramaniam et al., 2009), there is no study that examines the effect that the operating systems have on the success of the OSS projects' market penetration and human resource attraction. Albeit Midha (2007) is the only study (that the authors found) which examined the impact that the number of spoken languages a project is translated into has on OSS market penetration, and to the best of the authors' knowledge, no study has empirically investigated the effect that translation language might have on OSS market penetration. Project age is an important factor which can be used as a proxy for team members' group experience (Stewart et al., 2006). Despite its importance, project age has been mostly used as a control variable in studies on OSS success (Stewart et al., 2006; Subramaniam et al., 2009). Thus, this study examines project age as a potential antecedent to OSS projects' success. It seems as though that studies that have investigated the impact of programming language as a main variable (as opposed to a control variable) on OSS projects' market penetration and human resource attraction have only taken into account "C" and its

TABLE 1
Literature Gaps

Independent Variable	Dependent Variable	Current Literature	This Study
Operating system	Market penetration	None	Includes
Operating system	Human resource attraction	None	Includes
Translation language	Market penetration	Positive impact (Midha, 2007)	Includes
Translation language	Human resource attraction	None	Includes
Programming language	Market penetration	Positive impact for "C" languages (Crowston & Scozzi, 2002; Subramaniam et al., 2009)	Includes ("C" languages, PHP, and Java)
Programming language	Human resource attraction	Positive impact for "C" languages (Subramaniam et al., 2009)	Includes ("C" languages, PHP, and Java)
Project age	Market penetration	Used as a control variable and found no impact (Stewart et al., 2006)	Includes (as a main variable)
Project age	Human resource attraction	Used as a control variable and found a positive impact (Subramaniam et al., 2009)	Includes (as a main variable)

later versions as the favorite languages of OSS developers (e.g., Crowston & Scozzi, 2002; Subramaniam et al., 2009). However, as Midha (2007) and Rehman (2006) have argued, besides "C" and its subsequent versions, Java and PHP are also quite popular in the OSS community. Given the gaps in the literature, the authors chose operating system, translation language, programming language, and project age as OSS projects' characteristics and examined their effect on OSS success. The next section defines each characteristic and formulates the research model.

RESEARCH MODEL

Operating System

Each OSS project can choose to run on one or more operating systems. Operating system is defined as "system software responsible for the direct control and management of the hardware, basic system operations and operating the programs of application" (Balli & Korukoğlu, 2009, p. 119). The aim of operating systems is the provision of an environment for users to run their applications conveniently and efficiently (Silberschatz et al., 2004). Almost every device made with integrated circuits uses an operating system (e.g., personal computers, routers, cell-phones, internet servers, music players, etc.). Examples of operating systems are Linux, Unix, HP UX, GNU Hurd, Multics, Windows, Mac, Solaris/Sun, Net Ware, BSD, MS-DOS, Compaq/DEC VMS, and IBM Aix.

Operating systems offer diverse features and characteristics. Peng, Li, and Mili (2007) introduced two sets of attributes that characterize operating systems. This includes intrinsic technical attributes (e.g., ease of learning and ease of use) and extrinsic attributes (e.g., industrial and governmental support). One of the most important criteria for software users in selecting an operating system is ease of learning. A given user is less likely to consider an operating system unless he/she knows how to work with it. Ease of learning can be measured by the average number of hours a user needs to learn how to work with an operating system (Peng et al., 2007). Another important criterion is ease of use. Users tend to select an operating system that is easier to use (IEEE Standard Computer Dictionary, 1990).

Apart from those professional users who are experienced in using multiple operating systems, other OSS users are only familiar with one or two, which includes the operating system(s) they have experience with using in the past. When selecting an OSS application to satisfy a particular computing need, the user would tend to prefer the OSS products that can be run on an operating system with which he/she is familiar. Hence, in the interests of attracting a large number of users and developers it would be wise to develop software that is compatible with as wide a range of operating systems as possible (Midha, 2007).

The familiarity of the operating system is likely to be an important component in determining its legitimacy and its likelihood of support. Thus, the number of operating systems with which a project is compatible, would directly impact the number of users and developers who are familiar with that operating system, and, therefore, the number of the users who download the product and developers who join the project development team.

In light of the argument above, the authors formulate H1 as follows:

H1: The greater the number of operating systems with which a project is compatible, the more successful will be the OSS project with respect to (a) market penetration, (b) human resource attraction.

Translation Language

It is imperative for software producers and localization engineers to translate software applications into different spoken languages. By translating a software application into a local spoken language, people in that area are able to utilize the software with minimum effort to understand and use it (Zarei and Ghapanchi, 2008). There is a general consensus among software producers that translating a software application into users' spoken languages plays a significant role in the adoption and optimal use of the software (Osborn, 2007).

Many developing countries are still behind the curve of utilizing IT as an agent of socioeconomic development for many reasons, such as the unavailability of software in their native language. Users with a non-English native language (e.g., Portuguese, Arabic, and Spanish) tend to use software in their native languages (Ghapanchi and Aurum 2011b). This has created a good opportunity for software companies to provide various language translations for their products. For instance, Apple has translated a number of its software products into the main Indian dialects. Furthermore, the French and Spanish governments have enforced laws that require software companies to provide local versions of their software applications (Midha, 2007).

OSS projects may choose to translate their application into different spoken languages. If an OSS product is developed in a language other than a user's native language, the user has to spend extra effort to understand and use the software. This might reduce the software's perceived ease of use—"the degree to which a person believes that using a particular system would be free of effort" (Davis, Bagozzi, & Warshaw, 1989, p. 320)—and may result in a smaller user base or fewer people who join the project. This agrees with the studies in the literature that indicate that language is a significant antecedent to the effectiveness of a user interface, and a simple translation can result in a higher productivity of users (Merati et al., 2010). Midha (2007) also found a positive relationship between the number of languages into which an OSS project is translated and the level of the OSS project's market penetration.

On the basis of the foregoing, the authors formulate H2 as follows:

H2: The greater the number of languages into which a project is translated, the more successful will be the OSS project with respect to (a) market penetration, (b) human resource attraction.

Programming Language

A programming language is defined as "a set of characters, rules for combining them, and rules specifying their effects when executed by a computer" (Sammet & Hemmendinger, 2003, p. 1470). The programming language that an OSS project uses is an important feature of the project. OSS projects may choose to use any programming language for their development activities (e.g., C, C++, Java, Python, Visual basic, Lisp, etc.). "C" programming language has, for various reasons, performed a significant role since the inception of OSS development. First, "C" is a suitable option for developers when dealing with codes that need portability or quicker processing. Second, "C" is the language of system implementation for Unix, which is one of the most dominant operating systems for OSS projects. Furthermore, transcendent "C" compilers are free OSS products, which are accessible through the internet (Subramaniam et al., 2009). Thus, "C" and its later versions (e.g., C# and C++) have been very popular in the OSS community (Subramaniam et al., 2009).

Using a popular programming language will increase the number of "developer" users who may adopt an OSS project as well as the number of potential developers who may join an OSS project, because the developers need to know the programming language to be able to comprehend or even change the source code. Thus, it is expected that OSS projects that are written in popular programming languages will attract greater interest on the part of both users and developers.

A project written in a more common programming language has potentially larger source code resources. OSS projects often benefit from code re-use (Haefliger et al., 2008). Using a more common programming language enables an OSS project to take advantage of code-sharing and code re-use from many other projects (Chengalur-Smith et al., 2010). Accordingly, projects that are written in more popular programming languages are more likely to succeed.

Projects operating with a more common programming language have access to a bigger bank of developers (people with the knowledge of programming) who may download, contribute to or even join the project. Thus, such projects have higher human resource availability and are able to attract developers more easily (Crowston & Scozzi, 2002). Chengalur-Smith et al. (2010) also supports this relationship, stating, "a developer or user is likely to perceive a more familiar programming language . . . as more legitimate than an unfamiliar one, and will perceive the project that uses it as more legitimate by association, and will therefore render support to the project" (p. 664).

In the OSS literature, there is some supporting evidence for the potential impact of the use of popular programming languages on OSS project success. As an example, Crowston and Scozzi (2002) observed that OSS projects that are written in the "C" programming language are more likely to be used and remain active.

In light of the foregoing, the authors formulate H3 as follows:

H3: Projects that are written in more popular programming languages are more successful with respect to: (a) market penetration, (b) human resource attraction.

Project Age

Project age is defined as the number of months an OSS project has been in existence. Project age has been speculated to have a potential effect on the success of OSS projects (Stewart et al., 2006; Subramaniam et al., 2009), because it can be a surrogate to many positive outcomes of the project, such as the group experience of the developers (Stewart et al., 2006).

Newer projects are more likely to be perceived as less legitimate for several reasons. First, just as for other types of organizations, governance procedures are highly important in OSS projects. The fact that most OSS projects are run by developers who are geographically distributed makes these procedures even more critical. Newer projects have less perceived legitimacy because they have had less time to develop governance procedures, such as team development (Chengalur-Smith et al., 2010). Second, newer projects may not have had time to establish a framework for peer review, problem solving, documentation or requirement elicitation. Third, newer projects have had less time to create credibility for themselves as well as for people affiliated with these projects (Chengalur-Smith et al., 2010). Fourth, younger projects may not generate enough new ideas, add valuable software features, or fix the potential defects in the application (Chengalur-Smith et al., 2010). Hence, the community users may suspect the reliability of the software and therefore decide not to adopt it. A higher level of trust can contribute to higher performance for OSS projects (Stewart & Gosain, 2006). Unfortunately, newer projects are less likely to have a stronger level of trust because the project's participants have had less time to work together and to develop methods to increase trust (Chengalur-Smith et al., 2010). Finally, OSS projects operate in a broader professional, social, and business network. A strong connection with other projects might help them attract volunteer efforts, user, and developer interest (Hahn et al., 2008). However, younger OSS projects have had less time to make a strong relationship with other projects.

Given these reasons, newer OSS projects are less likely to be perceived as legitimate and reliable. As a result, newer projects are less likely to attract and retain human resources (i.e., developers and users). In light of this, the authors propose that project age has a positive impact on user interest and developer interest.

On the basis of the foregoing, the authors formulate H4 as follows:

H4: The longer a project has been running, the more likely it is to be successful with respect to (a) market penetration, (b) human resource attraction.

Developers normally contribute to OSS projects to demonstrate their programming expertise to possible employers or to receive recognition from their peers in the OSS community (Subramaniam et al., 2009). Given that successful OSS projects provide developers with greater visibility to potential employers and their peers, OSS developers tend to join more successful OSS projects. Moreover, new users might become attracted by the high level of project activity and download the project. Subramaniam et al. (2009) concluded that project activity in the current period has a positive effect on developer interest and user interest in the same period. In light of the foregoing, the authors formulate H5 as follows:

H5: Project activity in any period of time is positively associated with OSS project success in the same period in terms of (a) market penetration, (b) human resource attraction.

Figure 1 shows the research model that underlies this work.

RESEARCH METHODOLOGY

The present article seeks to discover relationships between OSS project characteristics and its success 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.

Sampling

The data for this research is collected from the largest web-based hosting portal for OSS projects, Sourceforge.net. Researchers have used Sourceforge as a main data source for empirical OSS research (Colazo & Fang, 2009; Koch & Schneider, 2002; Mockus, Fielding, & Herbsleb, 2002). As of May 2011, Sourceforge had 260,000 registered OSS projects, and it also has more than 2.7 million registered members (www.sourceforge.net).

In order to narrow the sample, the authors imposed some restrictions as follows: they excluded projects that had not had any file releases within the last 2 years (to discard inactive projects); they excluded projects whose development status was planning, pre-alpha, or alpha (because they normally don't have any software release); and they excluded mature and less active projects. Further selection criteria resulted in 1025 projects in the final sample. Figure 2 shows some demographic information on the sampled projects. The authors collected data on all the variables in the research model including control factors from the data publicly available on the projects' web site.

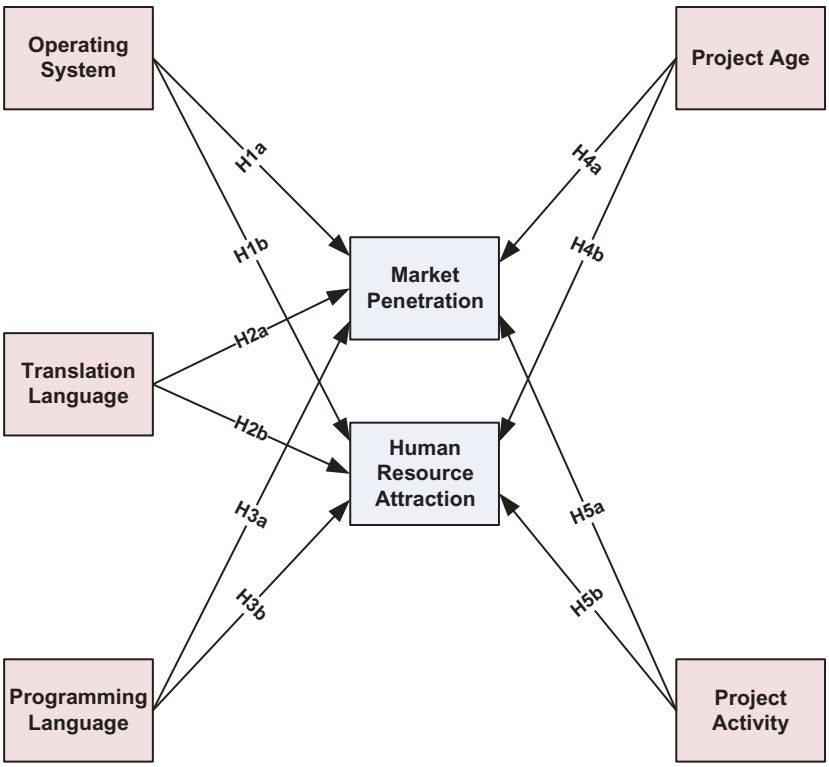

FIG. 1. Research model.

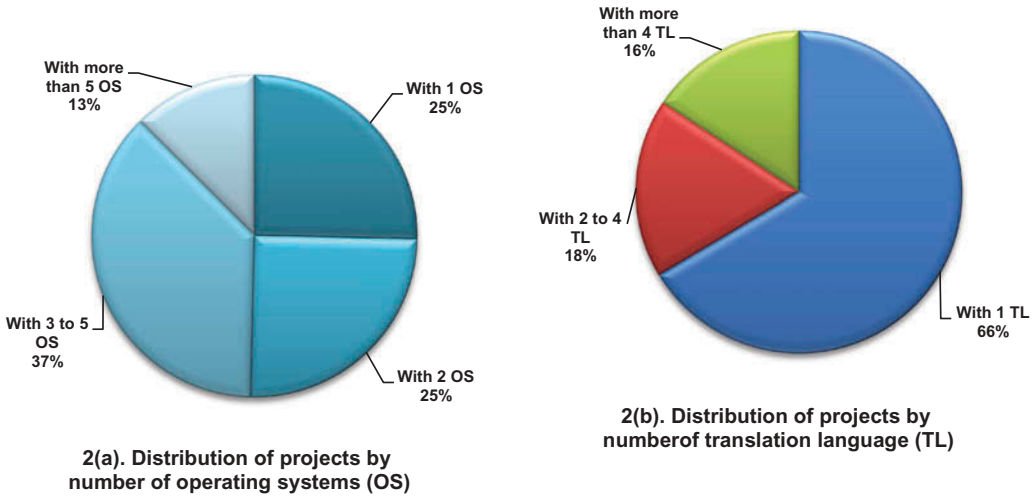

FIG. 2. Project distribution pie charts.

Measurement

The OSS projects' market penetration is operationalized by the number of times the OSS product has been downloaded (Stewart et al., 2006; Subramaniam et al., 2009). Human resource attraction is operationalized by the number of developers that are registered on the project profile as team members (Subramaniam et al., 2009). Project activity is measured by the number of code commits (Colazo & Fang, 2009) to the project's concurrent versioning system (CVS).

The project characteristics that are studied include the operating system, translation language, programming language, and age of the project. The operating system characteristic is measured by the number of operating systems that an OSS project can be run on. Translation language is measured by the number of languages into which an OSS has been translated (Midha, 2007). A dummy variable is used to measure the programming language: set to 1 if the project uses a popular programming language (e.g., C, C++, C#, Java, and PHP), and 0 otherwise

(Rehman, 2006; Subramaniam et al., 2009; Uchida et al., 2005). Project age is measured by the number of months for which a project has been running.

The authors control for various variables in the research model that might affect the success of OSS projects. One is the project audience. Crowston and Scozzi (2002) argue that projects that are targeted at developers have a higher chance of success because they have a larger pool of developer contributors. Project audience is measured by a dummy variable that is set to 1 if the project is developer targeted and 0 otherwise (Crowston & Scozzi, 2002). Another concern is whether the project uses Unix, Linux, or POSIX as an operating system. According to Subramaniam et al. (2009), being compatible with these three operating systems increases the likelihood of success.

Data Collection

Data on all the measures were extracted from the projects' web site on Sourceforge (in particular from the project profile page, the project statistics page, and the project CVS 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. Figure 3 shows the information for a sample project hosted on Sourceforge.net.

In accordance with Subramaniam et al. (2009) and Stewart et al. (2006) who have found that there is a lag time for the realization of the OSS project success, the authors examine the impacts of the success drivers on subsequent periods. For this reason, the data were collected in three snapshots: t1, t2, and t3. The time period between t1 and t2 (t1, t2) and the period between t2 and t3 (t2, t3) were both 8 months. Data on the drivers of OSS success (project characteristics) were collected twice, once at t1 (the first snapshot of data collection), and once at t2 (the second snapshot of data collection). Data on the measures of the success of the OSS projects (i.e., market penetration, human resource attraction, and project activity) were collected at all three snapshots of data collection (t1, t2, and t3). In order to calculate the value of each success measure in the period discussed, the authors subtracted the value of the success measure at the beginning of the period from the value of that at the end of the period. For example, in order to calculate market penetration in the first time period (t1, t2), the authors subtracted the number of project downloads at t1 from the number of project downloads at t2. Figure 4 illustrates the data collection process in accordance with the snapshot approach.

DATA ANALYSIS AND FINDINGS

Structural equation modeling (SEM) is a more powerful alternative to multiple regression analysis (Annacker and Hildebrandt, 2004). This study employs SEM for the data analysis because unlike regression analysis, the SEM, as a second generation approach, enables the researchers to simultaneously

FIG. 3. Sample project web page on sourceforge.net (summary page).

estimate the interrelation between multiple dependent and independent variables (Haenlein & Kaplan, 2004). This study uses structural modeling techniques using AMOS 17.0 (Debra & Abbie, 2004; Dwayne & Dorothy, 2006). In order to compare the relative importance of the predictor variables in impacting the dependent variables, the authors will use the standardized path coefficients. Standardized path coefficients will be computed using AMOS path analysis.

Model Fit

There is no best indicator of model fit in SEM, thus, it is recommended that researchers use multiple fit indices (Schermelleh-Engel, Moosbrugger, & Müller, 2003). The following indices form an adequate set of criteria which are recommended in the literature (Schermelleh-Engel et al., 2003): Goodness of fit index (GFI), adjusted goodness of fit index (AGFI), comparative fit index (CFI), and standardized root mean square residual (SRMR).

Table 2 summarizes all the model fit indices discussed earlier. According to Table 2, the research model shows a very good fit because all the fit indices have values within the recommended range. GFI and AGFI of 0.991 and 0.947 (respectively) both indicate a good fit for the model because they are in the acceptable range ($0.90 \leq \text{GFI} \leq 1$, and $0.90 \leq \text{AGFI} \leq 1$).

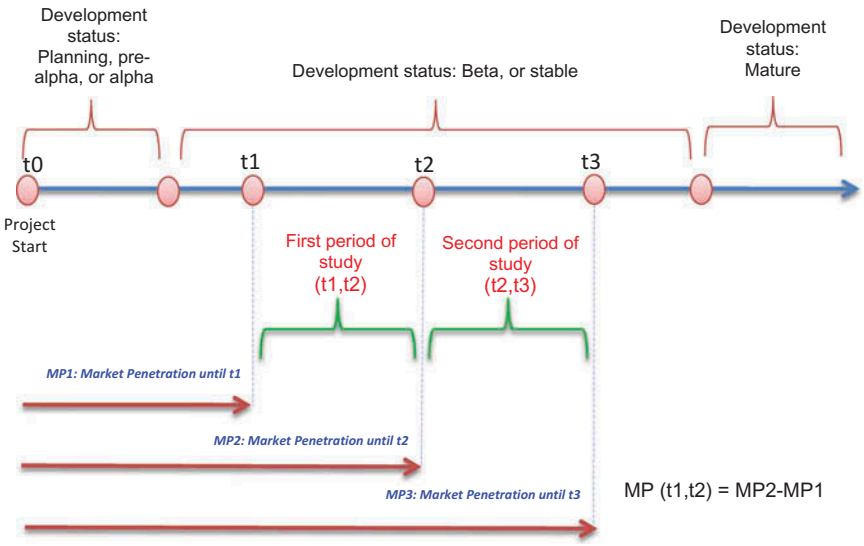

FIG. 4. Data collection: A three snapshots approach.

TABLE 2
Fit Criteria and Their Recommended Values

Fit Index	Value	Recommended Value
CFI	0.957	0.90 ≤CFI ≤ 1
GFI	0.991	0.90 ≤GFI ≤1
AGFI	0.947	0.90 ≤AGFI ≤1
SRMR	0.045	0.00 ≤SRMR ≤0.05

However, these two measures are not always reliable because they are sensitive to the sample size. The authors rather rely on SRMR and CFI because they are less sensitive to the size of the sample than the other fit indices. The CFI of 0.957 and the SRMR of 0.045 which are all within the recommended range demonstrate a good fit for the model. The research model explains approximately 22.8% (R^2 = 0.228) of the variance in user interest and 25.3% (R^2 = 0.253) of the variance in developer interest.

Cross-Sectional Results of Hypothesis Testing

Table 3 presents the cross-sectional results of the data analysis for both the time periods (t1, t2) and (t2, t3). The authors present the results of the hypothesis testing as follows:

A significant (at α = 0.001) and positive direct effect (coefficient = 0.113) was found between operating system and market penetration in the OSS project (H1a) for the first period. This impact was also positive (coefficient = 0.044) and significant (at α = 0.05) in the second period. This indicates that the more operating systems an OSS project is compatible with, the more likely it is to be adopted by community users. H1b posited that OSS projects that are designed to run on a wide range of operating systems are more likely to attract voluntary human resources. This impact was positive (coefficient = 0.065) and significant (at α = 0.01) in the first period, but, in the second period, this effect became insignificant.

The standardized parameters estimated for the impact of translation language on user interest (H2a) for the first and the second period are 0.281 and 0.22, which are both significant at the 0.001 level. This indicates that OSS projects that are translated into a greater number of languages are more likely to have a higher market penetration. H2b posited that OSS projects that have more language translations are more likely to attract voluntary human resources. The standardized parameters estimated for this impact for the first and the second period are 0.07 and 0.078 which are both significant at the 0.01 level. This demonstrates that OSS projects with more language translations will attract more human resources.

The standardized parameters estimated for the impact of programming language on market penetration (H3a) for the first and the second period are 0.039 and 0.094, which are respectively significant at the 0.1 and 0.001 level. Moreover, the standardized parameters estimated for the impact of programming language on human resources (H3b) for the first and the second period are 0.046 and 0.05, which are respectively significant at the 0.1 and 0.05 level. This implies that OSS projects that are written in the popular programming languages (e.g., "C" and its later versions, Java or PHP) may attract more human resources.

H4a predicted that the age of a project would affect the project's market penetration. Significant effects (at α = 0.001) for project age on market penetration supported H4a in both time periods (coefficient = 0.24, coefficient = 0.225). Similarly, project age was found to influence human resources attraction in both periods (coefficient = 0.258, coefficient = 0.264), which supports what was predicted in H4b.

TABLE 3
Path Coefficients and *t*-Values

	Period 1		Period 2	
Structural Path	Coefficient	*t*-Value	Coefficient	*t*-Value
H1a: Operating system → market penetration	0.113****	4.53	0.044**	2.24
H1b: Operating system → HR attraction	0.065***	2.63	0.034	1.39
H2a: Translation language → market penetration	0.281****	11.85	.22****	7.86
H2b: Translation language → HR attraction	0.07***	3.01	0.078***	3.39
H3a: Programming language → market penetration	0.039*	1.669	0.094****	3.62
H3b: Programming language → HR attraction	0.046*	1.95	0.05**	2.15
H4a: Project age → market penetration	0.24****	9.84	0.225****	9.06
H4b: Project age → HR attraction	0.258****	10.76	0.264****	11.11
H5a: Project activity → market penetration	0.248****	10.58	0.271****	11.28
H5b: Project activity → HR attraction	0.407****	17.68	0.42****	18.26

*Significant at 0.1 level; **significant at 0.05 level; ***significant at 0.01 level; ****significant at 0.001 level.

These show that older projects are more likely to be embraced by users and developers.

The results show that project activity is related positively and significantly (at $\alpha = 0.001$) to market penetration in both periods (coefficient = 0.248, coefficient = 0.271). These correspond to the assumptions of H5a. In addition, project activity was found to influence (at $\alpha = 0.001$) human resources attraction in both periods (coefficient = 0.407, coefficient = 0.42). Thus, H5a and H5b are both supported. These findings demonstrate that OSS projects that have more development activity in a time period are more likely to have higher market penetration and human resources attraction in the same period.

Longitudinal Results of Hypothesis Testing

Table 3 shows the standardized path coefficients for each relationship hypothesized in both periods. To find out which impact has significantly changed over time, the authors ran a multi-group analysis using AMOS and compared the model in period 1 with the model in period 2. The unconstrained model in the first period was treated as the baseline model, and the chi-square for this model was calculated (chi-square = 95.102, $df = 10$). Subsequently, the authors imposed equality constraints on each relationship, one at a time, and calculated the chi-square for the constrained model. The change in chi-square for each relationship was then computed by subtracting the chi-square of the unconstrained model from the chi-square of the constrained model. The change in the parameter estimate for the hypothesis under study is significant at the 0.1 level if the chi-square change is greater than 2.71. This value is 3.84, therefore it is significant at the 0.05 level. Table 4 shows these calculations for all 10 hypotheses of this study. In the following section, the authors explain which hypotheses have significantly changed over time.

Change in Operating System → Market Penetration

Although the authors' hypothesis that a higher number of operating systems attract a higher level of market penetration (H1a) is supported for both time periods, the magnitude of this impact is higher for the first period. This indicates that as the project matures the impact of operating system on market penetration is still significant but becomes weaker. One potential reason for this is that as the project matures the impact of other factors becomes stronger. This also demonstrates that users' criteria for choice of an OSS might change over time (as the project approaches the later stages of development).

Change in Translation Language → Market Penetration

The hypothesis that a higher number of language translations attracts a higher level of market penetration (H2a) is supported for both time periods. However, the magnitude of this impact is slightly lower for the second period. This indicates that as the project matures the impact of translation language on market penetration remains significant but becomes slightly weaker.

Change in Programming Language → Market Penetration

The hypothesis that using a more popular programming language attracts a higher level of market penetration (H3a) is supported for both time periods. However, the magnitude of this impact is slightly higher for the second period. One potential reason for this could be that in later stages of development, more developer users (users with knowledge of programming) might

TABLE 4
Chi-Square Test Results

Hypotheses	Chi-Square	Delta(Chi-Square—95.102)
H1a: Operating system → market penetration	98.202	3.1*
H1b: Operating system → HR attraction	95.909	0.807
H2a: Translation language → market penetration	97.922	2.82*
H2b: Translation language → HR attraction	95.156	0.054
H3a: Programming language → market penetration	98.012	2.91*
H3b: Programming language → HR attraction	95.119	0.017
H4a: Project age → market penetration	95.242	0.14
H4b: Project age → HR attraction	95.145	0.043
H5a: Project activity → market penetration	96.842	1.74
H5b: Project activity → HR attraction	95.171	0.069

*Significant at 0.1 level (chi-square change ≥ 2.71); **significant at 0.05 level (chi-square change ≥ 3.84).

potentially download an OSS (Subramaniam et al., 2009), and they certainly are concerned about the project's choice of programming language. This indicates that as the project matures the influence of programming language on market penetration not only remains significant but also becomes slightly stronger.

Validity

Data accuracy has been reported to be a potential validity concern in relation to OSS project data stored on Sourceforge, because certain project data is self-reported (e.g., translation language or operating system in this research) by project managers (Crowston & Scozzi, 2002). However, the authors believe that using OSS project data from Sourceforge is valid for their proposed research, because, according to Tirole and Lerner (2002), OSS project leaders seek to attract new developers, users and sponsors, so it is less likely for them to make "the project appear to be something other than what it really is" (p. 33), since it would not be a positive signal to new developers, users, and sponsors.

DISCUSSION AND CONCLUSIONS

This study has studied the effect of the drivers of success in OSS projects with respect to market penetration and human resource attraction. The data, which were collected from 1025 OSS projects hosted on Sourceforge.net, confirmed that operating system, translation language, programming language, and the age of the project all affect an OSS project's market penetration. All of the project characteristics except for operating system were found to influence human resource attraction. The authors also found that the impact of the factors on human resource attraction is almost static over time. The effect of operating system and translation language on market penetration was found to slightly decrease, while the impact of programming language on market penetration appeared to slightly increase over time. Finally, market penetration, human resource attraction, and project activity were found to be interrelated. The findings of this study and the incremental contribution of this article to the literature are summarized in Table 5.

Contribution to Theory

This article supplements previous research on the success of OSS projects by increasing the understanding of the effects on the success of projects characteristics. Although research on OSS success has been enormous, longitudinal studies on co-evolution of project success and its antecedents are rare, with only few exceptions (Qureshi & Fang, 2011). Therefore, this study collected longitudinal data (in three time points) to understand the co-evolution of project characteristics and the success of OSS projects.

This study complements previous research by simultaneously studying four characteristics of OSS projects: operating system, translation language, programming language, and project age. Although the effect of making projects compatible with specific operating systems (e.g., Linux) has been examined previously (e.g., Subramaniam et al., 2009), the authors are among the first to examine the effect that the range of operating systems with which an OSS project is compatible has on the success of the project. As shown in Table 5, the authors' study confirms the results of Midha's (2007) work by finding that the number of translation languages contributes to greater market penetration. The results also provide support for Crowston and Scozzi's (2002) and Subramaniam et al.'s (2009) findings that using a widely used programming language influences the OSS project's market penetration. Project age has been mostly used as a control variable in studies on OSS success (Stewart et al., 2006; Subramaniam et al., 2009); in contrast, the study reported herein used project age as a proxy for team members' group experience and found that it has a positive impact on market penetration and human resource attraction.

TABLE 5
Incremental contribution

Hypothesis	Predictor	Success Measure	Cross-Sectional Effect: Current Literature	Cross-Sectional Effect: This Article	Longitudinal Effect: Current Literature	Longitudinal Effect: This Article
Relationship between project characteristics and OSS success						
H1a	Operating systems	Market penetration	None	Positive impact	None	Slight decrease
H1b	Operating systems	HR attraction	None	No change	None	No change
H2a	Translation languages	Market penetration	Positive impact (Midha, 2007)	Positive impact	No significant change (Midha, 2007)	Slight decrease
H2b	Translation languages	HR attraction	None	Positive impact	None	No change
H3a	Programming language	Market penetration	Positive impact (Crowston & Scozzi, 2002; Subramaniam et al., 2009)	Positive impact	None	Slight increase
H3b	Programming language	HR attraction	Positive impact (Subramaniam et al., 2009)	Positive impact	None	No change
H4a	Project age t1	Market penetration	Used as a control variable and found no impact (Stewart et al., 2006)	Positive impact	None	No change
H4b	Project age t1	HR attraction	Used as a control variable and found a positive impact (Subramaniam et al., 2009)	Positive impact	None	No change
Relationship between the measures of OSS success						
H5a	Project activity	Market penetration	Positive impact (Subramaniam et al., 2009)	Positive impact	None	No change
H5b	Project activity	HR attraction	None	Positive impact	None	No change

This article is one of the few studies to have investigated the potential relationships between market penetration, human resource attraction, and project activity. As Table 5 shows, this study confirms the findings of previous research. The results presented herein provide more empirical support for Subramaniam et al.'s (2009) finding that strong project activity in any period of time has a positive effect on market penetration in the same period. In addition, the authors found that strong project activity in any given time period has a positive effect on human resource attraction in the same period.

Lessons for Practitioners

The study has several implications for corporations that are interested in adopting OSS products, administrators of OSS projects, potential OSS sponsors, OSS hosting portals, and OSS administrators who wish to start a new OSS project.

The article investigated the impacts of the following project characteristics on OSS projects' market penetration and human resource attraction: operating system, translation language, and programming language. These characteristics are all under the control of OSS project administrators. The results of the study revealed that being compatible with more operating systems, being translated into more languages, and using a widely used programming language (e.g., C, C++, C#, Java, and PHP) contributes to greater market penetration. The implication is that OSS project managers should make their software work on multiple operating systems and provide several language translations for their software. In addition, OSS administrators who wish to start a new OSS project should know that selecting a widely used programming language for the project might result in more developer users downloading their product. The implication of these results is that OSS projects that are (1) compatible with more operating systems, (2) translated into more languages, (3) built in a widely used programming language, and (4) have been running for a long time relative to other projects, are more likely to be successful.

Nowadays, several companies such as Sun Microsystems and IBM actively take part in OSS projects. The results of this study have implications for executives of companies that need to decide on 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 market penetration (Grewal et al., 2006). To achieve this, the results suggest that these companies should sponsor OSS projects that are made to run on multiple operating systems, have several language translations, are programmed in a popular programming language ("C" and its next versions, or Java) and are older.

The majority of OSS projects become inactive after a while, since developers abandon the projects. In fact, project abandonment is one of the major issues that afflict the OSS movement (Chengalur-Smith et al., 2010). The findings of this study can help OSS hosting portals such as Sourceforge.net to alleviate this situation. For example, when the OSS host is promoting successful OSS projects (e.g., the project of the month), the host could also provide some statistics on the project's number of operating systems and language translations, and its programming language. This way, the managers of other projects can become familiar with the characteristics of successful projects. OSS hosts could also recommend projects to developers on the basis of their experience, expertise, and interests.

In this research, gathering data in different time periods allowed the authors to study the influence of project characteristics in a longitudinal sense. This has several managerial advantages because it can assist OSS project managers to develop long-term strategies for their project management activities.

The longitudinal findings of this research have important implications for OSS project managers. First, the study showed that as the project matures, the impact of the operating system on market penetration stays significant but becomes weaker. Thus it would be more important for OSS project managers to make their software run on a broader range of operating systems at earlier stages of development rather than at later stages. Therefore, OSS project managers should be advised to make their application run on more operating systems as early as they can.

Second, this study concluded that as projects mature, the magnitude of the impact of translation language on market penetration remains significant but becomes slightly weaker. Thus the strategy of "translating the software into more spoken languages" in order to attract higher market penetration is relatively more effective at earlier stages of projects. Hence, OSS administrators can be recommended to translate their application into more languages as early as possible in their project life cycle.

Third, this study showed that as projects mature, the magnitude of the influence of programming language on market penetration remains significant but becomes stronger at later stages of the projects. This can be explained by the fact that developer users prefer adopting OSS applications from more mature projects as well as projects written in a popular programming language so that they can comprehend the source code; as a result, OSS projects which use popular programming languages are more likely to attract more developer users at later stages of development. A key message for project managers is that: in order to attract more developer users at later stages of their projects' life cycle, one effective strategy would be writing the application on a more popular programming language (i.e., C, C++, C#, Java, and PHP).

Fourth, 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 OSS projects that have certain characteristics. To achieve this, the authors' results advise these companies to sponsor OSS projects that (1) are compatible with more operating systems, (2) have more language translations, (3) are written in more popular programming languages, (4) and have higher project activity.

Limitations and Future Research Directions

Sourceforge offers ample accessible data from OSS projects, making it an attractive dataset for information systems researchers. There are limitations to the use of this information though; for instance the measures do not always match perfectly with the concepts that are used in the hypotheses proposed by researchers (Crowston & Scozzi, 2002). However, such data represent real practice, and 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).

Another limitation of this study is that the R-square for the research model is relatively low (0.228 for market penetration and 0.253 for human resource attraction), implying that the research model explains a small proportion of the total variance. This may be because the authors used the actual data on the projects rather than people's perception of the concepts. Other researchers who have collected data from Sourceforge have encountered this problem.

Given the limited time and resources of the researchers, the authors were not able to collect data at more than 3 time points. One potential extension to this study would be collecting data for more time points over a longer period of time in order to see further longitudinal effects of the factors under study.

The longitudinal part of this study was undertaken in an exploratory fashion. One avenue of study for future researchers would be to conduct a confirmatory study on the above relationships using an appropriate theoretical lens.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the editor for their insightful comments and suggestions.

AUTHOR BIOS

Amir Hossein Ghapanchi is a tenured academic in the School of Information and Communication Technology, Griffith

University in Australia. Previously, Dr. Ghapanchi lectured at University of Technology Sydney and University of New South Wales (UNSW). Amir obtained his PhD in Information Systems from UNSW. He has published over 40 journal articles in information systems and management journals such as *Information and Organization, Communications of the Association for Information Systems, Journal of the Association for Information Science and Technology, Electronic Markets, Journal of Organizational Computing and Electronic Commerce, Journal of Medical Internet Research, Journal of Computer and System Sciences*, and *International Journal of Project Management,* among others.

Madjid Tavana is Professor and Distinguished Chair of Business Analytics at La Salle University, where he serves as Chairman of the Business Systems and Analytic Department. He is Distinguished Research Fellow at Kennedy Space Center, Johnson Space Center, Naval Research Laboratory at Stennis Space Center, and Air Force Research Laboratory. He was recently honored with the prestigious Space Act Award by NASA. He holds a MBA, PMIS, and PhD in Management Information Systems and received his Post-Doctoral Diploma in Strategic Information Systems from the Wharton School at the University of Pennsylvania. He is the Editor-in-Chief of *Decision Analytics, International Journal of Applied Decision Sciences, International Journal of Management and Decision Making, International Journal of Knowledge Engineering and Data Mining, International Journal of Strategic Decision Sciences,* and *International Journal of Enterprise Information Systems.* He has published 10 books and over 180 research papers in scholarly academic journals.

REFERENCES

Aksulu, A., & Wade, M. (2010). A comprehensive review and synthesis of open source research. *Journal of the Association for Information Systems*, *11*, 576–656.

Annacker, D., & Hildebrandt, L. (2004). Unobservable effects in structural models of business performance. *Journal of Business Research*, *57*(5), 507–517. doi:10.1016/S0148-2963(02)00317-X

Balli, S., & Korukoğlu, S. (2009). Operating system selection using fuzzy AHP and TOPSIS methods. *Mathematical and Computational Applications*, *14*(2), 119–130.

Carillo, K., & Okoli, C. (2008). The open source movement: A revolution in software development. *Journal of Computer Information Systems*, *49*(2), 1–9.

Chengalur-Smith, I., Daniel, S. & Sidorova, A. (2010). Sustainability of Free/Libre Open Source Projects: A Longitudinal Study. *Journal of the Association for Information Systems*, *11*(11), 657–683.

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), 997–1011. doi:10.1002/asi.v60:5

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), 123–148. doi:10.1002/(ISSN)1099-1670

Crowston, K., & Scozzi, B. (2002). Open source software projects as virtual organisations: Competency rallying for software development. *IEE Proceedings - Software*, *149*(1), 3–17. doi:10.1049/ip-sen:20020197

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management Science*, *35*(8), 982–1003. doi:10.1287/mnsc.35.8.982

Debra, Z., & Abbie, G. (2004). Customer learning processes, strategy selection, and performance in business-to-business service firms. *Decision Sciences*, *35*(2), 169–203. doi:10.1111/j.00117315.2004.02338.x

DeLone, W. H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: A ten-year update. *Journal of Management Information Systems*, *19*(4), 9–30.

Dwayne, C., & Dorothy, L. (2006). Bringing IT back: An analysis of the decision to backsource or switch vendors. *Decision Sciences*, *37*(4), 605–621. doi:10.1111/j.1540-5414.2006.00140.x

Feller, J., & Fitzgerald, B. (2002). *Understanding open source software development*. London: Addison-Wesley.

Fuggetta, A. (2003). Open source software—an evaluation. *Journal of Systems and Software*, *66*(1), 77–90. doi:10.1016/S0164-1212(02)00065-1

Ghanbarzadeh, R., Ghapanchi, A. H., Blumenstein, M., and Talaei-Khoei, A. (2014). A Decade of Research on the Use of Three-Dimensional Virtual Worlds in Health Care: A Systematic Literature Review. *Journal of Medical Internet Research*, *16*(2), 1–20.

Ghapanchi, A., Aurum, A., & Daneshgar, F. (2012). The impact of process effectiveness on user interest in contributing to the project. *Journal of Software*, *7*(1), 212–219.

Ghapanchi, A. H. (2013). Rallying competencies in virtual communities: A study of core processes and user interest in open source software projects. *Information and Organization*, *23*(2), 129–148.

Ghapanchi, A. H. (2015). Predicting Software Future Sustainability: A Longitudinal Perspective, Information Systems, Vol. 49, pp- 40–51.

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. H., & Aurum, A. (2011b). The Impact of Project Licence and Operating System on the Effectiveness of the Defect-Fixing Process in Open Source Software Projects. *International Journal of Business Information Systems*. *8*(4), 413–424.

Ghapanchi, A. H., & Aurum, A. (2012a). Determinants of project performance: Case of open source software projects. *International Journal of Project Management*, *30*(4), 407–417.

Ghapanchi, A. H., & Aurum, A. (2012b). Competency rallying in electronic markets: Implications for open source project success. *Electronic Markets*, *22*(2), 117–127. doi:10.1007/s12525-012-0088-0

Ghapanchi, A. H., Aurum, A., & Low, G. (2011). Creating a measurement taxonomy for the success of open source software projects. *First Monday*, *16*, 8.

Ghapanchi, A. H., Wohlin, C., & Aurum, A. (2014). Resources contributing to gaining competitive advantage for open source software projects: An application of resource-based theory. *International Journal of Project Management*, *32*(1), 139–152. doi:10.1016/j.ijproman.2013.03.002

Gold, A. (2007). Open source solutions: Seek value beyond cost. *Enterprise Open Source Magazine*. September 21, SYS-CON Media Inc.

Grewal, R., Lilien, G. L., & Mallapragada, G. (2006). Location, location, location: How network embeddedness affects project success in open source systems. *Management Science*, *52*, 1043–1056. doi:10.1287/mnsc.1060.0550

Haefliger, S., von Krogh, G. & Spaeth, S. (2008). Code Reuse in Open Source Software. *Management Science*, *54*(1), 180–193.

Haenlein, M., & Kaplan, A. M. (2004). A beginner's guide to partial leastsquares analysis. *Understanding Statistics*, *3*(4), 283–297. doi:10.1207/s15328031us0304_4

Hahn, J., Moon, J. Y. & Zhang, C. (2008). Emergence of New Project Teams from Open Source Software Developer Networks: Impact of Prior Collaboration Ties. *Information Systems Research*, *19*(3), 369–391.

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.

IEEE Standard Computer Dictionary. (1990). A Compilation of IEEE Standard Computer Glossaries, New York, NY.

Koch, S., & Schneider, G. (2002). Effort, co-operation and co-ordination in an open source software project: GNOME. *Information Systems Journal*, *12*(1), 27–42. doi:10.1046/j.1365-2575.2002.00110.x

Lettl, C. (2007). User Involvement Competence for Radical Innovation. *Journal of Engineering and Technology Management*, *24*(1-2), 53–75.

Liu, X. (2008). *Design architecture, developer networks and performance of open source software projects*. Boston, MA: Boston University.

Long, J. (2006). Understanding the role of core developers in open source software development. *Journal of Information, Information Technology, and Organizations*, *1*, 75–85.

Lundell, B., Lings, B., & Syberfeldt, A. (2011). Practitioner perceptions of open source software in the embedded systems area. *Journal of Systems and Software*, *84*(9), 1540–1549. doi:10.1016/j.jss.2011.03.020

Merati, E., Zarei, B., & Ghapanchi, A. (2010). Project Process Re-engineering (PPR): A BPR Method for Projects. *International Journal of Information Systems and Change Management*, *4*(4), 299– 313.

Midha, V. (2007). *Antecedent to the success of open source software*. Greensboro, NC: The University of North Carolina at Greensboro.

Midha, V., & Palvia, P. (2012). Factors affecting the success of open source software. *Journal of Systems and Software*, *85*(4), 895–905. doi:10.1016/j.jss.2011.11.010

Midha, V., Singh, R., Palvia, P., & Kshetri, N. (2010). Improving open source software maintenance. *Journal of Computer Information Systems*, *50*(3), 81–90.

Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, *11*(3), 309–346. doi:10.1145/567793.567795

Neuman, W. L. (2006). *Social research methods* (6th ed.). Boston, MA: Pearson International Edition.

Osborn, D. (2007). Survey of Localisation in African Languages, and its Prospects: A Background Document. Retrieved from http://idl-bnc.idrc.ca/dspace/handle/123456789/31740.

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), 246–256. doi:10.1109/TSE.2004.1274044

Peng, Y., Li, F., & Mili, A. (2007). Modeling the evolution of operating systems: An empirical study. *Journal of Systems and Software*, *80*(1), 1–15. doi:10.1016/j.jss.2006.03.049

Purarjomandlangrudi, A., Ghapanchi, A. H., & Esmalifalak, A. (2014). A Data Mining Approach for Fault Diagnosis: An Application of Anomaly Detection Algorithm. *Measurement*, *55*, 343–352.

Qua, W. G., Yangb, Z., & Wanga, Z. (2011). Multi-level framework of open source software adoption. *Journal of Business Research*, *64*(9), 997–1003. doi:10.1016/j.jbusres.2010.11.023

Qureshi, I., & Fang, Y. (2011). Socialization in open source software projects: A growth mixture modeling approach. *Organizational Research Methods*, *14*(1), 208–238. doi:10.1177/1094428110375002

Rehman, R. U. (2006). *Factors that contribute to open source software project success* (Master thesis). Carleton University, Ottawa, Canada.

Sammet, J. E. & Hemmendinger, D. (2003). Programming Languages. Encyclopedia of Computer Science, pp. 1470–1475.

Schermelleh-Engel, K., Moosbrugger, H., & Müller, H. (2003). Evaluating the fit of structural equation models: Tests of significance and descriptive goodness-of-fit measures. *Methods of Psychological Research*, *8*(2), 23–74.

Sen, R. (2007). A strategic analysis of competition between open source and proprietary software. *Journal of Management Information Systems*, *24*(1), 233–257. doi:10.2753/MIS0742-1222240107

Shy, O. (1996). Technology Revolutions in the Presence of Network Externalities. *International Journal of Industrial Organization*, *14*(6), 785–800.

Silberschatz, A., Galvin, P. B. & Gagne, G. (2004). *Operating System Concepts* (8th ed.). Hoboken, NJ: John Wiley & Sons.

Spinellis, D., & Giannikas, V. (2012). Organizational adoption of open source software. *Journal of Systems and Software*, *85*(3), 666–682. doi:10.1016/j.jss.2011.09.037

Stewart, K. J., Ammeter, A.P. & Maruping, L.M. (2005, January). 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, Big Island, HI.

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), 126–144. doi:10.1287/isre.1060.0082

Stewart, K. J., & Gosain, S. (2006). The impact of ideology on effectiveness in open source software development teams. *MIS Quarterly*, *30*(2), 291–314.

Subramaniam, C., Sen, R., & Nelson, M. L. (2009). Determinants of open source software project success: A longitudinal study. *Decision Support Systems*, *46*(2), 576–585. doi:10.1016/j.dss.2008.10.005

Tirole, J., & Lerner, J. (2002). Some simple economics of open source. *Journal of Industrial Economics*, *50*(2), 197–234.

Uchida, S., Monden, A., Ohsugi, N., Kamiya, T., Matsumoto, K., & Kudo, H. (2005). Software analysis by code clones in open source software. *Journal of Computer Information Systems*, *45*(3), 1–11.

Wangba, J., Hub, Y. H., & Shankerc, M. (2012). Human agency, social networks, and FOSS project success. *Journal of Business Research*, *65*(7), 997–984.

Zarei, B., & Ghapanchi, A. H. (2008). Guidelines for Government-to-Government Initiative Architecture in Developing Countries. *International Journal of Information Management*, *28*(4), 277–284.