A taxonomy for measuring the success of open source software projects

Author

Ghapanchi, Amir Hossein, Aurum, Aybuke, Low, Graham

Published

Journal Title

Version

Version of Record (VoR)

DOI

Copyright Statement

Downloaded from

Griffith Research Online

fi ® s t m ¤ ñ d @ ¥
PEER-REVIEWED JOURNAL ON THE INTERNET

A taxonomy for measuring the success of open source software projects
by Amir Hossein Ghapanchi, Aybuke Aurum, and Graham Low

Abstract

Open source software (OSS) has been widely adopted by organizations as well as individual users and has changed the way software is developed, deployed and perceived. Research into OSS success is critical since it provides project leaders with insights into how to manage an OSS project in order to succeed. However, there is no universally agreed definition of "success" and researchers employ different dimensions (*e.g.*, project activity and project performance) to refer to OSS success. By conducting a rigorous literature survey, this paper seeks to take a holistic view to explore various areas of OSS success that have been studied in prior research. Finally it provides a measurement taxonomy including six success areas for OSS projects. Implications for theory and practice are presented.

Contents

1. Introduction

Open source software (OSS) has been widely adopted by organizations as well as individuals and has changed the way software is developed, deployed and perceived. According to the 2020 FLOSS Roadmap [1], around 50 percent of Global 2000 IT organizations will adopt at least one OSS application by 2011. The widespread adoption of OSS (*e.g.*, Linux, Apache, Mozilla) has created considerable interest among researchers as well as practitioners.

Adoption of OSS has resulted in US$60 billion per year savings to its consumers. Johnson (2008) states "… while it [OSS] is only six percent of estimated trillion dollars IT budgeted annually, it represents a real loss of US$60 billion in annual revenues to software companies". Although OSS constitutes less than one percent of global software spend, it contributes to reduce overall expenditure by 25 percent (Tiemann, 2009). However, despite its significant role and the increasing adoption of OSS, around 63 percent of projects on Sourceforge.net, the largest OSS repository in the world, don't succeed (Krishnamurthy, 2002). Thus, a critical area of academic interest in OSS has been investigating the reasons for OSS success.

Unlike traditional closed–source software development (CSSD), OSS projects mostly rely on volunteers spending time and energy on the project and coordinating the development without the governance of a common entity (Stewart, *et al.*, 2006). The product is usually delivered for free (Feller and Fitzgerald, 2002). Given these points of difference, the success measures employed in CSSD — *e.g.*, staying within the time, and budget, and meeting the specifications — might not necessarily fit into open source software development (OSSD) environment (Crowston and Scozzi, 2002; Stewart, *et al.*, 2006).

"What constitutes the OSS success and how it can be measured" is an interesting topic that prior researchers have attempted to address (Lee, *et al.*, 2009; Midha, 2007; Stewart, *et al.*, 2006; Subramaniam, *et al.*, 2009; Crowston, *et al.*, 2003; Crowston, *et al.*, 2006; Crowston,

et al., 2004). However, as of yet the field of OSS has not settled on what constitutes OSS success (Crowston, *et al.*, 2004) and researchers measure OSS success in different ways (*e.g.*, project activity and user interest, etc.). This paper seeks to create a taxonomy of the various ways OSS success has been studied in prior research to help OSS researchers as well as OSS project administrators enrich their investigations of "success". Hence, the research question underlying the current paper is: "What are different approaches to measure success in open source software development?"

This paper provides an overview of current state–of–the–art research in OSS success. This approach helps us to determine where the literature has recurring themes and where are gaps in the existing body of knowledge? By updating work in this area, this research seeks to help researchers to determine current state–of–the–art in OSS success. The paper also has important implications for OSS practitioners.

This paper is organized as follows. Section 2 provides background information. Section 3 presents research methodology. Section 4 introduces our measurement taxonomy for OSS success. Section 5 provides implications for theory as well as for practice followed by concluding remarks.

■ ─────────────────────

2. Research background

A new software development and distribution practice has emerged over last two decades which, in February 1998, was termed open source software (OSS) development by a group of "free software" supporters, including Eric S. Raymond and Tim O'Reilly (Midha, 2007). This new development model is actually a re–emergence of the original development model involving sharing program code in which the introduction of copyright laws has changed the sharing environment of the original software programmers (Stallman, 2002). The application of copyright laws enabled the regulation of source code in terms of availability of the code, redistribution of the code and redistribution of modified software. Its supporters claim that OSS development can produce software with higher quality (Paulson, *et al.*, 2004). On the other hand, OSS development's opponents claim that the openness is not all good because it provides potential hackers with "the opportunity to study the software closely to determine its vulnerabilities" (Brown, 2002).

Unlike proprietary software in which the program's source code is a trade secret, in OSS the source code is publicly available for anyone who would like to see it. OSS products are developed under an open source license that permits their users to observe, modify and redistribute the program's source code (Midha, 2007). Despite having some similarities, OSS and closed source software (CSS) are different in many ways. For instance, CSS projects employ developers and pay them to develop software and try to sell it, whereas most OSS projects seek to attract volunteer programmers to develop software under the terms of a license that eventually lets everybody have the software and even use its source code (depends on the license of the program). Open source licenses are introduced by entities supporting OSS development style most notably Open Source Initiative (OSI at http://www.opensource.org/) and Free Software Foundation (FSF at http://www.fsf.org/). OSS licenses normally differ in following characteristics (Midha, 2007): (i) source availability (whether it is at nominal or zero cost); (ii) redistribution of the source code; and, (iii) redistribution of modified software (without discrimination or on the same terms as the original one).

The large majority of OSS projects show little activity or even become inactive over time, meaning that they are abandoned by developers. In fact, the main reason for the higher failure rate of OSS project compared with proprietary software projects is their high dependence on volunteer developers and voluntary contributions from OSS community (Ghapanchi and Aurum, forthcoming). Krishnamurthy (2002) confirms this by stating that 63 percent of OSS projects on Sourceforge.net, the world's largest OSS host, experiences failure because of their inability to attract user interest and contributions from developer community.

In spite of high failure rate, there are many successful examples of open source projects that have achieved a huge success. Mozilla Firefox, Apache, Open office, and Linux operating system are examples of such projects. There are several factors that contribute to OSS projects success; for instance, background support and funding (*e.g.*, LinEX project which is an initiative of the regional government of Extremadura, Spain).

In simple terms, success means achieving something desired (Midha, 2007). Measuring success is difficult because it is subjective. Crowston, *et al.* (2006) believe that defining success measures for OSS projects is much harder than that for regular information system projects "because of the problems defining the intended user base and expected outcomes" [2]. That is why there are different perspectives in the literature on OSS success.

Traditional software development success models frequently focus on success indicators such as system quality, use, user satisfaction and organizational impacts (DeLone and McLean,

2003) which are more related to "use environment" of the software, while studies on OSS success tend to look more at the "development environment" (Crowston, *et al.*, 2006). One reason is that unlike traditional closed source software development (CSSD), OSS projects mostly rely on volunteers spending time and energy on the project. Another reason is that in CSSD, "development environment" is not publicly available but the "use environment" is less difficult to study, while in OSS the "development environment" is publicly visible, but the "use environment" is hard to study or even to identify (Crowston, *et al.*, 2003; Crowston, *et al.*, 2006). Therefore, OSS researchers have mostly focused on other measures of success which relate to the "development environment" of the software (*e.g.*, project activity).

As Figure 1 illustrates, DeLone and McLean (2003) identified six constructs for the success of information systems including: system quality, information quality, service quality, system, user satisfaction, and net benefit. They postulate that positive experience with "use" should lead to "user satisfaction". They also argued that a higher level of satisfaction could increase "intention to use" which in turn could lead to a higher level of "use".
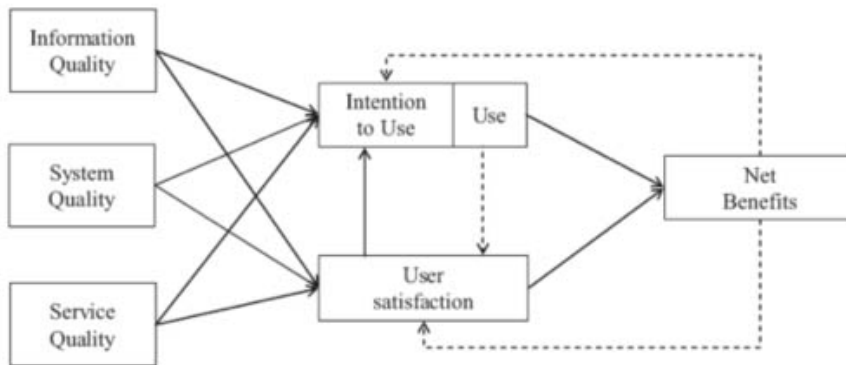

Figure 1: DeLone and McLean (2003) information system success model.

Several researchers have taken the topic of OSS success into consideration. For example, Lee, *et al.* (2009) customized DeLone and McLean (2003) model of success for OSS environment. Their model was like DeLone and McLean's (2003), excluding "information quality". They conducted a survey among the users of the Linux operating system. Their argument for deleting "information quality" was: "while information quality may be an important aspect of OSS–based application systems, our target OSS (*i.e.*, the Linux operating system) is not designed to produce any information [compared to a typical information system software]. For this reason, we drop the information quality construct from the DeLone and McLean IS [information system] success model in measuring the success of OSS". Stewart and Gosain (2006b) found that firstly the impact that team size has on perceived effectiveness is stronger early development stages of the project than in the later stages. Secondly, the influence of task completion on perceived effectiveness is stronger in later stages of development than in early stages. As another example, Subramaniam, *et al.* (2009) demonstrated that developer interest, user interest and project activity are correlated. Additionally, they concluded that project activity, developer interest, license and development status impact user interest in an OSS project.

Research on OSS success is of importance since it provides project managers with insights into how to manage an OSS project in order to succeed. Scholars have researched OSS success for nearly a decade. However, researchers still employ different project positive outcomes to refer to OSS success (Crowston, *et al.*, 2004). Crowston, *et al.* (2003) for example looked into product success to study OSS success. Stewart and Gosain (2006a), on the other hand, used project effectiveness to measure OSS success, while Subramaniam, *et al.* (2009) used user interest to measure OSS success.

3. Research method

This research used a literature survey to answer the research question underlying this study. This literature survey involved searching certain key words on certain academic databases (*e.g.*, IEEE Explore, ScienceDirect, Scopus, Business Source Premier, ProQuest database of Dissertations and Theses, etc.). Depending on the search services offered by search engine, the titles, abstracts, keywords, and in some cases full text of the publications in the included electronic databases were searched using the following search terms:

("Open source" OR "Open source software" OR "OSS"

OR "Open source project" OR "Open innovation" OR
"Open environment") **AND** ("Success" OR "Failure" OR
"Succeed" OR "Fail" OR "Performance" OR "Software
success" OR "Software quality" OR "Project success"
OR "Project outcome")

The search resulted in an initial set of 154 publications. Publications extracted (including journal papers, conference proceedings, dissertations, and working papers) were then reviewed and screened based on reading their titles and abstracts. As a result, we came up with 45 publications that have directly addressed our research question. shows the methodology employed by this paper to create OSS success taxonomy.
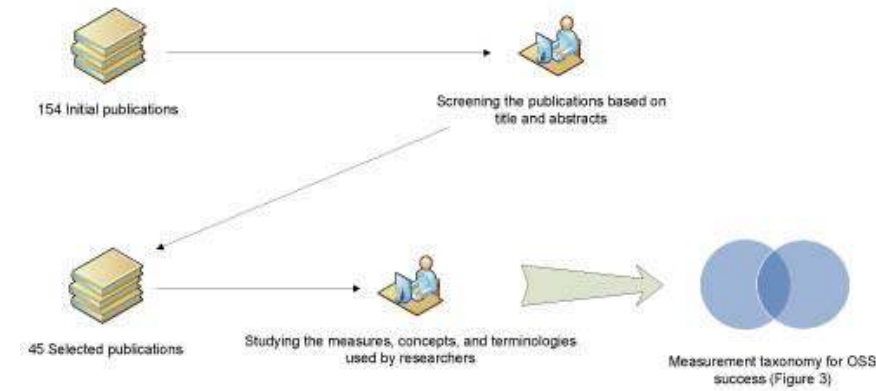

Figure 2: The process of creating the taxonomy.

Reviewing the 45 papers extracted, and in order to make more sense of the factors, we next sought to categorize them into meaningful clusters and create a measurement taxonomy for OSS success. We reviewed the extracted publications to identify the measures and terminology used by those publications to represent success. As a result, six broad success areas were identified namely: project activity, project efficiency, project effectiveness, project performance, user interest and product quality. We then allocated the 45 studies to the six success areas based on the terminology they used. The result is depicted in . Reviewing the extracted publications, we found that researchers have used different dimensions to investigate OSS success. Even some researchers who looked into the same dimension (*e.g.*, project effectiveness) employed different measures to gauge that particular dimension of OSS success. Thus, we examined 45 publications chosen for literature survey and synthesized them in order to come up with the measurement taxonomy for OSS success. In the following section the above–mentioned taxonomy is explained.

■ ────────────────────────

4. Measurement taxonomy for OSS success

As mentioned earlier, the methodology employed in this research resulted in 45 publications which investigated the success of OSS projects. The papers are listed in .

Table 1: References for each OSS success aspect in our source of literature survey.		
Main category	**Success area**	**Sample references**
Product success	Product quality	Conley, 2008; Colazo, 2007; Crowston, *et al.*, 2003; Liu and Iyer, 2007; Lee, *et al.*, 2009; Crowston, *et al.*, 2006; Fang and Colazo, 2010
Product/project success	User interest	Subramaniam, *et al.*, 2009; Hahn and Zhang, 2005; Stewart, *et al.*, 2006; Y. Long, 2006; J.

			Long, 2006; Midha, 2007; Crowston and Scozzi, 2002); Colazo, 2007; Liu and Iyer, 2007; Rehman, 2006; Long, 2004; Stewart and Ammeter, 2002); Grewal, *et al.*, 2006; Midha, *et al.*, 2010
Project success	Project performance		Liu and Iyer, 2007; Hahn and Zhang, 2005; Giuri, *et al.*, 2004; Y. Long, 2006
	Project effectiveness		Stewart and Gosain, 2006a; Stewart and Gosain, 2006b; Subramanian and Soh, 2006
	Project efficiency		Wray and Mathieu, 2008; Koch, 2009; Hahn and Zhang, 2005
	Project activity		Stewart, *et al.*, 2006; Hahn and Zhang, 2005; Giuri, *et al.*, 2004; Colazo, 2007; Y. Long, 2006; Grewal, *et al.*, 2006; Colazo and Fang, 2009; Chengalur–Smith, *et al.*, 2010

We mapped the literature on OSS success and came up with a taxonomy showing how prior researchers investigated OSS success and what aspects of success they considered. Figure 3 shows our taxonomy of studies on OSS success. The literature on OSS success can be divided into two broad categories: product success, and project success. According to Figure 3, the studies on project success fell into four success areas: project activity, project efficiency, project effectiveness, and project performance, while the publications on product success mainly focused on product quality. Interestingly, papers that studied user interest in OSS projects were split between the product and project perspective. In Figure 3 this success area is located in the intersection of project success and product success. In the next sections we examine each success area in Figure 3 along with examples from the literature.
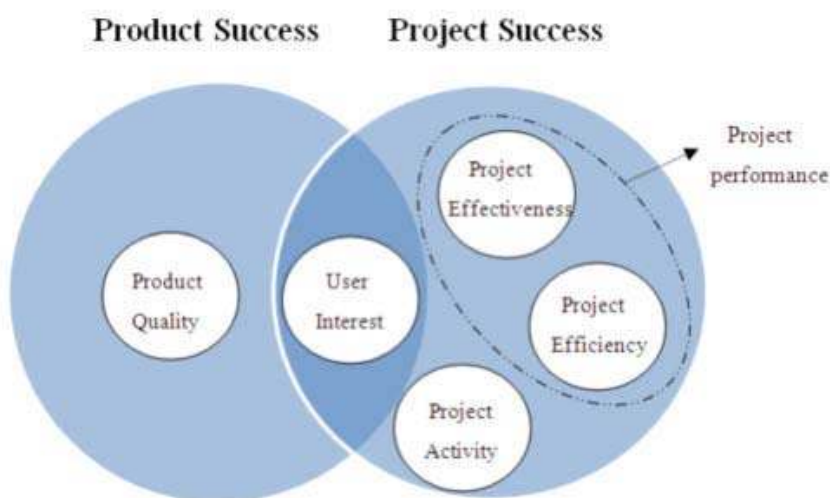

Figure 3: The measurement taxonomy for OSS success.

4.1. Product quality

Product success is one of the interesting research streams in OSS literature. Studies in this

success area focused on software quality as the output of the development process. However, different researchers have used different measures for product success. For example Crowston, *et al.* (2006) proposed code quality and documentation quality to measure software quality. The authors also paid attention to different aspects of quality (*e.g.*, completeness, maintainability, structuredness, efficiency, testability, usability, portability, consistency, conciseness, reliability, understandability). Crowston, *et al.* (2003) work is one of the most cited studies on OSS success. The authors conducted a content analysis on 201 messages produced in an online focus group through Slashdot.org. They identified seven main themes for OSS success namely: user, product, process, developers, use, recognition, and influence. Furthermore, they suggested some indicators for OSS product success including meeting the requirements, code quality, portability, and availability. Lee, *et al.* (2009), on the other hand, defined software quality using ease of use, user friendliness, and functionality. Borrowing from IS literature, they customized DeLone and McLean's (2003) model of success for OSS environment.

Beside product success, another important stream of OSS success research is OSS project success. Researchers have used different dimensions to refer to OSS project success, such as project performance and project efficiency. In the following, these success aspects are explained along with example studies from the literature.

4.2. Project performance

Project performance is one of the interesting research streams in the OSS literature. Studies in this area mainly focus on different measures to evaluate project outcomes "during development". Although various papers in this category used different indicators for project performance, they typically incorporate both efficiency and effectiveness. It is in line with the project management literature that posits that project performance is composed of project effectiveness and project efficiency (Crawford and Bryce, 2003). Efficiency simply refers to the extent to which output is created out of a particular amount of input (Efficiency=Output/Input). In other words, efficiency means doing things in the most economical way (Nichol,s 1999). Effectiveness, on the other hand, means the capability of producing an effect. In other words, effectiveness means getting the right things done (Nichols, 1999).

As mentioned previously, researchers have used various indicators to measure project performance. Liu and Iyer (2007), for example, used project velocity, product quality, and the project's market success (defined by the number of times a project's application has been downloaded). Hahn and Zhang (2005) looked into product use, development activity, and project efficiency. They found that project management practices like HR staffing, release management, communication and coordination, and compensation management impact on project performance. Y. Long (2006) on the other hand, measured project performance by project efficiency, activity, and popularity. Y. Long (2006) found that quality and quantity of knowledge sharing were found to affect project performance. Liu and Iyer (2007) found that product characteristics, team structure, the number of developers, developers' years of experience, and targeting developer impact on project performance.

The number of papers which take both OSS project effectiveness and efficiency into account is quite limited. Most of the studies either looked at project effectiveness (Stewart and Gosain, 2006a; Stewart and Gosain, 2006b; Subramanian and Soh, 2006) or project efficiency (Wray and Mathieu, 2008; Koch, 2009; Hahn and Zhang, 2005) which will be discussed in more detail in the next few sections.

4.3. Project effectiveness

OSS project effectiveness has caught several researchers' attention. Publications in this area have utilised various measures to evaluate project effectiveness and outcomes. Effectiveness means capability of producing an effect. In other words, effectiveness means getting the right things done (Nichols, 1999).

Publications in this category measured project effectiveness through different indicators. Subramanian and Soh (2006) used percentage of task completion. On the other hand, Stewart and Gosain (2006a) defined OSS project effectiveness comprising of two sets of variables: (1) input, including the number of developers the project has attracted, and the number of work weeks spent on the project; and, (2) output, the percentage of task completion (bug fix, patch, feature request, and support request).

One of the most cited papers on OSS project effectiveness was written by Stewart and Gosain (2006a). One of the results was that communication quality and team effort (the number of work weeks) affect task completion. As another example, Stewart and Gosain (2006b) looked at OSS project effectiveness perceived by OSS project administrators. Their findings showed that the influence of task completion on perceived effectiveness is stronger in later development phases compared with early stages.

4.4. Project efficiency

Project efficiency is another research stream in OSS studies. Studies in this success area look at the extent to which a project utilizes its resources to generate outcomes. Efficiency simply refers to the extent to which we create output out of particular amount of input

(Efficiency= Output/Input) (Nichols, 1999). In other words, efficiency means doing things in the most economical way. Researchers applied the concept of Data Envelopment Analysis (DEA) which is one of frequently used Multi Criteria Decision Making (MCDM) methods to calculate OSS project efficiency. DEA is a mathematical programming technique that measures the relative efficiency of multiple decision–making units (DMUs) based on multiple inputs and outputs (Eilat, *et al.*, 2006).

Although all papers found in this category used DEA to measure OSS project efficiency, they employed different input/output criteria to make their DEA model (see Table 2). Wray and Mathieu (2008), for instance, used the number of developers and the number of bug submitters as input criteria, and kilobytes per download, number of download and project rank as output criteria for their DEA model. Table 2 shows input and output criteria for the papers in this category.

As examples of studies in this success area, Wray and Mathieu (2008) ranked a set of 34 OSS projects based on a DEA model of two input and three output factors. Koch (2009) also found that adoption of Sourceforge.net tracker system and forum list as well as subversion and total tool adoption impact on project efficiency. Moreover, Hahn and Zhang (2005) identified that developer experience, user list, news list, tracker used and release speed significantly and positively affect efficiency of developer–targeted projects.

Table 2: DEA model of papers on OSS project efficiency.		
Author	**Input criteria for DEA**	**Output criteria for DEA**
Wray and Mathieu, 2008	Number of developers, number of bug submitters	Kilobytes per download, number of download, project rank
Koch, 2009	Number of downloads, number of years	Product size in bytes, number of code lines
Hahn and Zhang, 2005	Product size (bytes), development status	Number of developers, project age

4.5. Project activity

Project activity has been frequently regarded as one of the pillars of OSS project success. Papers in this success area take the quantity of project output generated either in a certain amount of time or in the project's whole lifespan into account. For instance, how frequently are defects fixed, new releases of the software posted, or support requests answered.

It is worth mentioning that researchers have used various types of indicators for project activity. For example Stewart, *et al.*, (2006), refer to the number of product releases as the indicator of activity. Hahn and Zhang (2005) used Sourceforge.net statistics on project activity. Interestingly, Giuri, *et al.*, (2004) used task completion (the number of bugs, patches, feature request completed as well as the number of software release) to measure project activity. Colazo (2007) employed the number of source code lines to measure project activity, while Grewal, *et al.* (2006) used the number of code commits on a project concurrent versioning system. OSS projects normally use a concurrent versioning system (CVS) manage their software development activities. Such tools enable the OSS project developers to store program source code at a central repository, thus ensuring that changes made by one developer are not accidentally deleted when another team member alters the source code. A commit to a project's CVS can include any number of lines of code (LOC) added or deleted (Colazo and Fang, 2009), therefore it reflects meaningful alternations to the source code (Grewal, *et al.*, 2006).

As examples of the papers in this category, Stewart, *et al.* (2006) found that user interest has a positive impact on the amount of OSS project development activity. Colazo (2007) also concluded that the number of core developers negatively impacts project activity. Giuri, *et al.* (2004) found that projects with developers who have higher level of different skills are more successful, and also project activity is determined by the ability of projects to attract users beyond the set of core contributors.

4.6. User interest

'User interest' is one the most relevant aspects of OSS project success, especially from OSS project administrators' point of view. User interest is defined as the ability of an OSS project to attract community users to adopt the project software (Stewart, *et al.*, 2006; Subramaniam, *et al.*, 2009). In other words, 'user interest' shows the level of popularity a project achieves in the community (Y. Long, 2006). Some indicators of "user interest" in prior research include traffic on the project Web site (Crowston, *et al.*, 2003; Crowston, *et*

al., 2006), downloads of the code (Subramaniam, et al., 2009; Crowston, et al., 2003; Crowston, et al., 2006), the number of developers who have joined the project team (Subramaniam, et al., 2009), and the number of people who have registered in the project mailing list to receive announcements such as a new release regarding a project (Stewart, et al., 2006; Crowston, et al., 2003; Crowston, et al., 2006).

Researchers who have worked on "user interest" (Stewart, et al., 2006; Subramaniam, et al., 2009) attributed different names to it including: "use" (e.g., Long, 2004; Hahn and Zhang, 2005; J. Long, 2006; Crowston and Scozzi 2002; Rehman, 2006), "popularity" (e.g., Midha, 2007; Stewart and Ammeter, 2002; Y. Long, 2006), "usage" (e.g., Colazo, 2007), "signal of market success" (e.g., Liu and Iyer, 2007), or even "commercial success" (e.g., Grewal, et al., 2006).

User interest is placed in the intersection of project and product success due to different indicators that researchers have used to measure it. Those papers which used the number of developers who have joined the project as an indication of user interest should be located in project success since they show user interest in the project. However the rest (e.g., the papers that used the number of downloads to measure user interest) should be located in product success since they represent the user community's interest in the software which is an output of the project, not the project itself.

Prior research on "user interest" has resulted in interesting findings. One of the first attempts in this regard was the research by Crowston and Scozzi (2002). They showed that using more common programming languages, having more developers, and more highly ranked or rated project administrators influence project success defined by activity, development status, and use. Moreover, Stewart, et al. (2006) showed that license restrictiveness is negatively associated with user interest, while having a sponsor has a positive impact. Furthermore, user interest has a positive impact on the amount of OSS project development activity. Subramaniam, et al. (2009) showed that developer interest, user interest and project activity are correlated. Additionally, they concluded that project activity, developer interest, and type of license impact user interest. Stewart and Ammeter (2002) also investigated that vitality has a significant impact on popularity over time showing that the more active a project is in terms of posting new releases and making announcements, the more attention it receives from the community.

---

5. Discussion and conclusion

This paper sought to review and advance the literature on OSS success through a survey of the literature. Based on the results, we suggested a measurement taxonomy for OSS success comprising six OSS success areas that have been studied in prior research. This taxonomy is based on the measures used in the studies and includes product quality, project performance, user interest, project efficiency, project effectiveness, and project activity.

5.1. Implications for researchers

From the insights gained by this study, we would like to augment OSS research communities' awareness regarding several issues.

Firstly, as discussed earlier, OSS researchers have been more inclined to use success measures that relate to "development environment" (e.g., number of downloads, number of lines of code, and number of developers) for a number of reasons such as ease of data collection from open repositories. One interesting area of research would be incorporating success measures that are related to "use environment" (e.g., net benefit, user satisfaction, etc.). The recent work by Lee, et al. (2009) is an initial attempt for this array of research.

Secondly, as discussed earlier, researchers have employed various measures when studying OSS success. However, many researchers have used a single measure to gauge the elusive phenomenon of OSS success. The implication of this for future researchers is to use multiple measures when gauging OSS projects' outcome to facilitate a more comprehensive view of OSS success.

Thirdly, we found that the current literature lacks from studies that simultaneously take into consideration both effectiveness and efficiency. In line with the project management literature that posits that project performance is composed of project effectiveness and project efficiency (Crawford and Bryce, 2003), we suggest future researchers to take into account both effectiveness and efficiency in order to be able to have a more holistic view on OSS success.

Fourthly, project performance has been defined in several different, and sometimes contradictory, ways in the existing literature. We call for a more precise conceptualization of OSS project performance.

Fifthly, success areas studied in prior research are looked into independently of each another. We suggest that future research takes into account potential relationships between

the various OSS success areas.

Finally, the success taxonomy proposed in this paper ([Figure 3](#)) provides a good starting point for a researcher interested in following up on one or more of the identified dimensions of OSS success discussed in this study.

5.2. Lessons for practitioners

Apart from strong theoretical implications, this study provides OSS project managers with useful insights. Previous studies have provided indicators for studying the success of OSS including the number of times an OSS product has been downloaded, the number of developers registered on the project, and the number of code commits produced in the project (Subramaniam, *et al.*, 2009; Stewart, *et al.*, 2006; Crowston and Scozzi, 2002). Project administrators should be aware that simply having a high download rate, or having a high level of code commit might not guarantee that their project will be successful in other respects. This paper provides a success taxonomy for OSS project managers including various success measures. A more holistic evaluation of an OSS project might then involve simultaneous measurement of user and developer interest, project activity, project effectiveness and efficiency, and product quality. [Table 3](#) provides a list of success measures proposed and used by prior studies on OSS success. These measures can help OSS project managers to gauge the success of their projects better.

Table 2: A practical list of OSS success measures.	
Success aspect	**Useful measures**
User interest	Traffic on the project Web site, downloads of the code, number of developers who have joined the project team, and the number of people who have registered in project mailing list to receive announcements such as new release regarding a project
Project activity	The number of software releases, number of patches, number of source code lines, number of code commits
Project effectiveness	The percentage of task completion (bug fix, feature request, and support request), number of developers the project has attracted, number of work weeks spent on the project
Project efficiency	Using a DEA model with one or some input indicators (*e.g.*, number of developers, number of bug submitters, number of years, product size (bytes), development status) and one or some output indicators (*e.g.* kilobytes per download, number of download, project rank, product size in bytes, number of code lines)
Product quality	Code quality, documentation quality, understandability, consistency, maintainability, program efficiency, testability, completeness, conciseness, usability, portability, functionality, reliability, structuredness, meeting the requirements, ease of use, user friendliness

The findings of this paper have several implications for corporations interested in adopting OSS projects. Selecting an OSS product from tens (and sometime hundreds) of available OSS products in the market has been reported as a very complicated task. The practical measures introduced in Table 3 not only can help OSS project managers to gauge the performance of their projects better, but they can also provide organizations who want to evaluate, compare, and finally adopt OSS projects with several criteria on which to base their analysis. 🔴

About the authors

Amir Hossein Ghapanchi is a Ph.D. candidate in the School of Information Systems, the University of New South Wales, Australia.

Aybuke Aurum is an Associate Professor in the School of Information Systems, the University of New South Wales, Australia.

Graham Low is a Professor in the School of Information Systems, the University of New South Wales, Australia.

Notes

1. International representatives from 30 countries involved in open source provided 2020 FLOSS roadmap in the 2008 open source forum, see http://www.2020flossroadmap.org/2010-version/, accessed 26 July 2011.

2. Crowston, *et al.*, 2006, p. 127.

References

K. Brown, 2002. "Opening the open source debate" (June), at: http://parrhesia.com/old_opensource_whitepaper.pdf, accessed 23 June 2009.

I. Chengalur–Smith, A. Sidorova, and S. Daniel, 2010. "Sustainability of free/libre open source projects: A longitudinal study," *Journal of the Association for Information Systems*, volume 11, number 11, at http://aisel.aisnet.org/jais/vol11/iss11/5/, accessed 26 July 2011.

J. Colazo, 2007. "Innovation success: An empirical study of software development projects in the context of the open source paradigm," Ph.D. dissetation, University of Western Ontario; see also http://gradworks.umi.com/NR/36/NR36686.html, accessed 26 July 2011.

J. Colazo and Y. Fang, 2009. "Impact of license choice on open source software development activity," *Journal of the American Society for Information Science and Technology*, volume 60, number 5, pp. 997–1,011.

C. Conley, 2008. "Design for quality: The case of open source software development," Ph.D. dissetation, New York University, Graduate School of Business Administration.

P. Crawford and P. Bryce, 2003. "Project monitoring and evaluation: A method for enhancing the efficiency and effectiveness of aid project implementation," *International Journal of Project Management*, volume 21, number 5, pp. 363–373.

K. Crowston and B. Scozzi, 2002. "Open source software projects as virtual organisations: Competency rallying for software development," *IEE Software Proceedings*, volume 149, number 1, pp. 3–17.

K. Crowston, H. Annabi, and J. Howison, 2003. "Defining open source software project success," *ICIS 2003: Proceedings of the 24th International Conference on Information Systems* (Seattle, Wash.).

K. Crowston, H. Annabi, J. Howison, and C. Masango, 2004. "Towards a portfolio of FLOSS project success measures," *Proceedings of the Workshop on Open Source Software Engineering, 26th International Conference on Software Engineering* (Edinburgh).

K. Crowston, J. Howison, and H. Annabi, 2006. "Information systems success in free and open source software development: Theory and measures," *Software Process: Improvement and Practice*, volume 11, number 2, pp. 123–148.

W. DeLone and E. McLean, 2003. "The DeLone and McLean model of information systems success: A ten–year update," *Journal of Management Information Systems*, volume 19, number 4, pp. 9–30.

H. Eilat, B. Golany, and A. Shtub, 2006. "Constructing and evaluating balanced portfolios of R&D projects with interactions: A DEA based methodology," *European Journal of Operational Research*, volume 172, number 3, pp.1,018–1,039.

Y. Fang and J. Colazo, 2010. "Following the sun: Temporal dispersion and performance in open source software project teams," *Journal of the Association for Information Systems*, volume 11, number 11, at http://aisel.aisnet.org/jais/vol11/iss11/4/, accessed 26 July 2011.

J. Feller and B. Fitzgerald, 2002. *Understanding open source software development* Boston: Addison–Wesley.

A. Ghapanchi and A. Aurum, forthcoming. "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*.

M. Giuri, M. Ploner, F. Rullani, and S. Torrisi, 2004. "Skills and openness of OSS projects: Implications for performance," at http://idei.fr/doc/conf/sic/papers_2005/giuri.pdf, accessed 26 July 2011.

R. Grewal, G. Lilien, and G. Mallapragada, 2006. "Location, location, location: How network embeddedness affects project success in open source systems," *Management Science*, volume 52, number 7, pp. 1,043–1,056.

J. Hahn and C. Zhang, 2005. "An exploratory study of open source projects from a project management perspective," at http://www.krannert.purdue.edu/academics/mis/workshop/hz_110405.pdf, accessed 22 October 2010.

J. Johnson, 2008. "Free open source software is costing vendors $60 billion," at http://www.standishgroup.com/newsroom/open_source.php, accessed 16 April 2008.

S. Koch, 2009. "Exploring the effects of SourceForge.net coordination and communication tools on the efficiency of open source projects using data envelopment analysis," *Empirical Software Engineering*, volume 14, number 4, pp. 397–417.

S. Krishnamurthy, 2002. "Cave or community?: An empirical examination of 100 mature open source projects," *First Monday*, volume 7, number 6, at http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/960/881, accessed 26 July 2011.

S.–Y.T. Lee, H.–W. Kim, and S. Gupta, 2009. "Measuring open source software success," *Omega*, volume 37, number 2, pp. 426–438.

X. Liu and B. Iyer, 2007. "Design architecture, developer networks and performance of open source software projects," *Proceedings of the Twenty–Eighth International Conference in Information Systems* (Montreal), at http://www.softwareecosystems.com/design_architecture_people_network.pdf, accessed 26 July 2011.

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

J. Long, 2004. "Understanding the creation and adoption of information technology innovations: The case of open source software development and the diffusion of mobile commerce," Ph.D. dissertation, University of Texas at Austin.

Y. Long, 2006. "Social structure, knowledge sharing, and project performance in open source software development," University of Nebraska — Lincoln, at http://digitalcommons.unl.edu/dissertations/AAI3216339/, accessed 26 July 2011.

V. Midha, 2007. "Antecedent to the success of open source software," Ph.D. dissertation, University of North Carolina at Greensboro.

V. Midha, P. Palvia, R. Singh, and N. Kshetri, 2010. "Improving open source software maintenance," *Journal of Computer Information Systems*, volume 50, number 3, pp. 81–90.

P. Nichols, 1999. *An introduction to the logframe approach: Course workbook & materials*. Melbourne: IDSS.

Open Source Initiative, 2005. "Open Source Initiative," at http://www.opensource.org, accessed 26 July 2011.

J. Paulson, G. Succi, and A. Eberlein, 2004. "An empirical study of open–source and closed–source software products," *IEEE Transactions on Software Engineering*, volume 30, number 4, pp. 246–256.

R. Rehman, 2006. "Factors that contribute to open source software project success," Master's thesis, Carleton University.

R. Stallman, 2002. "Why 'free software' is better than 'open source'," In: *Free software, free society: Selected essays of Richard M. Stallman* Boston: Free Software Foundation, and at http://www.gnu.org/philosophy/free-software-for-freedom.html, accessed 26 July 2011.

K. Stewart and T. Ammeter, 2002. "An exploratory study of factors influencing the level of vitality and popularity of open source projects," *Proceedings of the Twenty–Third International Conference on Information Systems*, pp. 853–857.

K. Stewart and S. Gosain, 2006a. "The impact of ideology on effectiveness in open source software development teams," *MIS Quarterly*, volume 30, number 2, pp. 291–314.

K. Stewart and S. Gosain, 2006b. "The moderating role of development stage in free/open source software project performance," *Software Process: Improvement and Practice*, volume 11, number 2, pp. 177–191.

K. Stewart, A. Ammeter, and L. Maruping, 2006. "Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects," *Information Systems Research*, volume 17, number 2, pp. 126–144.

C. Subramaniam, R. Sen, and M. Nelson, 2009. "Determinants of open source software project success: A longitudinal study," *Decision Support Systems*, volume 46, number 2, pp. 576–585.

A. Subramanian and P. Soh, 2006. "Knowledge integration and effectiveness of open source software development projects," *Proceedings of the Tenth Pacific Asia Conference on Information Systems* pp. 914–925, and at http://aisel.aisnet.org/pacis2006/111/, accessed 26 July 2011.

M. Tiemann, 2009. "How open source software can save the ICT industry one trillion dollars per year," at http://www.opensource.org/files/OSS-2010.pdf, accessed 3 March 2010.

B. Wray and R. Mathieu, 2008. "Evaluating the performance of open source software projects using data envelopment analysis," *Information Management & Computer Security*, volume 16, number 5, pp.449–462.

Editorial history