

## End User in Charge - Social Framework for Open Source Development

Kwabena Ebo Bennin, Shahid Hussain, Arif Ali  
Khan, Solomon Mensah, Ernest Pobee  
Department of Computer Science  
City University of Hong Kong  
Hong Kong, China  
{kebennin2-c,shussain7-c,aliakhan2-c,smensah2-c,  
epobee2-c}@my.cityu.edu.hk

Mohammed Alqadhi  
Department of Computer Science  
Jazan University  
Jazan, Saudi Arabia  
malqadhi@jazanu.edu.sa

**Abstract**— Open Source Software (OSS) is often developed in a public collaborative manner. Online OSS repositories such as GitHub, Google Code and SourceForge support collaborative OSS development by offering services such as subversion management, bug tracking and others. However, OSS mostly favors end-users who are programmers or have some prerequisite programming skills. The normally short README description file provided by the OSS developers does not contain enough information to help the novice end-users who intend to use the software in terms of installation and usage. Also, despite being equipped with social coding feature to support distributed multi-developer work environment, most OSS repositories provide only a storage space for the OSS files and this limits end-users just to their bugs/review comments on a different platform and naturally, people would also like to be key stakeholders like changing the functionality and accessibility of software they could use. Some online OSS repositories do not make provision for users to frequently communicate with the developers of the OSS to discuss about the published content on the repository. In this paper, we propose a social framework for OSS development to address the aforementioned issues. The framework is aimed to allow: (1) knowing the degree of matching between the sought user's requirements and the available OSS by presenting the end-user with the business domain model of a candidate OSS associated to its textual requirements description and (2) a lifetime communication between the users and OSS developers and even inviting other developers out of the OSS development team if needed.

**Keywords**-Open Source Software; OSS; End User; Crowdsourcing; Social development.

### I. INTRODUCTION

Open Source Software (OSS) development is an approach to the design, development, and distribution of software offering accessibility to software's source code for modification or enhancement. OSS has contributed to software technology by providing end-users from many sectors such as governmental/non-governmental organizations, businesses and individuals around the world

leverage to customize OSS for their personal needs (see Table I). Basically, this wide adoption of OSS is because an OSS is a freeware and promotes reuse through code transparency and a quality alternative to close source software [8][9]. Such quality is a result of collaborative efforts of developers from all over the globe and also the flexibility of allowing team members to contribute as much as they can, whenever they want [7]. Therefore, this partially defeats the classic concept that only a centralized management and strong control on the access to the source code produces a good and high quality software product. OSS has also dispelled in practice the view that rigorous management and a clearly defined design is instrumental for a successful software development project because many open source software projects have been successfully completed even without a clear initial design and formal management process [1].

Most OSS developers contribute to development of projects not because of money but as a way of giving to the society freely [6]. There exist many Web-based software repositories for hosting OSS such as GitHub [10] and SourceForge [11] on the world wide Web. They provide the service of social software development by facilitating multi-developer OSS projects and offering subversion control, bug tracking, release management, mailing lists and wikis.

TABLE I. EXAMPLES OF POPULAR OPEN SOURCE SOFTWARE

Usage Domain	OSS Example
Office Productivity Suites	Apache openOffice, libreOffice, Neo Office, Calligra
Finance and Accounting Applications	GnuCash, TurboCASH
ERP Software	ADempiere, OFBiz, OpenERP
CRM Software	SugarCRM, OpenCRX, Fat Free CRM
Communication and Telephony Software	AsteriskNOW, Elastix
Content Management Systems	Drupal, Wordpress, OpenCms, Joomla
E-commerce Tools	OpenCart, PrestaShop

The most often usage scenario for nowadays' Web-based OSS repositories begins with an end-user who looks for an OSS project that satisfies a set of requirements. The presented results based on the end-user's search on the OSS repository Website will be of one of these two possibilities: (1) finding a list of preexisting OSS that possibly partially fulfills the end-user's requirements or (2) finding no candidate OSS that matches or satisfies the end-user's requirement, thus, the end-user would have to create a totally new system from scratch. Also, most available OSS repositories do not provide the user with useful documentation information to help decide the degree to which a stored OSS satisfies a targeted user's requirements.

Mostly, an OSS will be associated only with a README file containing only technical information such as configuration, installation and others. For example, a simple search for a point of sale software may bring results of more than 30 different OSS systems. The end-user will be confused as to which software he/she should select. The end-user could end up installing/testing nearly half of the resulted list of systems before obtaining the desired system. Another limitation of these OSS repositories is the lack of developer support. This difficulty stems from the lack of frequent communication channels between OSS users and developers. Usually, end-users have to contact developers through their personal homepages and e-mails. For that, Websites for network of questions and answers, such as StackOverflow.com are the popularly targeted venues by OSS end-users. StackOverflow helps end-users to discuss and solve their problems with developers who voluntarily offer help and support. Research works in [3][4] have studied a group of OSS authors (i.e., those who developed the OSS) and the committers (i.e., those who reuse the OSS) who have been identified as active on both GitHub and StackOverflow platforms. Both studies observed that there is a positive connection between participating in StackOverflow and the productivity on GitHub. Also, the study in [5] found that end-users do not follow up with the OSS project within the OSS repository but rather, they go to the community Websites such as StackOverflow [2] when faced with any problem concerning the OSS project. All of the aforementioned studies can be indicatives of the difficulty the end-users endure in trying to communicate with the authors and developers of an OSS project on the OSS repository platform.

To address these two issues, we propose a social framework for OSS projects that will: (1) provide a mechanism for storing software requirements on an easy-to-manipulate format in order to facilitate the process of matching between the functionalities provided by the OSS and the sought end-user's requirements. (2) incorporate social networking feature to frequently connect developers and users. (3) Moreover, end-users are involved in the process of social reviewing and crowd testing of the OSS being developed.

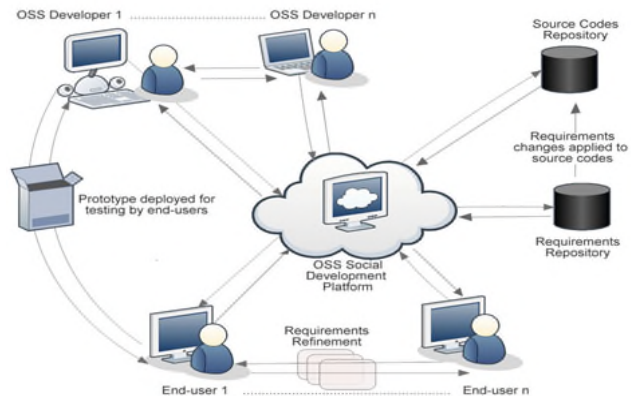


Figure 1. An overview of the proposed framework.

The framework (Figure 1) will be a Web-based service that offers a platform to undertake a multi-developer OSS project. It also provides a social networking facility to ensure all users (i.e., the developers and end-users) are connected.

## II. OVERALL FRAMEWORK

### A. Framework Components

The framework mainly consists of three components as follows:

1) *Source Code Repository*: this repository will contain the source code files for all OSS projects on the platform. Also, it will keep track of all changes made by different users by means of subversion management facility to ensure that each user's corresponding changes are linked to the changes in the requirements repository.

2) *Requirements Repository*: this will store the requirements documentation for all OSS projects available on the repository in XML-based format namely, XML-based Requirements Description Language (XRDL).

3) *Social Network*: this component will frequently connect all users in the repository (developers and users).

### B. XML-Based Requirements Description Language (XRDL)

We propose an XML-based language for providing a well-organized, structured and easy-to-manipulate format for storing requirement documents in the OSS requirements repository. This feature emanates from the inherent dynamicity of XML, as it is a dynamic markup language where one can define his/her own structures and constructs. Dynamic requirements description can be accomplished through user-defined requirements tags that define different requirements-related constructs. The tags may define the following constructs:

- Requirements
- Service type
- System Category
- Language
- User
- User Story
- Use-Case
- Class

```
<?xml version="1.0" encoding="UTF-8"?>
<requirements reqid="001"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="requirement.xsd">
  <system> <service>new</service>
</platform>PHP</platform></system>
<preference> <category>Information
systems</category></preference>
<story><user>tenant</user>
<body>user should get management fee receipt, report incidents, check
incident handling progress</body>
</story>
<story><user>Security Guard</user>
<body>update fee payment for tenants, update incident report status,
generate incident report</body></story>
</requirements>
```

Figure 2. A sample XRD document mapped from requirements collection form.

An XRD document (Figure 2) will be generated automatically by mapping the information retrieved from a form filled by the user at the beginning of the OSS development cycle. State-of-the-art techniques like textual analysis already employed within contemporary CASE tools will help in extracting artifacts such as actors, use-cases and classes from the user entries in the form. By utilizing XML (Figure 3) hierarchical power, these artifacts information will be transformed into different representations. We care about creating different representations of the requirements due to the fact that the platform will bring together volunteering developers from different schools of thought on software development. For instance, if a developer adopts Scrum development method then, he/she will be interested in viewing requirements as user stories. In contrast, a developer who follows a systematic software engineering approach needs to deal with use-cases and class diagrams which are doable by applying textual analysis to the user stories.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="requirement">
<xs:complexType>
<xs:sequence>
<xs:element name="system"><xs:complexType> <xs:sequence>
<xs:element name="service" type="xs:string"/>
<xs:element name="platform" type="xs:string"/>
</xs:sequence></xs:complexType></xs:element>
<xs:element name="preference"><xs:complexType>
<xs:sequence>
<xs:element name="category" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="story" maxOccurs="unbounded">
<xs:complexType><xs:sequence>
<xs:element name="user" type="xs:string"/>
<xs:element name="body" type="xs:string"/>
</xs:sequence></xs:complexType></xs:element>
</xs:sequence> <xs:attribute name="reqid" type="xs:string"
use="required"/>
</xs:complexType></xs:element>
</xs:schema>
```

Figure 3. An XML schema to capture requirements information in XRD document shown in Figure 4.

III. ILLUSTRATIVE OSS DEVELOPMENT STORY

The story begins with an end-user who has a set of business requirements and searches for an OSS that satisfies the requirements. The user will go to the platform and fill in a form that is designed in a very simple and easy-to-understand display to end-users (Figure 4). Upon submission of the form, all developers on the platform will be notified of the newly created requirements record.

Interested developers will pursue and provide feedback to the end-user. Other end-users who have similar requirements or needs can also join thus, they end up in a single development group within the social software development platform.

In the first project’s requirements collection form, the collected requirements are mostly composed of user stories. With this proposed framework, since there will be a number of end-users on the same platform, it will be easy to interact and get a clearer picture of the requirements through a social requirements refinement process. That is, the developers and other end-users who are subscribed to the social development platform can help in breaking down and refining the collected requirements (Figure 5).

Figure 4. A sample project’s requirements collection form.

**Requirements Comments**

- End users and reviewers comments**: Customers should be able to generate reports. 2 upvotes, 1 downvote. 2 days ago by Mohammed
- upvotes and downvotes**: Store owner should be able to know the most rented movie. 5 upvotes, 1 downvote. 1 day ago by Ernest
- Developer replies**: What kind of reports are you referring to? Could you clarify if the most rented movie feature is only available to store owners. Thank you. 1 day ago by Ebo

Figure 5. A sample social refinement for OSS requirements.

Should there be more than two end-users and developers on the platform, as end-users keep on refining the requirements, the developers develop and design prototypes. After each prototype, it can be pushed to the end-users for testing. This cycle will be repeated until a fully functional application is developed and all stakeholders are satisfied with the results. Another alternative scenario is when the end-user logs in to the OSS social development platform and searches the software repository for a possible OSS that matches his/her requirements. For the search, a set of indicative keywords related to the users of the systems and usage scenarios to search the repository could be used. The output of the search will be one of the following two scenarios:

- a) Returning a list of preexisting OSS that possibly partially fulfills the end-user's requirements.
- b) Finding no candidate OSS prompting the end-user to create a totally new system from scratch by filling in the requirements collection form. In case of availability of a match, then the platform will display the candidate list of OSS projects for the user associated with its most updated requirements. The presented requirements will be in the form of a business domain model associated with text-based requirements description (Figure 6). Through the presented business domain model, the end-user can quickly understand the functionality provided by the available OSS projects. The end-user will go through these listed OSS and check whether it matches his requirements either partially or fully.

If an OSS matches the user's requirement fully, then he/she will proceed to use it. However, it is more likely the OSS will match the end-user's requirements partially, so there will be a need to slightly change requirements and consequently, the corresponding OSS code accordingly. Therefore, the user will fill a change request form stating the minor changes to be implemented upon submission to the social development platform. Members who are experienced in the same programming language in which the selected OSS was written or domain or who worked on the previous OSS application will be notified with a new user request. This therefore will ensure there is still continuity of the project and communication with developers.

All changes or updates made to the source code will be saved in subversions. Similarly, the requirements documents will be updated with the latest refined requirements and stored in the repository in XRDL.

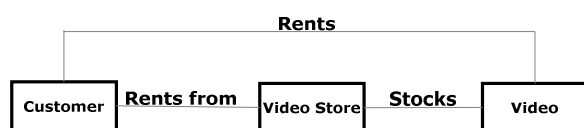


Figure 6. Part of the business domain model for a video rental system.

Finally, after completion of development of software, should the end-user encounter any challenges in the usage of the software, they could always return to the framework and interact with the developers to get their bugs fixed or request any update(s).

#### IV. BENEFITS

We list the potential benefits of using this framework.

**Social Requirements Refinement:** All active stakeholders (including end-users) who are experts in their domain together with the developers can collaborate and agree on the best requirement data and hence generate the best and necessary requirement for the developers to work with. There will be no misunderstanding with the output at the end of the project since requirements were clearly defined so many times during each information session.

**Social Review of Source Code:** Since the development is a crowd effort, other developers have the opportunity to review the source codes and make contributions in the areas of bug detection, refactoring and efficient and smart algorithms to make the system as robust as required.

**Crowd Testing:** A unique robust testing technique is applied in this framework. Testing is not only done by developers but end-users also participate in this process. Hence, bugs hidden during white box testing by programmers will be exposed by end-users. Also, with a large community, a rigorous stress testing approach is applied to the system to ensure the system is stable and can withstand heavy load.

**Reuse and sharing of components:** The source code repository will be available for reuse and therefore shared among the community.

#### V. CONCLUSION

The Web is now proliferated with so many open source projects or applications which are at the disposal of end users. The success of OSS on the Web is a clear indication of how end-users would like to be involved in the development of software. However, the platforms for OSS development mostly favor users who are programmers or have some pre-requisite programming skills and provide limited or no technical support. In this paper, we have addressed the challenges faced by end-users in finding and using OSS in the current Web-based OSS repositories. We proposed a social framework that involves end-users during the development of an OSS project.

End-users who need software will present their problems to an open community of developers and end-users and people who have the same or similar interest will collaborate to provide solution for the end-users. In this framework, contributors can willingly divide themselves into groups such as requirements analyst, developers and testers. The framework encourages crowdsourcing and crowd testing to support social end-user development by

which people can easily share problems and associated solutions together with the underlying rationale that the completed project can be used by all contributors.

#### REFERENCES

- [1] J. M. Gonzalez-Barahona and C. Daddara "Free software/open source: Information society opportunities for europe?" Working group on Libre Software, [http://eu.conecta.it/paper/cathedral\\_bazaar.html](http://eu.conecta.it/paper/cathedral_bazaar.html) (2000).
- [2] StackOverflow: Q&A Website. Available from: <http://www.stackoverflow.com>. [accessed December 2014]
- [3] B. Vasilescu et al., StackOverflow and GitHub: Associations between Software Development and Crowdsourced Knowledge, In Social Computing (SocialCom), 2013 International Conference on (pp. 188-195). IEEE.
- [4] B. Vasilescu, Human aspects, gamification, and social media in collaborative software engineering, in Companion Proceedings of the 36th International Conference on Software Engineering. 2014, ACM: Hyderabad, India. pp. 646-649.
- [5] C. Ayala et al., OSS Integration Issues and Community Support: An Integrator Perspective, in Open Source Systems: Long-Term Sustainability, I. Hammouda, et al., Editors. 2012, Springer Berlin Heidelberg. pp. 129-143.
- [6] A. Westenhof, (Ed). The Janus face of commercial open source software communities: An investigation into institutional (non) work by interacting institutional actors. Copenhagen Business School Press DK, 2012.
- [7] A. Hemetsberger, and C. Reinhardt. "Collective development in open-source communities: An activity theoretical perspective on successful online collaboration." *Organization studies* 30.9 (2009): pp. 987-1008.
- [8] A. Boulanger,. "Open-source versus proprietary software: Is one more reliable and secure than the other?." *IBM Systems Journal* 44.2 (2005): pp. 239-248.
- [9] B. Fitzgerald,. "A critical look at open source." *Computer* 37.7 (2004): pp.92-94.
- [10] C. Gandrud,. "Github: A tool for social data set development and verification in the cloud." Available at SSRN 2199367 (2013).
- [11] SourceForge Software repository. <https://www.sourceforge.net> [accessed December 2014]