2012 2nd International Conference on Information Communication and Management (ICICM 2012)
IPCSIT vol. 55 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V55.20

A Packaging Support System for Open Source Software

Amnart Pohthong [1+] and Supat Sriyanaluk [2]

[1] Information and Communication Technology Programme,
[2] Department of Computer Science, Faculty of Science, Prince of Songkla University, Hat Yai, Thailand.

Abstract. Open source software (OSS) has affected organizations and communities in many ways since it was introduced. OSS makes a significant impact on software development strategies and adoption in many organizations. OSS was used in software development in order to reduce cost, provide shorter time-to-market, and increase productivity. However, the documents needed in software engineering process for OSS development are often missing. Without these documents, software developers may have to take more time to modify available source code and it may increase the risk of misunderstandings. Therefore, the research reported here proposes a packaging support system for OSS in order to reduce these problems. The preliminary evaluation of system performance was conducted in our laboratory by the selected subjects acting as two groups of users: members and administrators. The results of system evaluation in term of users' satisfaction, using Likert scale, were rated as good quality (averaged 4.2) and very good quality (averaged 4.86) by members and administrators respectively.

Keywords: open source software, packaging support system, software packaging

1. Introduction

In the past, proprietary software played a key role in software development process. This factor led to the business and hacker culture. Open source software (OSS) was introduced in order to reduce these problems and also to avoid software piracy problems. The use of OSS has gained more popularity from software developers and user communities during the last decade for saving cost, providing shorter time-to-market, increasing productivity and also supporting software reuse [1]. Many OSS products have been widely adopted such as Linux, Apache, MySQL, FreeMind, and EasyPHP [2,3]. In addition to success factors mentioned in [4], OSS engineering process including documentation also becomes another key factor for adopting OSS.

Documentation seems to be a boring task for software developers. Especially, OSS developers pay less attention to documentation within the voluntary environment than within the commercial environment [5]. To reduce this problem, related documents to OSS should be packaged together with their source code. This packaging process should be included in OSS engineering process.

2. Research Backgrounds

2.1. Open Source Software

The open source definition and its distribution terms have been developed by the Open Source Initiative (OSI) [6] as follows:

- *free redistribution* without payment
- distributed and publicized access to *source code*
- modification and distribution of *derived works* must be allowed
- *integrity of the author's source code*

[+] Corresponding author. *E-mail address*: amnart.p@psu.ac.th

- *no discrimination against persons or groups*
- *no discrimination against fields of endeavor*
- *distribution of license*
- *license must not be specific to a product*
- *license must not restrict other software*
- *license must be technology-neutral*

Hao et al also suggested four important characteristics of OSS in [7]: open-based process, network-based form, flat-based organization, and share-based product. Gacek and Arief classify OSS users into two main groups [8]: passive users and active users (contributors). Active users are involved in software development activities while passive users are not involved in software development. Active users can be either non-developers or developers. OSS developers can be core developers or co-developers.

2.2. Software Packaging

Unlike in many industrial productions where product packaging is very important within the process, in software production process we are not concerned very much with software packaging. Most source codes, executable codes as well as related documents are actually recorded in the form of file structures on storage media. Software packaging has been defined as the collection of necessary program files and related documents in specific configuration for distribution and installation on desired environment.

3. Proposed System

3.1. Proposed Framework

The open-source packaging system was designed and operated as shown in Fig. 1 and the following steps.

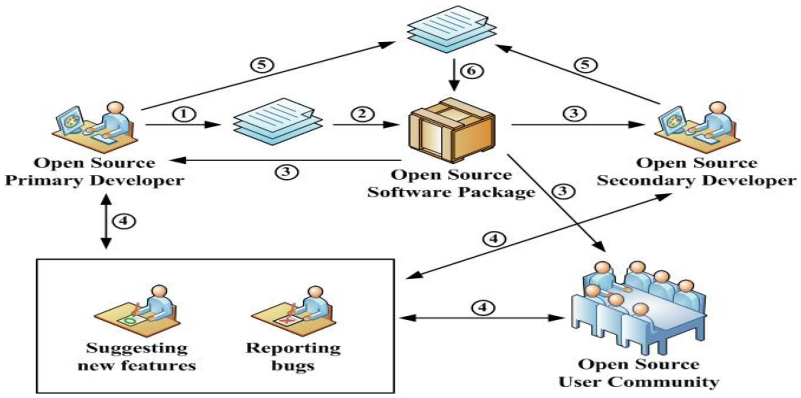

Fig. 1: The proposed system framework

(1) An OSS primary developer creates his or her source code together with its related documents.

(2) OSS packaging is performed as the collection of source code, related documents and program configuration for installation.

(3) An OSS package is distributed to OSS communities.

(4) Suggesting new features and reporting bugs are informed by OSS communities.

(5) OSS improvement engineering is performed by the primary developer or a secondary developer.

(6) OSS packaging is redone and a new OSS package is released.

3.2. System Design

From the proposed framework, the system architecture was designed in style of a repository model as shown in Fig. 2. It consists of seven main sub-systems as follows:

- Elementary Service: for applying membership, authentication (sign in and sign out), about the system, and contact us.
- OSS Packaging Management: for packaging source code and its related documents, searching, displaying, modifying and downloading software package.

- OSS Opinion Management: creating, displaying and deleting an opinion.
- OSS Category Management: creating, modifying, displaying category data for OSS.
- OSS News Management: creating, modifying, displaying and deleting news.
- OSS Fault Management: reporting, displaying and deleting fault information.
- Member Management: for changing password, searching, displaying and modifying a member's data.

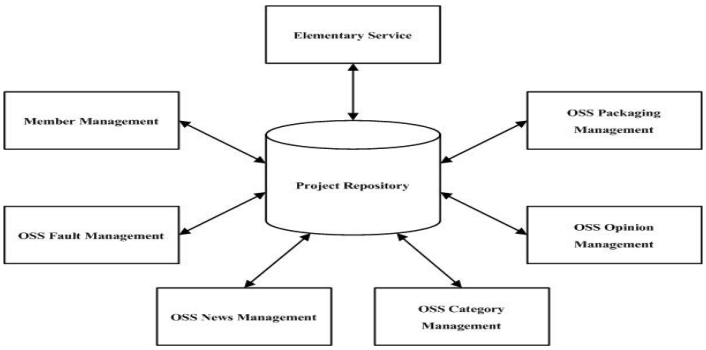

Fig. 2: The system architectural model

The system was divided in to several core modules corresponding to three groups of users: non-member users, members, and administrators, as shown in Fig. 3.

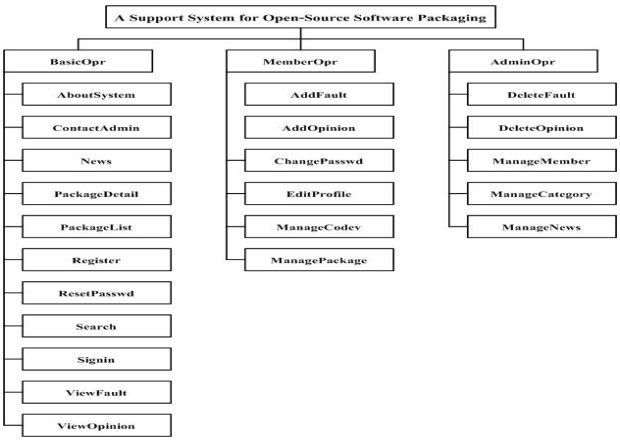

Fig. 3: The system modular structure

The system database was designed using an entity-relationship (E-R) diagram as shown in Fig. 4.

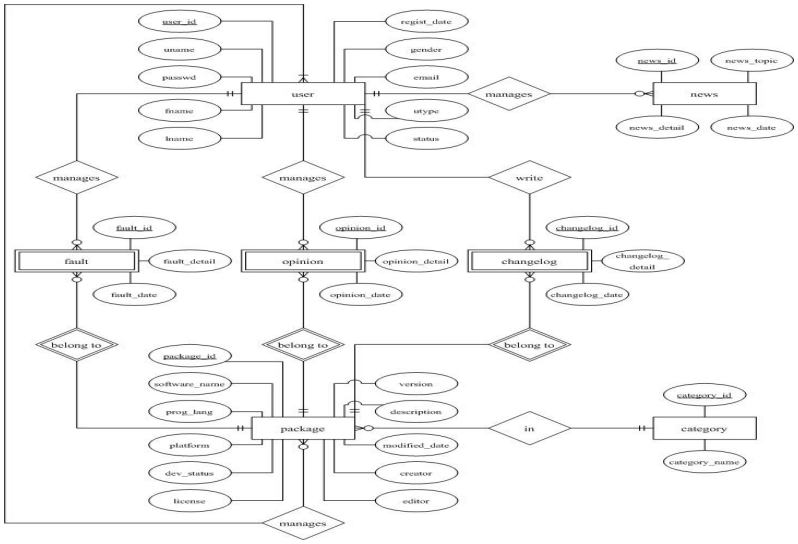

Fig. 4: The system database

3.3. System Implementation

The system was implemented in MVC(Model-View-Controller) style. Some examples of user interfaces such as the system main menu, the window for creating package, the main menu for system members, and the list of created open source software, as shown as Fig. 5–8.

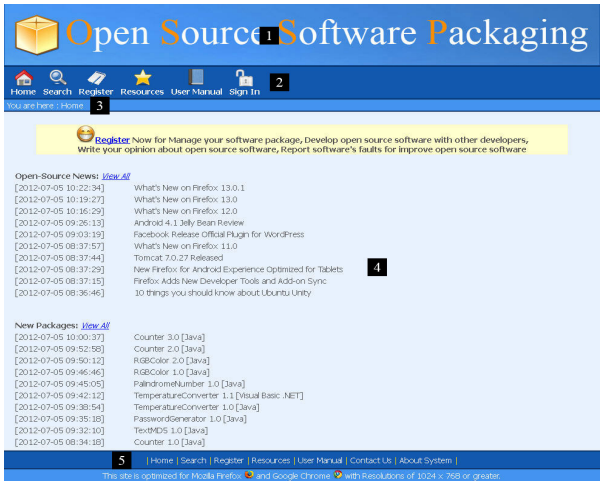

Fig. 5: The system main menu

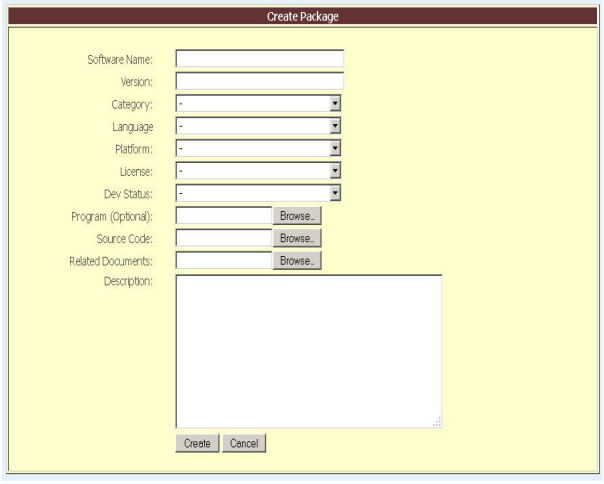

Fig. 6: The window for creating package

Fig. 7: The main menu for system members

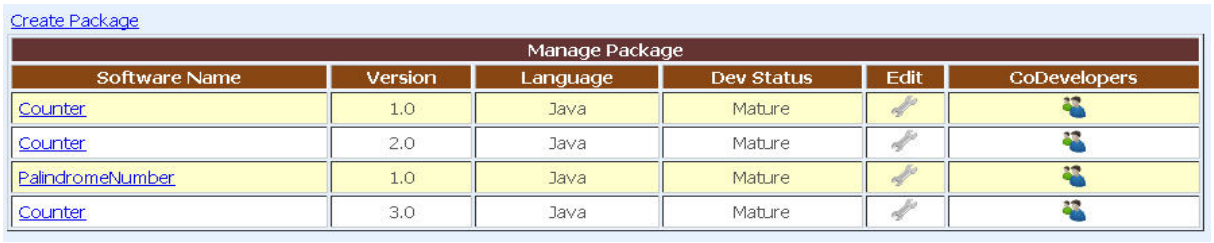

Fig. 8: The list of created open source software

3.4. System Evaluation

The system was evaluated in our laboratory for users' satisfaction in terms of four categories: system usage, efficiency, dependability, and value added for OSS. Seven subjects were voluntarily selected. They were drawn from programmers and graduate students working and studying at the Faculty of Science, Prince of Songkla University, Thailand. Five of the subjects hold their M.Sc.'s degrees and the rest hold their B.Sc.'s degrees, all related to Computer Science or Information Technology. Two of them were assigned to act as system administrators and the rest to act as system members.

The evaluation tools were the given problems and questionnaire. The given problems corresponding to the type of users and consisting of several questions, allow the subjects to explore and use the most related functions and features of the system. The questionnaire consists of three parts: personal data, questions for rating satisfaction level using Likert scale (from 1=very low to 5=very good), and opened suggestions. The evaluation procedures started with giving a brief introduction about the system and its evaluation method, taking around 5 to 10 minutes. Then, the subjects performed the given problems. Finally, all subjects completed the questionnaires.

The overall results in terms of users' satisfaction were rated as good quality (averaged 4.2 from 5) by members and as very good quality (averaged 4.86 from 5) by administrators.

4. Discussion and Conclusion

The packaging support system was proposed in order to enhance OSS engineering process in term of documentation improvement. The system allows users and developers to share their source code as well as related documents such as software specification, design, and test cases. These documents would help open source communities to quickly develop and modify software. It could also lead to increased productivity and reduce the risk of misunderstanding the source code. In addition to the packaging support function, the functions and features of the system are similar to simple open-source management systems. Therefore, the systems may be useful for organizations interested in developing their own open source communities. However, a prototype of the system is based on our experience and problems found when we used open source software. This may be the limitation of this study. In future, the system should be distributed to open source communities.

5. References

[1] S. A Ajila and D. Wu. Empirical study of the effects of open source adoption on software development economics. *The journal of System and Software*. 2007, 80: 1517-1529.

[2] O. Hauge, C. Ayala and R. Conradi. Adoption of open source software in software-intensive organizations- a systematic literature review. *Information and Software Technology*. 2010, 52: 1133-1154.

[3] R. Kemp. Current developments in Open Source Software. Computer Law&Security Review.2009, 25: 569-582.

[4] V. Midha and P. Palvia. Factors affecting the success of Open Source Software. *The journal of System and Software*. 2012, 85: 895-905.

[5] M. Host and A. Orucevic-Alagic. A systematic review of research on open source software in commercial software product development. Information and Software Technology. 2011, 53: 616-624.

[6] The Open Source Initiative. The Open Source Definition. http://www.opensource.org/docs/osd (accessed 19/08/2012)

[7] X. Hao, Z. Zhengang, L. Chunpei and D. Zhuo. The study on innovation mechanism of Open Source Software. Proc. the 4th International Conference on Wireless Communications, Networking and Mobile Computing, Dalian, China, 2008: 1-5.

[8] C. Gacek and B. Arief. The many meanings of Open Source. IEEE Software. 2004, 21(1): 34-40.

Amnart Pohthong was born in Trang, Thailand in 1959. He earned his B.Sc. degree in Mathematics in 1981 and M.Sc. in Computer Science in 1991. Both degrees are from Prince of Songkla University (PSU), Thailand. He received a Ph.D. in Computer Science in 2000 from Keele University, U.K. He has been working at PSU since 1983. Over the years he has held several positions and experienced in teaching, research, and administrative work. Currently, he is an assistant professor at the Faculty of Science, PSU. His research interests include software engineering, information systems, and knowledge management.

Supat Sriyanaluk was born in Nong Khai, Thailand in 1986. He earned his B.Sc. degree in Information Technology in 2007 and M.Sc. degree in Computer Science in 2012. Both degrees are from Prince of Songkla University, Thailand. His current research interest is software engineering.