SOFTWARE PROCESS IMPROVEMENT AND PRACTICE
Softw. Process Improve. Pract. 2007; **12**: 000–000
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/spip.361

Research Section

An Empirical Study on Software Development with Open Source Components in the Chinese Software Industry•

AQ1

Weibing Chen[1]* , Jingyue Li[2], Jianqiang Ma[1], Reidar Conradi[2], Junzhong Ji[1] and Chunnian Liu[1]
[1] *Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, Beijing University of Technology, Beijing 100022, China*
[2] *Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway*

Chinese software companies are increasingly using open source software (OSS) components in software development. Development with OSS components faces challenges with respect to component selection, component integration, licensing compliance, and system maintenance. Although these issues have been investigated in the industry in other countries, few similar studies have been performed in China. It is therefore difficult for Chinese software companies to be aware of their special issues and to make the necessary improvements. This article describes a questionnaire-based survey of software development with OSS components in Chinese software companies. Data from 47 completed development projects in 43 companies were collected. The results show that the main motivation behind using OSS components was their modifiability and low license cost. Using a web search engine was the most common method of locating OSS components. Local acquaintance and compliance requirements were the major decisive factors in choosing a suitable component. To avoid legal exposure, the common strategy was to use components without licensing constraints. The major cost of OSS-based projects was the cost to learn and understand OSS components. Almost 84% of the components needed bug fixing or other changes to the code. However, close participation with the OSS community was rare.

KEY WORDS: component-based software engineering; open source software component; empirical study

1. INTRODUCTION

Developing new software systems using pre-made components is an attractive way to achieve lower cost, shorter time-to-market, and better system quality (Li *et al*. 2005a). It is increasingly common to develop with open source software (OSS) components (Brown and Booch 2002, Fitzgerald and

* Correspondence to: Weibing Chen, Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, Beijing University of Technology, Beijing 100022, China
† E-mail: weibingchen@gmail.com

Kenny 2004, Norris 2004, Spinellis and Szyperski 2004). Reusing OSS components (and 'external' components in general) creates challenges for their appropriate selection and proper integration (Madanmohan and De 2004). If an OSS-based system is going to be distributed or sold to the general market, another challenge is how to comply with the terms of licensing for the OSS components that have been used (Brown and Booch 2002, Ruffin and Ebert 2004).

Several theoretical studies (especially around component selection) (Brown and Booch 2002, Giacomo 2005) and industrial case studies (Fitzgerald and Kenny 2004, Norris 2004) have been performed on OSS-based development. One industrial survey investigated the state of the practice of OSS-based development in three European countries (Li *et al*. 2005a). Although China has become a major actor in employing OSS in industry, especially regarding software platforms such as Linux, little research has been performed on the challenges of reusing OSS components efficiently in the Chinese software industry.

We performed an industrial survey to investigate why Chinese companies decide to use OSS components, how they select and integrate them, and how they maintain the OSS-based system. A structured questionnaire was used to collect data. Name lists from a membership-based organization for Chinese software companies were used to get a representative subset of companies. Data from 47 finished projects in 43 companies were collected.

The results show that Chinese software companies are similar to the European companies investigated (Li *et al*. 2005a) in the following respects: the motivations for using OSS component, the processes used to locate components, and the criteria adopted to evaluate them. The results also show the following. Chinese software companies did not consider possible support from OSS communities to be an important issue when selecting components. Few Chinese companies have actively participated in OSS communities, although they want to. The cost of learning OSS components is one of the major costs of OSS-based projects. Using these results as a basis, we offer four recommendations on how to facilitate OSS-based development in China and worldwide.

The rest of this article is organized as follows. Section 2 describes related work. Section 3 presents the research design. Section 4 describes the participants. Section 5 illustrates answers for each research question, followed by detailed discussions. Section 6 contains a general discussion. Section 7 concludes and presents possible avenues for further work.

2. RELATED WORK

There have been two main kinds of empirical studies of OSS:

- *Cultural-oriented studies* concentrate on how to develop new OSS and its components. The focuses are the OSS participators' motivation and the evolution of OSS projects (Lakhani and Wolf 2005).
- *Technical-oriented studies*, such as the one reported herein, concentrate on process issues in reusing and integrating existing OSS components to develop new software (Ruffin and Ebert 2004, Merilinna and Matinlassi 2006). Relevant work with respect to the technical-oriented studies is described below.

2.1. Motivation of Using OSS Component

The use of OSS components in system development is growing rapidly, because of its well-known advantages. One study shows that one major motivation for using OSS components is to produce software at 'zero cost or as cheap as possible' (Fitzgerald and Kenny 2004). Another study illustrates that the main motivation for using either commercial-off-the-shelf (COTS) or OSS components in European countries is to achieve shorter time-to-market, less development effort, and better system quality (Li *et al*. 2005a).

2.2. OSS Component Selection

Selecting an appropriate component is a central factor for the success of OSS-based development. In recent years, the number of OSS components has increased dramatically. More than 159,000 OSS projects had been registered at *sourceforge.net* by October 2007. Faced with so many alternatives, it is difficult to select the 'best' one to use (Ncube and Maiden 2004).

Typically, the process of selecting components includes identifying candidate components (searching), evaluating them on the basis of predefined

DOI: 10.1002/spip

criteria, and deciding upon one or several components (Madanmohan and De 2004, Ncube and Maiden 2004). Most previous studies on component selection focus on selecting COTS components (Ncube and Maiden 2004, Briand 1998). Due to differences between OSS and COTS components, the proposed process for selecting COTS components may not be suitable for selecting OSS components (Giacomo 2005). In practice, 'ad hoc' processes for selecting OSS components are performed, under considerable risk and uncertainty (Li *et al*. 2005a).

2.3. OSS Component Integration and OSS Licensing Issues

After the OSS components have been selected, the next step is to integrate them into the target system. To ensure the success of integration, the integrators need to consider both technical issues and the terms of licensing of the selected OSS components. There are more than 50 different OSS licenses (Open Source Initiative 2005). Some licenses have strict constraints on the distribution or resale of the system that is derived from OSS components. For example, the GNU Public License (GPL) type licenses do not give the licensee unlimited redistribution rights. (Madanmohan and De 2004, Ruffin and Ebert 2004). Although major legal aspects of using OSS components and related strategies for mitigating risks have been discussed (Ruffin and Ebert 2004), few follow-up studies have been performed to examine how the licensing issues are managed in practice.

2.4. Maintenance of the OSS-Based System

After the OSS components have been integrated into a software system, it is important to maintain and update those components properly for long-term use. Given that support from OSS communities is provided mainly by loosely organized volunteers, it is difficult to control the quality of support. To receive high-quality and long-term support, one proposed strategy is to establish a long-term working relationship with the corresponding OSS communities (Norris 2004). According to this proposal, the users of OSS components not only download software from the OSS community but also upload the modified software to it (Norris 2004, Merilinna and Matinlassi 2006). Such a relationship between users and the OSS community is supposed to benefit both OSS communities and users (Brown and Booch 2002). However, the software industry has not supported this proposed practice.

To maintain OSS-based systems more effectively, another proposal is to build an internal OSS reuse repository, which includes the source code, documentation, and previous users' feedback about OSS components (Morad and Kuflik 2005). However, some researchers regard having such an internal comprehensive repository of OSS components as unrealistic (Dagdeviren *et al*. 2005).

2.5. Cost Distribution of the OSS-Based Project

One important issue for the success of OSS-based development is the lifecycle cost (Abts *et al*. 2000). Although OSS components can be acquired free or at very low cost, there are indirect costs, such as costs related to component selection, learning, adaptation, and maintenance. For most IT projects, indirect costs can increase the total cost of a project dramatically (Wang and Wang 2001, Giacomo 2005). The imprecise estimation of the effort required for selecting and integrating OSS components is a problem that occurs frequently in OSS-based development (Li *et al*. 2005b). A cost-estimation model can help avoid the imprecise cost estimation. However, there are still no well-formulated cost models for OSS-based development (Madanmohan and De 2004).

3. RESEARCH DESIGN

China is a major user of OSS components in information systems. The Chinese government has played an important role in promoting the Chinese OSS movement. For example, the Japan-China-Korea (JCK) open alliance was announced in November 2003 to promote OSS by mutual cooperation (Kshetri 2005). Owing to the Chinese government's encouragement regarding the use of Linux and OSS, more and more Chinese software companies are starting to use OSS components when developing software. No other country comes even close to the level of advancement that China has achieved in deploying OSS, particularly Linux (Kshetri 2005). The current scale of OSS-based development is large enough to be noticed at the global level. However, there are few empirical

DOI: 10.1002/spip

studies on OSS-based development in the Chinese software industry. *The aim of this study is therefore to establish empirically based guidelines to ease OSS-based software development.*

The study focused on development based on OSS components. A software component is defined as in Li *et al*. (2004): Software components are executable units of independent production, acquisition, and deployment that can be used to build a functioning system. An OSS component is defined as a software component that is: (a) provided by the OSS community; (b) subject to licensing constraints; and (c) not a software platform (e.g. Linux, Mysql, or similar).

3.1. Research Questions

We formulated five research questions and corresponding subquestions with respect to the issues presented in Section 2. First, we want to know why Chinese developers decided to use OSS components. The research question RQ1 was formulated as following:

RQ1: Why do Chinese developers decide to use OSS components in software development?

We were also interested in how Chinese developers selected OSS components, how they dealt with OSS licensing terms when integrating OSS components, and how they maintained and updated the OSS-based system. Thus, our research questions RQ2 to RQ4 were formulated as following:

RQ2: How do Chinese developers select OSS components?

- RQ2.1. What methods are used to locate candidate OSS components?
- RQ2.2. What evaluation criteria are used to evaluate and compare OSS candidates?

RQ3: How do Chinese developers deal with OSS licensing terms?

- RQ3.1. How well do developers understand OSS licenses?
- RQ3.2. Do developers read related OSS licensing terms?
- RQ3.3. Do developers encounter problems caused by OSS licenses?
- RQ3.4. What strategies are used to avoid possible problems with OSS licensing?

RQ4: How do Chinese developers maintain and update OSS-based system?

- RQ4.1. Does engagement with the OSS community facilitate the maintenance of OSS-based systems?
- RQ4.2. Are any internal knowledge repositories being used to facilitate the maintenance of OSS-based systems?

In addition, we were interested in the cost distribution among possible activities of OSS-based projects. The research question RQ5 was therefore formulated as following:

RQ5: What is the cost distribution of OSS-based projects?

3.2. Sample Selection and Data Collection

To collect data that could be used to answer the research questions, we used a survey. First, we consulted the literature and used it as a basis for designing a preliminary questionnaire with both open-ended and closed questions. Second, we performed a prestudy to validate the quality of the questions in the preliminary questionnaire and to get answers to the open-ended questions. After assessing the results of the prestudy, we reformulated most of the open-ended questions in the preliminary questionnaires as closed questions. In addition, we revised any questions in the preliminary questionnaire that we found to be problematic. Then, we used the revised questionnaire to collect data in a main study. Details of the procedures for selecting samples and collecting data are described in Chen *et al*. (2007).

4. BACKGROUND INFORMATION OF THE •PARTICIPATING COMPANIES AND PROJECTS

AQ2

Participating companies. The companies include large, medium, and small ones (see Figure 1 in Chen *et al*. (2007) for details). The main applications or services offered by these companies are shown in Figure 1.

Projects. It was found that 53% of the projects used one or two different OSS components and 6% used more than five components. And 47% of the projects used Java, 32% used C/C++, and the remaining projects used other programming languages.

DOI: 10.1002/spip

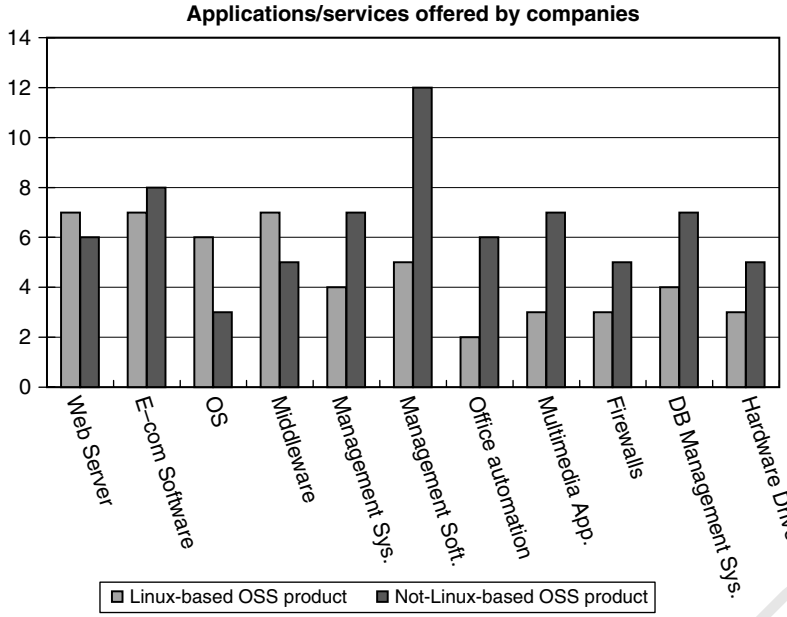

Figure 1. The main applications or services offered by the participating companies

Human respondents. Most respondents had a relevant education background. Twenty-two of the respondents had a bachelor's degree, 22 a master's degree, and the remaining three a Ph.D. degree. Thirty-five respondents studied computer science and 12 studied other subjects. The detailed information of the respondents' working experience is presented in Chen *et al*. (2007).

5. RESULTS AND DISCUSSION OF RESEARCH QUESTIONS

We now present the results for each research question, followed by a detailed discussion.

5.1. Investigating RQ1: Why do Chinese Developers Decide to Use OSS Components?

5.1.1. Results of RQ1

The following motivations for using OSS components were taken from the literature Dagdeviren *et al*. (2005) and Li *et al*. (2005a) and our prestudy: (a) have better potential applications (extendable, easier to update etc.); (b) OSS was already involved in one of the products; (c) reduce development cost; (d) more standardized function and architecture; (e) reduce learning cost; (f) existing knowledge in the company about the OSS; (g) become a part of OSS community and share values of the OSS movement; (h) gain access to products which are not available on the brand-name market. The survey questions were based on the above motivations.

The answers were measured by a five-point Likert scale as 'Don't agree at all', 'Hardly agree', 'Agree somewhat', 'Mostly agree', 'Strongly agree', plus 'Do not know'. We assigned an ordinal number 1 to 5 to the above alternatives (5 means strongly agree, 0 means do not know). The results are shown in Figure 2 and show that changeable applications, compliance with existing components, and low cost were the major motivations (with median value 4 and upward skewness) for using OSS components. Reusing OSS-related knowledge, participating in the OSS community, and acquiring special components were the least important motivations (with median value 3).

5.1.2. Discussion of RQ1

Our results support the conclusion that the motivations for using OSS components are the changeable source code (Li *et al*. 2005a) and low cost of licensing (Fitzgerald and Kenny 2004, Madanmohan and De 2004). Our results reveal that mingling with the OSS community and sharing the ideology of the

DOI: 10.1002/spip

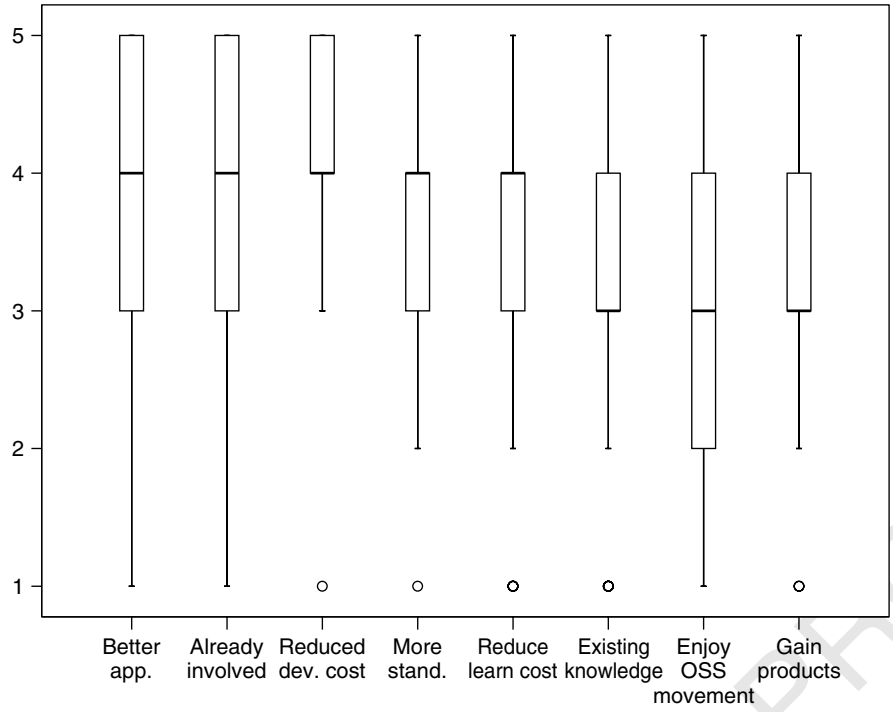

Figure 2. Motivations for using OSS components

OSS movement were not regarded as important by Chinese developers.

5.2. Investigating RQ2: How do Chinese Developers Select OSS Components?

5.2.1. Results of RQ2.1

The survey questions assessed the following methods of locating component candidates: (a) have used it (them) before; (b) from colleagues of the same company; (c) from friends of other companies; (d) through reading related magazines (e.g. programmer magazine); (e) through visiting trade shows and exhibitions; (f) using search engines (e.g. Google); (g) visiting OSS project portals (e.g. sourceforge.net).

The respondents were asked to state whether they had performed such activities to locate OSS candidates. The results reveal that locating OSS candidates was mostly based on either search engines (e.g. Google or the search feature in Sourceforge.net) or internal experience (e.g. having used the components before, reading magazines, getting advice from internal colleagues). External information channels, such as getting advice from persons in other companies, were rarely used.

5.2.2. Results of RQ2.2

Possible criteria for evaluating and comparing component candidates are taken from Madanmohan and De (2004) and Dagdeviren *et al*. (2005), as follows: (a) requirements compliance; (b) architectural compliance; (c) quality of components (security, reliability, usability etc.); (d) functionality; (e) OSS licensing terms; (f) licensing price; (g) reputation of components or supplier; (h) quality of documentation; (i) expected support from the OSS community (updates, bug fixing, clarification etc.); (j) environment or platform. The survey questions were based on the above criteria. We used the same scales as for RQ1. The results illustrate that requirements compliance (with median value 4 and upward skewness) was regarded as the most important criterion to be considered, while licensing price and support were regarded as the least important criteria (with median value 3 and downward skewness).

5.2.3. Discussion of RQ2

Our results support the conclusion that most companies use a manual (brute force) method (Madanmohan and De 2004), e.g. searching with

DOI: 10.1002/spip

Google or Sourceforge.net, to locate component candidates. However, our results show that the developers used Google more frequently than OSS project portals. Although a previous study observed that companies were willing to listen to experience from other companies and were also willing to share their experience with others (Merilinna and Matinlassi 2006), our results reveal that experience sharing between persons in different organizations was not common. One possible reason is that there is a lack of potential channels to share experience of using OSS components between different organizations.

Our results support the conclusion that one of the most important criteria to be considered when evaluating OSS component is requirements compliance, rather than architecture compliance, as was proposed in Madanmohan and De (2004). Although previous studies have claimed that technical support was very important for OSS-based systems (Fitzgerald and Kenny 2004, Tuma 2005), our data provide evidence against that claim and show that the possibility of receiving support from the OSS community was not considered by Chinese developers as critical when they were evaluating components.

5.3. Investigating RQ3: How do Chinese Developers Deal with OSS Licensing Terms?

5.3.1. Results of RQ3.1–RQ3.3

RQ3.1 and •RQ3.3 were measured using Likert scales, as for RQ1. Answers to these research questions are presented in Table 1.

5.3.2. Results of RQ3.4

Possible actions that might be taken to avoid license-related problems were derived from Brown and Booch (2002), Norris (2004), Madanmohan and De (2004), Ruffin and Ebert (2004): (a) use other components without licensing constraints; (b) consult legal experts for help; (c) encapsulate GPL-regulated modules with public application interfaces; (d) package the proprietary code separately to avoid GPL restrictions; (e) contact the owner of the OSS license and agree on a certain license to reduce the effect of licensing; (f) upload all 'derived programs' that are affected by licensing terms back to the OSS community. The survey questions were based on the above courses of action. We used the same scale as for RQ1. The results show that using OSS components without license constraints was the most commonly used strategy, while uploading all 'derived programs' back to the OSS community was the least used strategy.

5.3.3. Discussion of RQ3

The OSS integrators' main concern regarding the terms of licensing is whether the system reusing OSS components is defined as 'a derived program' (Brown and Booch 2002). If so, according to many OSS licenses, the 'derived work' should be published. However, the source code of a system is the private property of a company, which will hide its intellectual property (IP) from its competitors and make profits on IP investments (Madanmohan and De 2004). This concern about IP probably explains our findings that Chinese companies would rather select components without strong licensing constraints, in order to avoid making their entire code public.

5.4. Investigating RQ4: How do Chinese Developers Manage to Maintain and Update OSS-Based Systems?

5.4.1. Results of RQ4.1

We first investigated whether developers needed to fix bugs and to change the source code of OSS

Table 1. Results of RQ3.1–RQ3.3

RQs	Questions in the questionnaire	Results
RQ3.1	How well do developers understand OSS licenses?	Most respondents do not understand OSS licensing terms very well.
RQ3.2	Do developers read related OSS licensing terms?	Respondents have only partly read OSS licensing terms.
RQ3.3	Do developers encounter problems caused by OSS licenses?	Twenty-one percent of the respondents had never encountered OSS license-related troubles. The remaining respondents rarely encountered such problems.

DOI: 10.1002/spip

components. If the answer was 'Yes', the follow-up questions were directed toward what they did and the effort they expended. The results show that 45% of the respondents needed to fix bugs and 39% needed to change code for other reasons. When developers took steps to fix bugs or change code, most of them did it by themselves, rather than ask for help from developers of the OSS community. The developers who fixed bugs by themselves expended more effort (with mean value 40 person-hours) than developers who asked for help from the OSS community (with mean value 11 person-hours). One possible reason is that the OSS community is much more familiar with the code than the developers. The developers who performed general change by themselves expended less effort (with mean value 35 person-hours) than those who asked for help from the OSS community (with mean value 60 person-hours). One possible reason is that an OSS community needs a long time to accept and carry out suggested changes.

To answer RQ4.1 we also investigated whether there were local developers, during the investigated project, who participated in the OSS community. Only four respondents said 'Yes'. For the respondents with 'No' answers, we suggested the following reasons and asked the respondents for their opinions: (a) there was no need to take part in the community; (b) there were insufficient resources (such as time and human resources); (c) it was difficult to participate due to the hierarchy of the OSS community. The results show that most developers thought it is necessary to take part in the OSS community and it is not difficult to join corresponding OSS •projects. However, limited time and personnel prevented them from participating actively in the OSS communities.

5.4.2. Results of RQ4.2

The survey question asked whether the investigated projects have a repository for component knowledge. The respondents were asked to answer 'Yes' or 'No'. The results show that 40% of the respondents answered 'Yes' and the remaining respondents said 'No'. We asked those respondents who answered 'Yes', who is responsible for the repository. The results show that two companies had dedicated teams to maintain the knowledge repository. Other companies had only one senior developer or architect to manage the knowledge related to OSS components. We asked those respondents, who answered 'No', the reasons for not having such a repository, using open questions. The answers can be classified into three categories. (a) There is no need to support such a repository. (b) Human resources and the budget are limited. (c) The quality of components is generally good and components are not overly complex, so tracking components' further maintenance did not require a person dedicated to the task.

5.4.3. Discussion of RQ4

Contributing to OSS projects and getting contributions published are helpful for OSS users (Merilinna and Matinlassi 2006). Our results reveal that most respondents would like to participate in the OSS community. However, few respondents have actually contributed to the OSS community due to limited time and personnel resources. Other ways of participating in the OSS community, such as providing feedback and reporting bugs (Holck *et al.* 2005, Merilinna and Matinlassi 2006) or proposing new features and trial implementations of these features (Tuma 2005, Merilinna and Matinlassi 2006), may be more cost-effective for such respondents. Our results support the observations that most projects did not have a managed, comprehensive component repository (Dagdeviren *et al.* 2005).

5.5. Investigating RQ5: What was the Cost Distribution?

5.5.1. Results of RQ5

The survey questions regarding the possible costs of OSS-based projects assessed the following factors: (a) selection; (b) learning; (c) consulting; (d) developing gluecode; (e) adaptation and configuration; (f) maintenance; (g) upgrade; and (h)license fee. We used the same Likert scale as for RQ1. The results are shown in Figure 3.

The results show that learning, developing gluecode, adaptation, and maintenance are the highest costs (with median value 3 and upward skewness), while consulting costs and license fees are the lowest (with median value 2).

5.5.2. Discussion of RQ5

Learning and understanding OSS components is a new activity in OSS-based software development (Li *et al.* 2006). Although the available source

DOI: 10.1002/spip

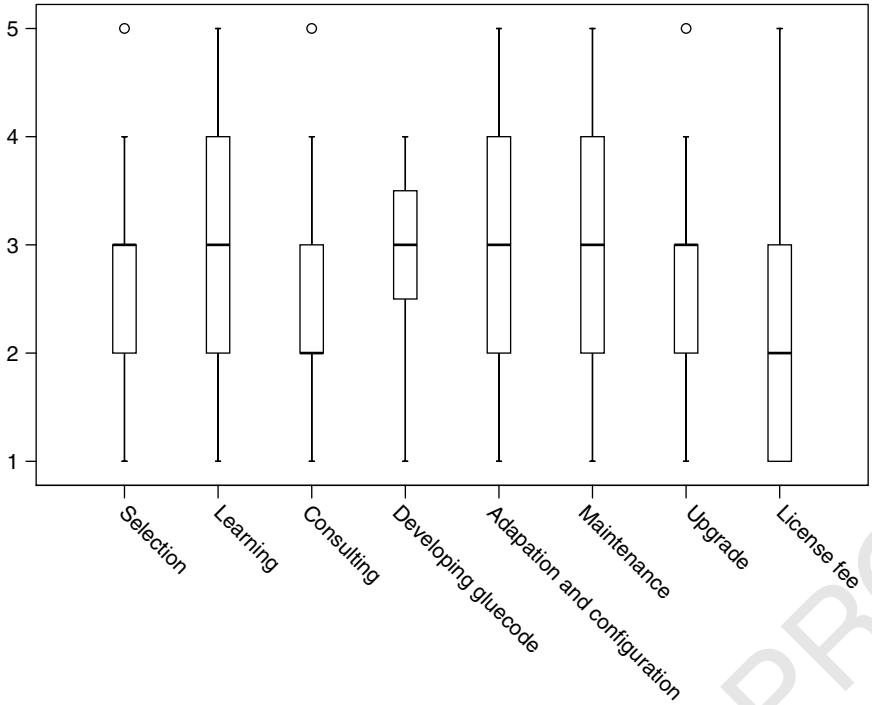

Figure 3. The cost distribution of the investigated projects

code of OSS components may ease the burden of learning and understanding, our results reveal that the effort required to learn components still needs to be considered seriously in effort estimation, as proposed for COTS-based development in Boehm *et al*. (2003). Our results also reveal that consulting costs and license fees are lower than other kinds of cost. A possible explanation is that the source code of OSS is available and there are a lot of valuable resources in the OSS community. Therefore, integrators can, to a large degree, learn and understand OSS components themselves, which means that consulting costs can be saved. The reuse of OSS did not require additional licenses as an installation grows, which may help save license costs (Madanmohan and De 2004).

6. GENERAL DISCUSSION

In this section, we first offer four recommendations to facilitate OSS-based development, using our results as a basis. We then discuss possible threats to validity of this study.

6.1. Recommendation 1: Improve the OSS Search Engine

Although several methods can be used to locate OSS components, our findings regarding RQ2 show that the most commonly used methods are web search engines and OSS project portals, as reported in Madanmohan and De (2004). Using web search engines is simple and fast. However, the search results may be imprecise and huge. OSS projects in OSS project portals are properly classified. However, one OSS project portal cannot include all OSS projects. People have to search in several portals to get all possible component candidates. Although the new 'Google Code Search' helps solve the above shortcomings by combing portals of the open-source domain, greater effort is needed to facilitate the search for OSS components beyond the source code level.

6.2. Recommendation 2: Understand and Comply with OSS Licensing Terms

To develop commercial software with OSS components, it is important for OSS users to carefully read, understand, and comply with the licensing terms

DOI: 10.1002/spip

of the OSS component being used (Madanmohan and De 2004). Our results regarding RQ3 show that most respondents did not read and understand the OSS licensing terms properly. There are more than 50 OSS licenses approved by opensource.org. However, five major license types (i.e. GPL, LGPL, BSD, AL, and MIT) cover 90% of OSS projects (Tuma 2005, Ueda 2005). It may be wise for OSS users to learn and understand these major license types before they start to select and integrate OSS components.

6.3. Recommendation 3: Participate More Actively in the OSS Community

During the maintenance phase of an OSS-based project, project developers may need to fix bugs in OSS components and to add or revise the components' functionalities. Our results regarding RQ4 show that our respondents used more effort, on average, to fix bugs than did developers of the OSS community. Thus, a better way to get bugs fixed might be to report bugs and ask for help from the OSS community. One study shows that 83% of OSS community participants live in Western countries and 55% of them contribute to OSS projects during working hours (Lakhani and Wolf 2005). However, our results show that only 9% of the investigated projects had dedicated developers taking part in an OSS community during the project. Thus, one of the primary tasks of Chinese users •should be to mingle with the OSS community (Wang and Zhang 2004).

6.4. Recommendation 4: Facilitate the Sharing of Internal and External Experience

Our results regarding RQ5 show that learning cost is one of the major costs of OSS-based projects. The proper reuse of previous experience and knowledge will reduce the later learning cost. Although the results regarding RQ2 show that internal expertise is consulted when selecting and evaluating OSS components, the results regarding RQ4 show that most companies do not have a systematic mechanism for managing knowledge so that developers and maintainers can share experience. In addition, the results regarding RQ2 show that knowledge sharing between companies is rare. To facilitate knowledge sharing internally, one possible strategy is to have an internal component 'knowledge keeper' (Li *et al*. 2006). Such a person can collect thorough knowledge of relevant OSS components and evaluate the newest version of the components. Another possible remedy is to create a human (and/or computer supported) knowledge repository (Morad and Kuflik 2005). To facilitate the sharing of experience between different companies, a centralized experience portal for sharing OSS •component-related knowledge between organizations, probably using a global OSS Wiki (Ayala *et al*. 2007), could be a solution.

6.5. Possible Threats to Validity

Our unit of study was a finished project. Thus, a possible threat to the internal validity of this study is that the respondents may have failing memory on past events. Since China has no comprehensive, national database of software companies, it is difficult to select a random sample of participants in such surveys, even if the present one is maybe as good as we can get. This may bring external validity threats to our conclusions.

7. CONCLUSIONS AND FUTURE WORK

More and more software companies in China and elsewhere are reusing OSS components as part of their software development. Such companies need empirically based guidelines for OSS-based development. The main findings from our survey are these:

- Developers who use an OSS component focus on its potential application, such as being extensible and updating easily. However, mingling with the OSS community to share the value of the OSS movement and to gain brand products was not regarded as being as important as potential application.
- The selection of OSS components is based mainly on existing web search engines, followed by local expertise for evaluation and decision.
- OSS licensing terms are not a barrier to software companies when reusing OSS components in system development.
- In 84% of the development projects, system maintenance leads to bug fixing or other code changes in the selected OSS components and

DOI: 10.1002/spip

involves the OSS community on a case-to-case basis.

- The learning cost is one major expense when reusing OSS components. We recommend that the experience and knowledge pertaining to relevant OSS components are handled by an internal 'knowledge keeper', a global OSS Wiki, and more active participation in the OSS community. This last course of action is also expressed by the developers themselves, but not followed up, perhaps for cultural and organizational reasons.

By 2011, at least 80% of commercial software will contain significant amounts of open source code, according to Gartner (Computerworld 2007). Although we were the first to perform such an empirical study in the industry in China, we have, thus far, collected a small amount of data. We will perform further studies to align ourselves with the latest progress in this field.

REFERENCES

Abts C, •*et al*. 2000. COCOTS: A COTS Software Integration Lifecycle Cost Model – Model Overview and Preliminary Data Collection Findings. Technical report USC-CSE-2000-501, http://sunset.usc.edu/publications/TECHRPS/2000/usccse2000-501/usccse2002-501.pdf.

Ayala C, *et al*. 2007. Open source collaboration for fostering off-the-shelf components selection. *Proceedings 3rd International Conference Open Source Systems*. •Springer: 17–30, DOI: 10.1007/978-0-387-72486-7_2.

Boehm BW, *et al*. 2003. Not all CBS are created equally COTS-intensive project types. *Proceedings 2nd Conferences COTS-based Software Systems*, LNCS Vol. 2580. Springer: Ottawa, 36–50.

Briand LC. 1998. COTS evaluation and selection. *Proceedings International Conference Software Maintenance*, Bethesda, 222–223, DOI: 10.1109/ICSM.1998.738512.

Brown AW, Booch G. 2002. Reusing open-source software and practices: the impact of open-source on commercial vendors. *Proceedings 7th International Conference Software Reuse*, LNCS, Vol. 2319. Springer: Austin, TX, 123–136.

Chen W, *et al*. 2007. A survey of software development with open source components in chinese software industry. *Proceedings International Conference Software Process*, LNCS Vol. 4470. Springer: Minneapolis, MN, 208–220, DOI: 10.1007/978-3-540-72426-1_18.

Computerworld. 2007. •http://www.computerworld.com.au/index.php/id;1025846193.

Dagdeviren H, *et al*. 2005. An exploratory study for effective COTS and OSS product marketing. *Proceedings 27th International Conference on Information Technology Interfaces*, Cavtat, Croatia, 644–649, DOI: 10.1109/ITI.2005.1491205.

Fitzgerald B, Kenny T. 2004. Developing an information systems infrastructure with open source software. *IEEE Software* **21**(1): 50–55, DOI: 10.1109/MS.2004.1259216.

Giacomo PD. 2005. COTS and open source components: are they really different on the battlefield?. *Proceedings 4th International Conference on COTS-Based Software Systems*, LNCS, Vol. 3412. Springer: Bilbao, 301–310, DOI: 10.1007/b105900.

Holck J, *et al*. 2005. Managerial and technical barriers to the adoption of open source software. *Proceedings 4th International Conference COTS-Based Software Systems*, LNCS, Vol. 3412. Springer: Bilbao, 289–300, DOI: 10.1007/b105900.

Kshetri N. 2005. Structural Shifts in the Chinese Software Industry. *IEEE Software* **22**(4): 86–93, DOI: 10.1109/MS.2005.107.

Lakhani K, Wolf RG. 2005. Why hackers do what they do: understanding motivation and effort in free/open source software projects. *Perspectives on Free and Open Source Software*. MIT Press: Cambridge, 3–22.

Li J, *et al*. 2004. An empirical study of COTS component selection processes in norwegian IT companies. *Proceedings International Workshop on Models and Processes for the Evaluation of COTS Components*, Edinburgh, 27–30.

Li J, *et al*. 2005a. An empirical study on off-the-shelf component usage in industrial projects. *Proceedings 6th International Conference Product Focused Software Process Improvement*, LNCS Vol. 3547. Springer: Oulu, 54–68, DOI: 10.1007/11497455_7.

Li J, *et al*. 2005b. Preliminary results from a state-of-the-practice survey on risk management in off-the-shelf component-based development. *Proceedings 4th International Conference COTS-based Software System*, LNCS Vol. 3412. Springer: Bilbao, Spain, 278–288, DOI: 10.1007/b105900.

Li J, *et al*. 2006. An empirical study of variations in COTS-based software development process in norwegian IT industry. *Journal of Empirical Software Engineering* **11**(3): 433–461, DOI: 10.1007/s10664-006-9005-5.

Madanmohan TM, De R. 2004. Open source reuse in commercial firms. *IEEE Software* **21**(6): 62–69, DOI: 10.1109/MS.2004.45.

Morad S, Kuflik T. 2005. Conventional and open source software reuse at orbotech – an industrial experience. *Proceedings IEEE International Conference Software – Science, Technology & Engineering*, Herzelia, Israel, 110–117, DOI: 10.1109/SWSTE.2005.11.

Merilinna J, Matinlassi M. 2006. State of the art and practice of open source component integration. *Proceedings 32nd EUROMICRO Conference Software Engineering and Advanced Applications*, Cavtat, Croatia, 170–177, DOI: 10.1109/EUROMICRO.2006.61.

Ncube C, Maiden N. 2004. Selecting COTS anti-virus software for an international bank: some lessons learned! *Proceedings International Workshop on Models and Processes for the Evaluation of COTS Components*, Edinburgh, 17–21, DOI: 10.1049/ic:20040427.

Norris JS. 2004. Mission-critical development with open source software: lessons learned. *IEEE Software* **21**(1): 42–49, DOI: 10.1109/MS.2004.1259211.

Open Source Initiative. 2005. •Available at http://www.opensource.org/index.php.

Ruffin M, Ebert C. 2004. Using open source software in product development: a primer. *IEEE Software* **21**(1): 82–86, DOI: 10.1109/MS.2004.1259227.

Spinellis D, Szyperski C. 2004. How is open source affecting software development? *IEEE Software* **21**(1): 28–33, DOI: 10.1109/MS.2004.1259204.

Tuma D. 2005. Open source software: opportunities and challenges. •*Journal of Defence Software Engineering*: 6–10.

Ueda M. 2005. Licenses of open source software and their economic values. •*Proceedings 2005 Symposium Applications and the Internet Workshops*, 381–383, DOI: 10.1109/SAINTW.2005.1620054.

Wang H, Wang C. 2001. Open source software adoption: a status report. *IEEE Software* **18**(2): 90–95, DOI: 10.1109/52.914753.

Wang G, Zhang X. 2004. Chinese linux open source encounters close. *IT Time Weekly* **22**: 20–21.