An Empirical Study on Decision Making in Off-The-Shelf Component-Based Development

Jingyue Li¹, Reidar Conradi¹,², Odd Petter N. Sylgstad¹
¹Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway
²Simula Research Laboratory, P.O. BOX 134, NO-1325 Lysaker, Norway
{jingyue, conradi, osylngst}@idi.ntnu.no

Christian Bunse³
³Fraunhofer IESE, Fraunhoferplatz 1, D-67663 Kaiserslautern, Germany
Christian.Bunse@iese.fraunhofer.de

Marco Torchiano⁴, Maurizio Morisio⁴
⁴Dip. Automatica e Informatica, Politecnico di Torino
Corso Duca degli Abruzzi, 24, I-10129 Torino, Italy
{marco.torchiano, maurizio.morisio}@polito.it

ABSTRACT
Component-based software development (CBSD) is becoming more and more important since it promotes reuse to higher levels of abstraction. As a consequence, many components are available being either open-source software (OSS) or commercial-off-the-shelf (COTS). However, it is still unclear how the decision for acquiring OSS or COTS components is made in practice. This paper describes an empirical study on why project decision-makers selected COTS instead of OSS components, or vice versa. The study was performed as an international survey in Norway, Italy and Germany. It focused on decision making on using off-the-shelf (OTS) components. We have gathered answers from 83 projects using only COTS components and 44 projects using only OSS components. Results of this study show significant differences and commonalities of integrating OSS or COTS components. Moreover, the study illustrates several research questions that warrant future research.

Categories and Subject Descriptors
D.2.13 [Software Engineering]: Reusable Software – Reuse model

General Terms: Management

Keywords: Commercial-Off-the-Shelf Software, Open Source Software, Empirical Study, Component-based Development

1. INTRODUCTION
Long-term success in commercial software development is becoming increasingly challenging. The software industry is increasingly turning to approaches which offer a larger return on software development effort. Component-based software development (CBSD) [19] is one of these approaches. Besides reusing in-house built components, software companies are using an increasing amount of COTS and OSS components. Although both COTS and OSS component claim to save development effort, they are different in their nature and appearance. COTS components are owned by commercial vendors with agreed specific support. The users of COTS component normally have no access to the source code. On the other hand, OSS components are provided by open source communities with freely accessible source code but promise no specific support [14].

When planning a new software project, decision makers must decide whether they should buy a COTS component or acquire an OSS component, if it was decided to use OTS components. To make such a decision, it is important to investigate previous projects using such components. Unfortunately, the relevant decision-making processes are often intuitive, and if at all, only sparsely documented. Thus, further research is needed to examine the current state-of-the-practice and to define guidelines for OTS-based development.

This paper investigates the state-of-the-practice in OTS-based development on use of COTS and OSS components, by means of an international representative survey in Norway, Italy and Germany. We have gathered answers from 83 projects using only COTS components and 44 projects using only OSS components. Results of this study illustrated who are using OTS components, why was it decided to use such components, and what were the problems. Results also illustrated the commonalities and differences in integrating COTS or OSS components.

The remainder of this paper is structured as follows: Section two presents related research. Section three introduces research design. Section four presents obtained results and discussion. A short summary and some future work are given in Section five.

2. RELATED WORK
COTS components promise faster time-to-market and increased productivity. At the same time, COTS software introduces many risks, such as the unknown quality of COTS components, or possible instability of the COTS vendor support [20]. Although
some research stated that OSS makes only sense for operating systems [5], the use of OSS in industrial products is growing rapidly [11, 13, 16, 18]. OSS offer many advantages: it is usually freely available for public download. The collaborative, parallel efforts of globally distributed developers allow much OSS to be developed more quickly than conventional software. Many OSS products are recognized for high reliability, efficiency, and robustness [4]. Despite its wide appeal, OSS software faces several challenges and constraints: the documentation, testing, and field support may not be available [4], and long-term support is not guaranteed [11]. The OSS license may restrict the user of changing and integrating the OSS components [18]. Previous studies have looked at using COTS and OSS components in software development [2, 13]. However, few studies have investigated a higher level question: Why do I use COTS instead of OSS components, or vice versa?

Some studies compared the differences between COTS and OSS products per se and concluded that there is still empirical evidence that OSS fosters faster system growth and that OSS is more modular than closed source software [15]. However, there were few systematic empirical studies of the commonalities and differences in integrating COTS vs. OSS components in a software product. In order to guide the project decision maker in deciding between an OSS component and a COTS component, it is necessary to empirically investigate how the decision was made and what was the result of the decision in finished projects.

3. RESEARCH DESIGN

The empirical study on decision making OTS based development followed a three-step approach. We started from a pre-study of state-of-the-practice and the challenges in integrating COTS components. The pre-study was performed by structured interviews of 16 COTS projects in 13 Norwegian IT companies selected by convenience [8]. The study reported in this paper represents the second step, and investigated decision making in OTS component-based development by a questionnaire-based survey. This study extended the pre-study in two dimensions. First, it included OSS. Second, the study included randomly selected IT companies from Norway, Italy and Germany. The third future step will focus on investigating the underlying reasons of the conclusions in the second step and more studies on OSS projects. It will be performed with structured interviews and case studies.

3.1 Research Questions

Based on the design of the study a couple of research questions were defined. First, we wanted to know the profiles of those using OTS components. Therefore, the first research question is:

- RQ1: What are the commonalities and differences in profiles of projects using COTS components vs. those using OSS components?

We then wanted to know their expectation of using OTS components. The second research question therefore is:

- RQ2: What are the commonalities and differences in the expectation of projects using COTS components vs. those using OSS components?

After we know who and why, we need to know the actual risks/problems of using such components. We investigated whether the motivation and expectation of using OTS components are proper or not. So, the third research question is:

- RQ3: What are the commonalities and differences in possible risks (problems) of projects using COTS components vs. those using OSS components?

3.2 Research Method

In this study, we used a questionnaire to collect data. The detailed information of the questionnaire and sample selection are reported in [3, 9, 10].

4. RESULTS AND DISCUSSION

In [10], we reported the preliminary results of this study with 71 projects using only COTS components and 39 projects using only OSS components. Now, the data collection is finished. In total, we gathered results from 83 projects using only COTS components, 44 projects using only OSS components, and six projects using both COTS and OSS components.

4.1 RQ1: What was the Profile of Projects Using OTS Components?

To answer RQ1, we investigated the commonalities and differences of some requirements of the actual system, such as time-to-market, effort (cost), reliability, security, performance, maintainability, new functionality (first launch in the market), and improved functionality (over competitors).

We compared each requirement in systems using COTS vs. systems using OSS components. The results show that there are no such significant differences. The most emphasized requirements of the final systems were time-to-market, reliability, and performance. Security was surprising listed as the least emphasized requirement of the system. There are many discussions on whether OSS components are more secure than COTS components. Raymond [17] argued that “given enough eyeballs, all bugs are shallow.” Other argued that it is the quality, not the number, of the eyes looking at code that count [7]. Our results show that OTS component users are still very cautious of using either COTS or OSS components in the security-critical systems. So, software security aspect may not be distinguished between COTS and OSS components. However, trust in the security of third-party components may become an issue in the future.

4.2 RQ2: What were the Expectations of Using OTS Components?

To answer RQ2, we investigated the respondent’s general expectations using OTS components, and specific expectations of using COTS and OSS components.

The general expectations of using OTS components are extracted from our pre-study [8] and literature reviews [4, 20]. Details of these expectations are shown in [10]. Results show again that there are no significant differences. The main expectation of using OTS components is to shorten the time-to-market and to save development cost. Pizka [16] reported experience on building the same system with three different strategies, such as wrapping an
OSS component, adapting/changing the source code of the OSS component, or building the same component from scratch. Their results gave strong support on reusing OSS components and concluded that re-implementing existing OSS components should be avoided if there are no serious reasons for doing so.

The specific expectations of using COTS components listed in the questionnaire were shown in [10]. Results show that COTS users believe that paid software will give good reliability and will follow the market trend. They also believe that the COTS vendor will provide good technical support. We also listed specific expectations of using OSS components listed in the questionnaire. Results show that the main motivation of using OSS components is that code could be acquired for free and that the source code is available.

In summary, results show that business considerations, such as reducing time and cost, are the main motivation for using OTS components. OSS component users care more about the initial investment. Another attraction of OSS components is that the marginal cost of scaling up is zero, i.e. using OSS does not require additional licenses as the number of installations grows [18]. This benefit is especially important when integrating OSS components into a software product with many installation bases.

Concerning the use of domain-specific OTS components, it is important to ensure that the evolution of the selected OTS component will follow the market trend. Although a COTS user cannot directly influence the evolution of the selected COTS component, results of this study show that these users still trust that the COTS vendor will update the component according to market needs. OSS components can be changed out of similar market reasons, but can also be changed out of political or social reasons – not necessarily in a direction suitable to the organization [6]. It is therefore important for OSS users to ensure that they are planning to use the software in the way it was intended to be used and the way most other users are using it. If OSS users stray too far away from the beaten path, the OSS developers might not want to help them when they run into trouble, or OSS developers will not accept the changes contributed by them [11].

4.3 RQ3: What were the Risks of Using OTS Components?

To answer RQ3, we formulated 15 possible problems (see Table 3 in [10]) on OTS-based development, and asked the respondents if these problems had happened in their projects. Results showed that common problems both in projects using COTS and in those using OSS are: the effort to integrate OTS components was not satisfactorily estimated and it was difficult to identify whether the defects were inside or outside the OTS component.

The estimation of the component-integration effort in COTS-based development projects has been investigated by several studies [1], whereby the estimation of integration effort in OSS-based development has been neglected so far. It is therefore interesting to know the commonalities and differences in cost models for COTS-based vs. OSS-based projects.

Without source code, it is difficult for COTS component users to identify whether the defects are inside or outside the component. Although the source code of OSS components is available to their users, many OSS users did not change or even read its source code. As Hissam and Weinstock proposed in [6], OSS is the other commercial software, i.e. they were used as COTS in practice [9, 11]. OSS user might not have bothered to understand the source code of an OSS component before starting to use it. It is therefore difficult for them to figure out if the defects are inside an OSS component or not.

Results of RQ3 also show that COTS users had significantly more difficulties in estimating the component selection effort and following requirement changes. It is not easy or cost-effective for COTS users to switch to another COTS component once they have made the decision and paid for some selected components. Therefore, COTS users might be more cautious in selecting the right COTS component and spend more effort than estimated on selecting the best one. Without paying for the OSS components, OSS users might be less cautious and might spend less effort on selecting the best OSS component from several candidates. In addition, it might be easier for OSS users to change the source code of the OSS component if their customers’ requirements do change. Therefore, OSS components are ideal when the software solution requires high customization [11].

OSS users had more problems on getting the information on a providers’ support reputation. Although the preliminary data [10] showed the differences between the OSS users and COTS users were significant, the current data (six months after the analysis of the preliminary data) does not show such significant differences any longer. This might be because more and more commercial organizations provide support to OSS components. In fact, OSS components with support by a commercial organization usually become more popular than components without such support [12]. In addition, more and more OSS portals, such as sourceforge.net and freshmeat.net, have started to measure the vitality and popularity of an OSS project. These metrics might have helped OSS users to estimate the quality and duration of support they can expect.

5. CONCLUSION AND FUTURE WORK

This paper has presented the results of an international empirical study to investigate the decision making of integration COTS or OSS components in software products. In general, it appears that the motivation for using either COTS or OSS components and the risks involved are important factors to be concerned. Future research has to pay specific attention to these facts before defining systematic guidelines and methods. Results of this study have answered the following three questions:

RQ1: Who were using OTS components?

Both COTS and OSS components are used in projects with different application domains and different non-functional requirements. There was no significant difference between the profiles of COTS-based and OSS-based projects.

RQ2: Why was it decided to use OTS components?

The main expectation of using either COTS or OSS components is to obtain shorter time-to-market and less development effort. COTS users have higher expectation on COTS component quality and COTS vendor support. Possible no-cost source code is the key motivation of OSS users. OSS users prefer to have access the source code, so that they can revise it when necessary.
RQ3: What are the possible problems of using OTS components?

It is more difficult for COTS component users to follow requirement changes than OSS users. It is also more difficult for COTS users to estimate the selection effort. The uncertainty of OSS users is higher than that of COTS users concerning the providers’ support reputation.

Based on these results we identified several emerging research questions that should be investigated in the near future:

- **Understand the reasons behind the current results**

This study is a state-of-the-practice survey that provides evidence of some phenomena in OTS-based development. These phenomena have to be studied in more detail to better understand the nature of OTS projects and to provide systematic support. Examples of interesting issues are: How is the support capability of an OSS project estimated, how is support for OSS components realized, and how to avoid the licensing restriction.

- **Understand the OSS projects per se**

To provide guidelines on using OTS component in practice, it is important to understand how OSS projects are organized. Furthermore, it is important to know how to cooperate with OSS developers, for example, how to ensure that changes of users are accepted by the OSS community, and how to motivate developers to enhance and adapt OSS component into a desirable direction. Another important issue is the definition of metrics to obtain reliable data about the concrete nature of OSS projects and products. For example, the organization support proved to give a positive effect on the popularity of an OSS component [12]. Future studies have to examine OSS projects in order to identify, define, and validate more such measures. A good candidate seems the question: whether the vitality and popularity of data provided by an OSS portal gives insight on the longevity of an OSS projects?

6. REFERENCES


