Information and Software Technology 53 (2011) 616–624

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

A systematic review of research on open source software in commercial software product development

Martin Höst *, Alma Oručević-Alagić

Department of Computer Science, Lund University, P.O. Box 118, SE-221 00 Lund, Sweden

ARTICLE INFO

Article history:
Available online 25 December 2010

Keywords:
Open source software
Proprietary
Commercial
Component based software engineering
Business models

ABSTRACT

Context: The popularity of the open source software development in the last decade, has brought about an increased interest from the industry on how to use open source components, participate in the open source community, build business models around this type of software development, and learn more about open source development methodologies. There is a need to understand the results of research in this area.

Objective: Since there is a need to understand conducted research, the aim of this study is to summarize the findings of research that has ben carried out on usage of open source components and development methodologies by the industry, as well as companies' participation in the open source community.

Method: Systematic review through searches in library databases and manual identification of articles from the open source conference. The search was first carried out in May 2009 and then once again in May 2010.

Results: In 2009, 237 articles were first found, from which 19 were selected based on content and quality, and in 2010, 76 new articles were found from which four were selected. Twenty three articles were identified in total.

Conclusions: The articles could be divided into four categories: open source as part of component based software engineering, business models with open source in commercial organization, company participation in open source development communities, and usage of open source processes within a company.

1. Introduction

Traditional software development is often perceived as a proprietary, in-house software development, with developers working in a geographically centralized or distributed company's location. Open source software is developed free of charge through a community driven development process, and as such, it is also provided to public at no cost, but under certain usage and distribution conditions. Many of the traditional software companies have tried to take advantage of the free software, not just by using the software, but also by creating business models and strategies around the open source software.

For example, in the mobile industry there are several attempts to form open source communities for development of software, such as the Android project[1] and the Symbian project.[2] Using and relying on open source software can be seen as an alternative way to reduce development costs and stay competitive. Hence, in a way, it can be compared to other similar business methods and strategies, such as outsourcing or acquirement of off-the-shelf components.

This open source business ecosystem, which has been growing over the past two decades, is quite complex and there exists a need to better understand many of its aspects. Some of the aspects are interesting in at least two different ways. Firstly, an organization can include open source components in its proprietary software product. This is comparable to including any other third-party component, although the difference is that the component is now obtained from an open source community instead from a commercial organization. Secondly, an organization can provide its own proprietary software to open source community and that way reduce development costs in long run, reposition itself on the market, create a new source of income through new services, etc.

Already in 2001, Lerner and Tirole [1] identified "opening proprietary code" as an important research area, and observed that large open source projects often start based on software provided by "academic or semi-academic institutions". This motivates systematically investigating what research has been published in the area.

The outline of this paper is as follows. In Section 2 background on open source software and some related work is presented. In Section 3 the methodology with respect to search strategy and

* Corresponding author. Tel.: +46 46 222 90 16.
E-mail addresses: martin.host@cs.lth.se (M. Höst), alma.orucevic-alagic@cs.lth.se (A. Oručević-Alagić).

[1] http://www.android.com/.
[2] http://www.symbian.org/.

0950-5849/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2010.12.009

inclusion and exclusion criteria are presented, and the resulting set of articles is presented in Section 4. Finally, there is a discussion in Section 5, and conclusions presented in Section 6.

2. Background and related work

2.1. Open source software

Open source software has been around since the very beginning of electronic computing. In the early days of information technology it was quite natural and financially sound for developers to share source code among very few and very expensive computing machines. As the machines became smaller, more diversified, and cheaper, the number of developers grew, and the source code, in general, became more complex. Development of free software was especially flourishing in the academic environments. Barkley Software Distribution (BSD) is a license developed for distribution of the BSD version of the Unix operating system developed by the University of California, Berkeley, from 1977 to 1995 in collaboration with AT&T labs, as described in [2]. At the beginning of the development, code was shared between AT&T and Berkeley. Due to anti-monopoly laws at the time, AT&T could not sell software, but as the company was using it to sell phone-related services, it had vested interest in improving the software. During the beginning of the 1980s and the market deregulation, AT&T was granted the right to sell software. In order to continue distribution of BSD Unix, a lot of code that was not developed by University of California Berkeley had to be backed off and rewritten.

Since the beginning of 1980s, the idea of close-sourced/proprietary software became mainstream, taking the place that free software sharing has held for a long time. The open source supporters went to found their own organizations such as free software foundation (FSF) founded by Richard Stallman, as described in [3]. The FSF did not have desired impact on bringing back open source software development to the mainstream. However, this situation was about to change with the successful release of the Linux kernel. The system was initially developed by Linus Torvalds as part of an academic project, and with the support of the developer community it became a very complex, sophisticated software that was free for everyone to use. Eric Raymond was very much inspired by this set of events, and in his now famous book "The Cathedral and the Bazaar" [2] he talks about the importance of Linux, as it was the very first time the open source developer community showed that not only complex and sophisticated software can be built in such way, but also that business models can be built around such way of software development and distribution.

In 1998, Raymond was one of the main contributors to the Open Source Initiative (OSI), an organization that is envisioned as open source educational and advocacy organization. Many companies have followed the suit, and decided to open source a piece of their proprietary software as a part of business strategy to deal with the competition. Thus, among the initial suitors we can find Netscape corporation, who by open sourcing Netscape internet browser tried to compete against closed source and free distribution of Microsoft's Internet Explorer [2].

In the past 10 years, many companies have entered the open source business arena, using some of the business models proposed in [2]. Unfamiliar with the environment, companies had very quickly to readjust their way of doing business in order to ripe some perceived benefits of open source trends. Besides open sourcing software, companies tend to participate and contribute to open source projects, as well as adopt some of software development methodologies such distributed and voluntary based development community as open source utilizes.

2.2. Related work

Stol and Ali Babar [4] have made a review of the broad area of "open source" from the conference on Open Source Systems, OSS. They manually selected empirical papers from the conference and investigated them. The scope of the review that we present in this paper is more narrow (open source in commercial organizations) but we searched a broader set of articles (we searched articles in library databases with a search string, and we searched articles from the OSS conference manually, as explained in more detail below).

Stol and Ali Babar [5] have also compiled a list of challenges in using open source as components in product development, based on a literature review. In this review, where they did not require any empirical grounding of the findings, they identified 21 challenges.

In [6] earlier results of this work is presented, based on a search in bibliographic databases that was carried out 18 May 2009. The search that is the basis for this paper was conducted 14 May 2010, and it resulted in 4 additional articles.

3. Review method

This research is carried out as a systematic literature review, based on the guidelines presented in [7].

3.1. Research questions

The objective of this research is to understand the result of the research that has been carried out on the usage of open source software and open source software development in proprietary software development organizations. Before the review, this was broken down to the following research questions:

1. What approaches and processes are applied by commercial organizations to introduce open source products in their proprietary products?
2. What approaches and processes are applied by commercial organizations to provide their software products to the open source community?
3. What experience is available from identified approaches and processes, for example, with respect to quality of the software products, cost of development for the providing organization, time taken to introduce new functionality, etc.?
4. What are the main motivations and business incentives for the procedures and processes identified in question 1 and question 2?

That is, we address the need to understand both what research that has been done in the area, what methods and approaches that exist, and what experience is available for the different methods. It should be noted that the objective of the research has not been to derive quantitative knowledge of which methods perform the best. The objective is more to understand which methods are used and how well the methods work in a qualitative way. If the field was more mature, and it could be expected to find a large number of empirical studies investigating the performance of alternative methods, it would of course be interesting to synthesize this knowledge. However, it is not realistic to find this many studies of this type. The objective of this work is instead to review the research that has been conducted, and in particular what kind of experience that is available for these kind of questions. That is, the research has elements of a mapping study (see for example [7]). However, since the objective is to summarize the findings and to understand the total result of the research that has been