OSS Industry Savings

SIG report on software industry savings resulting from use of open source software

April 2010

Miguel Ferreira
Joost Visser
research@sig.eu

public

Software Improvement Group

P.O. Box 79071
1070 NC Amsterdam
The Netherlands
t +31 20 314 0950
f +31 20 314 0955
info@sig.eu
www.sig.eu

Executive Summary

Open source software may be used as a building block for software development in commercial settings. Can the economic impact of such reuse be quantified?

This report presents research on the usage of open source libraries in commercial software. The results show that open source libraries are widely used, and their usage has introduced estimated savings of around € 375,000,000 in a set of about 330 systems. This amounts to an estimated average of about € 1,1 million development cost savings per commercial system.

We conclude that open source software has a significant positive impact on the global economy through its reuse in proprietary systems.

Table of contents

1 Introduction

Software has a key role in most areas of modern society. A fair share of all software in use is open source, meaning that its source code is available to inspect, extend, use, and re-distribute. Although there is money to be made by developing and supporting open source software, there is no point in selling it since anyone can build the software products from the freely available source code. This means the efforts of the communities that develop open source software are contributing to the global economy indirectly.

The contributions made by open source projects range from complete operating systems (Linux) and suites of office applications (OpenOffice), to tools, components, and libraries used in the construction of software products. For example, Eclipse is an open source Integrated Development Environment used by software engineers to support their programming tasks. MySQL is an open source database management system that is used as a component with which other components of a system communicates. Log4j is a library with standard logging functionality that is invoked from the source code of a software system to program new components.

In this report, we focus on the use of open source libraries for the construction of commercial systems and the indirect economic contribution that results from this.

In the remainder of this report, we address the following research questions:

- Which open source libraries are most commonly used in commercial software?
- In which combinations are open source libraries used in commercial software?
- What is the economic value of the use of open source libraries in commercial software?

In Chapter 2 we describe the design of our study. In Chapter 3 we present the results. In Chapter 4 we discuss the significance of our findings.

2 Study design

To answer the previously mentioned research questions, we conducted an analysis on a repository of commercial systems. In this section we explain the origin of the systems and the steps of the analysis process.

Systems selected for analysis

The systems in this repository have been supplied to SIG by its customers in the context of its commercial assessment and monitoring services. The purpose of these services is to provide IT management of large organisations with recommendations for risk mitigation, strategic planning, progress assurance, etc.

As part of these services the software analysis laboratory of SIG performs tool-assisted analysis and evaluation of the source code of the systems. The analysis and evaluation is performed according to standardized procedures, guarded by a quality management system conform ISO/IEC 17025.

As provider of these services, SIG has access to a large number of commercial software systems that cover many different domains of application: information systems, embedded systems, workflow support and management, airport passenger check in, etc. Also, these systems make use of a wide variety of programming platforms, ranging from COBOL mainframes to modern web-based applications.

From the available systems, we selected those that have Java as their main development language, and that make use the Maven framework for building software products from Java source code. These selection criteria guarantee that we can extract precise information about the dependencies of each system on external libraries and frameworks.

Analysis steps

The analysis was conducted in three steps:

1. **System extraction**: Raw data is extracted from the source code of each system regarding the internal dependencies of its components and the external dependencies on libraries. More than 300 systems that complied with the defined criteria (Java as main language and Maven as build system) where scanned.

2. **System processing**: Aggregation steps are applied to the raw data to determine which libraries are used by which systems. We exploit dependency data recorded in the Maven configuration files to determine (i) which source code parts together form a complete system and (ii) which external dependencies refer to sub-parts of a single library.

3. **Library retrieval**: Among the used libraries, those are identified that (i) are open source and (ii) are used by a significant number of systems. For these libraries, the source code of open source libraries used by the analysed systems is

retrieved. Libraries that are used by only a small number of systems are not considered representative for commercial use of open source.

4. **Library cost analysis**: Based on their source code an estimation was made of the rebuild cost of each of the retrieved libraries. To this end we made use of industry-average productivity statistics.

5. **Savings analysis**: The rebuild cost of the retrieved libraries was combined with the frequency of their use by the analysed commercial systems to estimate the overall savings achieved by using the libraries rather than building similar functionality into the separate systems.

The results of the analysis are presented in the following chapter.

3 Study results

Based on the selection criteria mentioned above, we have selected and scanned the source code of 334 commercial software systems. The scanned systems were determined to have a total of 445 external dependencies. Analysis of the data extracted from these systems has allowed us to answer the following research questions:

- Which open source libraries are most commonly used in commercial software?
- In which combinations are open source libraries used in commercial software?
- What is the economic value of the use of open source libraries in commercial software?

We will provide the answers to these questions below.

Which OSS libraries are most commonly used in commercial software?

Table 1 lists the ten open source libraries that are most frequently used in our corpus of commercial software systems. The table provides the name of the library, the percentage of systems that use it, a description of the library, an indication of the type of functionality it offers, and the open source license under which it is distributed.

Table 1: Ten most frequently used open source libraries used in commercial systems.

Library	Systems	Description	Kind	License
Spring Framework	42%	Application framework for Java and .NET	Application framework	Apache 2.0
JUnit	42%	Unit test framework for Java	Testing	Common Public 1.0
Log4j	39%	Logging framework for Java	Logging	Apache 2.0
Javax Servlet	22%	The javax.servlet package describes and defines the contracts between a servlet and the runtime environment.	GUI	Apache 2.0
Commons Logging	19%	Implementation independent logging utility.	Logging	Apache 2.0
Hibernate	17%	Object-relational mapping library.	ORM	LGPL 2.1
Commons Lang	16%	Extra functionality for Java lang package.	API extension	Apache 2.0
EasyMock	15%	Mock Objects for interfaces and classes.	Testing	Apache 2.0
Slf4j	13%	Implementation independent logging utility.	Logging	MIT
Commons Collections	12%	Extra functionality for Java collections.	API extension	Apache 2.0

As the table shows, the most used open source library is the Spring Framework: an application framework that got popular for simplifying the use of the Enterprise JavaBean model. Nowadays, the Spring Framework goes beyond application configuration and dependency injection. It is now a framework for building, running and managing applications, and is present throughout the entire lifecycle of an application. Spring's Apache 2.0 Licence enables it to be used for the development of both open and closed source software.

The second most used library is the JUnit framework for creating and running white-box developer tests (unit testing). The Common Public 1.0 licence under which JUnit is distributed also allows its use in commercial (closed source) applications, as long as JUnit is used without modifications. Although less permissive than the Apache 2.0 license, the Common Public 1.0 is still usable in a commercial setting.

Which OSS libraries are used most commonly in combination with each other?

Typically, certain libraries are used in combination. Table 2 lists the ten most commonly used combinations of libraries.

Table 2: Ten most commonly used combinations of open source libraries.

Combination	Systems
Spring Framework, Log4j	17%
Log4j, JUnit	17%
Spring Framework, Log4j, JUnit	14%
Javax Servlet, Log4j	14%
Spring Framework, Javax Servlet, Log4j	12%
Log4j, Commons Lang	10%
Hibernate, Log4j	10%
Commons Logging, Spring Framework, Log4j	10%
Spring Framework, Log4j, Commons Lang	10%
Hibernate, Spring Framework, Log4j	10%

Log4j, a logging utility for Java, is the most commonly used library in combination with other open source libraries, as it is present in all the top-ten combinations. The most common combination is with the Spring Framework, with JUnit, or with both. These three libraries each address very general software engineering issues (building and deployment, testing, and logging) and are frequently used together in the construction of commercial applications.

What is the economic value of using open source libraries in commercial software?

Whenever an open source library is used in a commercial application it saves the application developers the effort of implementing the extra functionality they get for free from the library. The efforts invested by the members of the open source community that developed the library thereby introduces savings in the projects that use the fruits of their labour. A quantification of the economic value of these savings can be measured

through the calculation of the rebuild value, the cost of implementing such components from scratch, of each library. Table 3 shows this quantification for the top-ten most commonly used libraries. The table lists the volume of these libraries in lines of code (LOC), the corresponding rebuild value in Euros (cost of rebuilding the library from scratch assuming average Java programmer productivity), the percentage of systems in which the library is used, and the estimated total savings induced by this usage for the 334 commercial systems in our study.

Table 3: Quantification of savings in commercial system development due to use of open source libraries.

Library	Volume (LOC)	Rebuild Value (€)	Systems Using	Saving (€)
Commons Collections	27,000	300,000	12%	12,000,000
Commons Lang	19,000	210,000	16%	11,000,000
Commons Logging	3,000	30,500	19%	2,000,000
EasyMock	4,000	45,400	15%	2,000,000
Hibernate	92,000	1,040,000	17%	59,000,000
Javax Servlet	2,000	25,000	22%	2,000,000
JUnit	6,000	70,000	42%	9,000,000
Log4j	25,000	276,000	39%	36,000,000
Slf4j	7,000	72,200	13%	3,000,000
Spring Framework	150,000	1,700,000	42%	239,000,000

As the table show, the Spring Framework not only represents the largest rebuild value among the top-ten most commonly used libraries, but also induces the largest savings. In fact, the Spring Framework induces more savings than the other nine libraries together. This is due to the fact that it is both the largest library and, at the same time, the most used one.

The library that induces the second-highest saving is Hibernate, a library for object-relational mapping between application code and database. Although Hibernate is only the sixth most commonly used library, it has the second largest rebuild value and this factor brings its savings to the second highest of this top ten list.

The total amount of savings induced by the top-ten most commonly used open source libraries for our set of 334 commercial software systems is € 375 million.

4 Discussion

In the debate around open source software, open and closed source are often portrayed as incompatible alternatives. Our study shows that, in fact, open source software is commonly used nowadays as a building block for commercial software systems. This usage is not incidental, but pervasive.

Our quantification of the economic value of the savings for commercial software development induced by the use of open source libraries is an estimation where many factors of uncertainty must be assumed. For example, we have used the entire rebuild value of each library as a factor in our calculation, while in many cases only part of the functionality of the library is used. Therefore, when not using the library, only part of its functionality would need to be replicated. On the other hand, we have only used the top-ten most commonly used libraries and disregarded the savings induced by the remaining ones.

The estimated savings of € 375 million for 334 commercial systems implies that on average the savings due to usage of open source libraries amounts to about € 1.1 million. From this estimate, one may safely conclude that the existence of open source libraries has thoroughly changed the economic reality of commercial software development and that, perhaps, the development efforts of the open source community have enabled commercial development to be faster, cheaper, or more ambitious.