Risks in the adoption of OpenOffice.org by governmental institutions

Miguel Ferreira
m.ferreira@sig.eu

Joost Visser
j.visser@sig.eu

Software Improvement Group

Abstract

SIG analyzed three versions of OpenOffice.org. The source code was measured and rated against the SIG Quality Model. Source code measurements show that although OpenOffice.org is over 4 million lines of code, it is still eligible for the SIG/TÜViT certification scheme. Furthermore, we identified risks that governmental institutions might be incurring when adopting OpenOffice.org due to the recent acquisition of Sun Microsystems by Oracle. Based on the identified risks we recommend that parties interested in adopting OpenOffice.org because of its open source nature, wait until the transaction to Oracle is complete, and Oracle has made a commitment in respect to its vision for OpenOffice.org's future.

Table of contents

1 Introduction

In order to determine technical quality of software, SIG [1] has developed the SIG Quality Model (SQM) [2], which is based on ISO 9126. SIG uses this model to rate software systems based on source code measurements, and to assess potential risks.
SIG analyzed OpenOffice.org (OOo) versions 2.4.1, 3.0.0 and 3.1.1 to assess their technical quality (Figure 1 shows an overview of the volume of each version). We have verified that the same level of quality was sustained over these three versions. Since we are interested in assessing risks in the adoption of OOo by governmental institutions, this document will briefly cover the evolution of OOo's source code, where the main focus is on the technical quality of version 3.1.1.

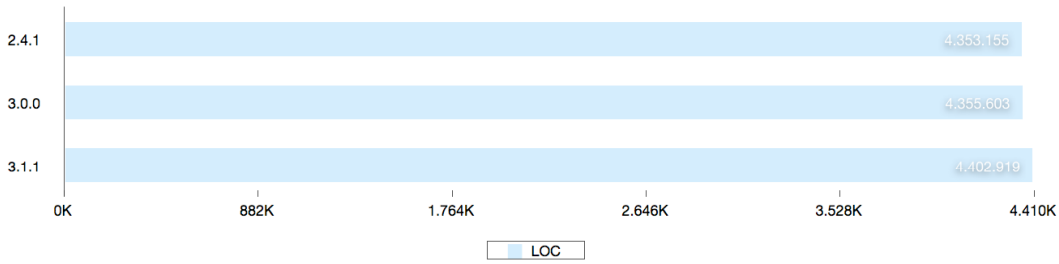

Figure 1 - Volume distribution for all analyzed versions

2 Background

OOo is an open source suite of office tools. OOo is currently the leading office suite in the open source domain. It was first released in October 2000, after Sun Microsystems bought StarDivision. StarDivion's commercial office suite, StarOffice, was then made open source. Since then, OOo is Sun Microsystems' main development project regarding office suites, whereas StarOffice is now based on OOo with a few additional, closed source, components [3].

Although Sun Microsystems leads the development of OOo by supplying most of the development capital, other major IT players have joined the project [4]. Most of the industrial partners that have joined Sun Microsystems in the development of OOo, see it as an opportunity for cutting costs of using commercial office suites. Currently the development team of OOo consists of developers from the following companies: Sun Microsystems, Novell Inc., IBM Corp., Intel Corporation, Debian, Red Hat Inc., among others.

Governments, schools and universities, from six different continents have adopted OOo as their preferred office suite [5]. Private companies throughout five continents have also followed in the adoption of OOo (e.g. Bangkok Airways in Asia, Peugeot Citroën in Europe, Novell in North America, MIP Holdings in Oceania, and Casas Bahia in South America).

The governance of the OOo project is delegated to the OOo Community Council. This council sets the project goals, handles conflicts of interest, gathers funds and provides a forum for discussion among the OOo community. According to the Community Council Charter [6], the council should be formed by 3 representatives of code contributers, 3 representatives of product development teams, 2 representatives of native languages projects, 1 community contributer, and 1 representative from Sun Microsystems. However, the actual council members follow a slightly different distribution [7].

The owner of OOo, Sun Microsystems, is in the process of being bought by Oracle. Although some steps have already been taken, the operation is not yet complete. Oracle has published [8] its plan for OOo, and it basically intents to keep developing and supporting OOo as an open source system. Still, because the acquisition operations is ongoing, Oracle has not yet committed to this vision.

3 SIG Quality Model

SQM is based on ISO 9126 and focused on maintainability. It rates software systems according to facts extracted from source code. It is a product of nearly 10 years of research and application of static analysis techniques to software technical quality evaluation. To rate the maintainability of a system SIG first measures six source code properties: volume, duplication, complexity, coupling, unit size, and unit interfacing.
These system properties are rated from one to five stars. Afterwards, they are aggregated, in a two step process, to maintainability sub-characteristics and finally to an overall maintainability rating (also from one to five stars).

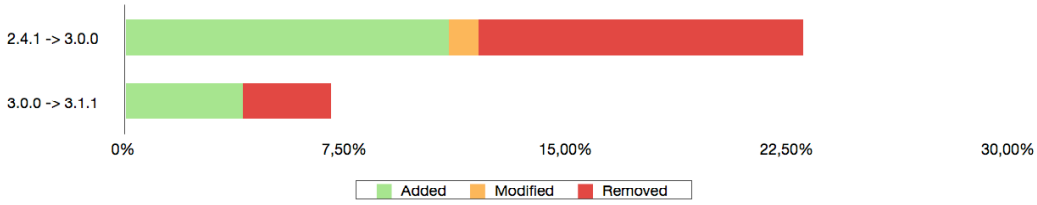

Figure 2 - Distribution of added, modified and removed code between versions

4 OpenOffice.org Overall Quality

According to SQM, OOo sustained its Maintainability rating (3 stars) throughout the three analyzed versions, despite the fact that roughly 23% of the code base was changed (11% added, 1% modified, and 11% removed) from 2.4.1 to 3.0.0; and 7% was changed (4% added, and 3% removed) from 3.0.0 to 3.1.1. (See Figure 2 for change distribution.)The sustained maintainability rating leads us to conclude that OOo's development and management processes are successful in controlling software quality.

Aside from the volume rating (0.5 stars) the metrics that score worst according to SQM are complexity (1.5 stars) and unit size (1.5 stars). Since OOo is intended to compete in the segment of office suites a large volume is unavoidable. Otherwise it would not be a fully integrated suite offering all the functionally users expect. Furthermore, according to SQM, OOo could even double its size and still reach a 5 stars rating, provided the other system property ratings are substantially improved.

The impact of new highly maintainable code in a system of the size of OOo is evermore reduced. So, in order to make OOo more maintainable it would be necessary to refactor the existing code, which would result in higher technical quality.

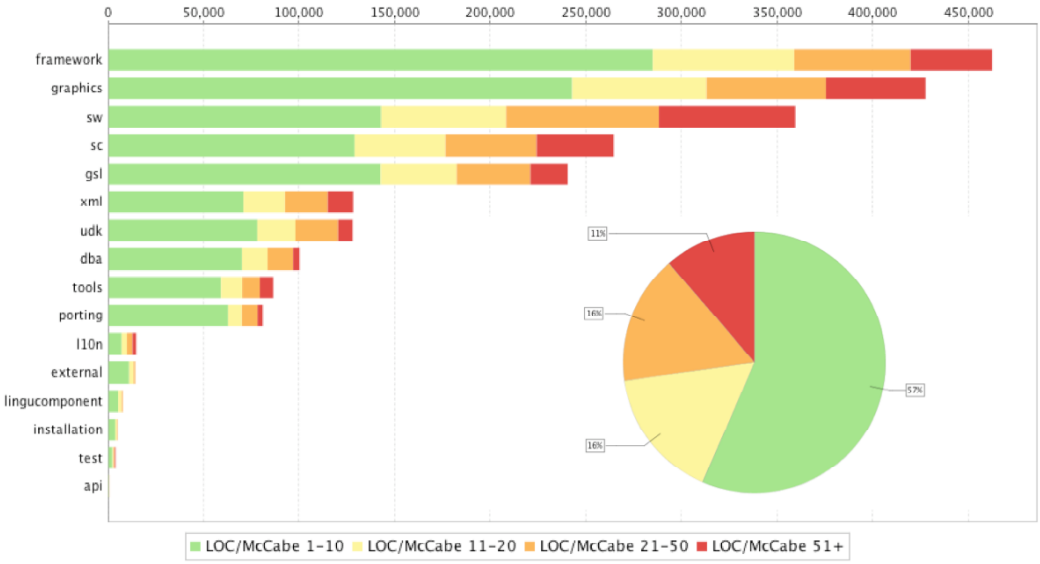

Figure 3 - Complexity distribution in C++ code for version 3.1.1

5 Quality Assurance

Quality Assurance (QA) within OOo is an independent project, aiming at raising and maintaining the quality of the contributions made to the code base, as well as the overall quality of each public release of OOo [9]. QA has established C++ coding standards to guide the developers, and a code review processes to verify the adherence of code to those standards [11].

The coding standards that are in place at the moment are mainly focused on programming best practices and guidelines for uniformity among modules and classes. The verification of the code's adherence to these standards is done manually during code review sessions.

6 Risks

SIG sees the risk of fading development effort by the main contributer, Sun Microsystems, after the recent acquisition by Oracle is complete. Fading development effort will mean slow, or no, evolution of new features into the software, and lack of support. It is fair to ask: *"If I have a working version of OOo, why do I need support?"* Well, both operating systems and application platforms evolve reasonably fast. This usually requires adaptations on the application side, either to drop deprecated functionality or take advantage of new additions. Sometimes these adaptations go even deeper if the technology changes drastically (ex: the update from 32-bit to 64-bit architectures).

With this in mind, consider the adoption of OOo by your institution, and the associated costs. People have to adapt to OOo, in some rare cases even receive specific training. Although all the existing documents and templates can be automatically converted to new formats, they still require manual review. If after all this OOo becomes a typical commercial product, it is reasonable to assume that there will be no more free support and development. However, it is also reasonable to think that if the source code of OOo is available now, then anyone can support and further develop it. Here is where the management, quality assurance and source code maintainability issues come in:

1. Since there is a significant portion of highly complex code, its understandability is therefore reduced. This means that in the case the current developers leave the project, others will most certainly have great difficulties in understanding the code that is left behind, let alone adapt it;
2. Any third party interested in maintaining OOo would need to invest a great deal to be able to succeed. On one hand, assuming that OOo is in version 2.4.1 and this third party will take it to version 3.0.0. According to Figure 2, this means changing 23% of the code base. Based on the properties of the code and its overall technical quality, according to industry average productivity an effort of 100 man year, with a cost around the 10 million euros. On the other hand, in a scenario of pure maintainability (15% change) the third party would need to employ and effort of 70 man year, costing around 7 million euro.
3. There is a vast infrastructure for development and QA already in place in OOo. If the management of the project leaves, it will be hard for others to take control over such infrastructure.
4. If development and management teams are broken apart, it is reasonable to assume that the individual contributers that want to keep contributing will, be, by themselves, less productive. Also, if the quality assurance infrastructure, which is currently implemented by Sun Microsystems employees, ceases to perform its function, then it is foreseeable that the technical quality starts to deteriorate. Thus, the required effort to maintain and develop increases even more.

These four points show that maintaining OOo is not straightforward, even though it is open source.

7 Recommendations

OOo is a very large system of average quality. It is able to actively compete with long established proprietary solutions in the same segment. Still, there is plenty room for improvement.

With respect to the technical quality of OOo, SIG recommends the OOo Community Council to:

- add more coding standards specifically targeting the technical quality of the code;
- complement the manual code reviews with automated verification of adherence to coding standards.

Regarding the adoption of OOo, because of its open source nature, by governmental institutions, SIG recommends that interested parties wait for the conclusion of the ongoing process of acquisition of Sun Microsystems by Oracle. Even though Oracle has stated its intention to continue the development of OOo, it has also opened the door for a future introduction of *"a typical commercial license option"* in the product [8].

8 References

1. “sig.eu”. Software Improvement Group. Retrieved 2009-12-17. http://www.sig.eu

2. Ilja Heitlager, Tobias Kuipers, Joost Visser: A Practical Model for Measuring Maintainability. QUATIC 2007: 30-39

3. “OpenOffice.org - Wikipedia, the free encyclopedia”. Wikimedia Foundation, Inc. Retrieved 2009-12-17. http://en.wikipedia.org/wiki/Openoffice

4. “Domain Developer - OpenOffice.org Wiki”. OpenOffice.org Wiki. Retrieved 2009-12-17. http://wiki.services.openoffice.org/wiki/DomainDeveloper

5. “Major OpenOffice.org Deployments - OpenOffice.org Wiki”. OpenOffice.org Wiki. Retrieved 2009-12-17. http://wiki.services.openoffice.org/wiki/Major_OpenOffice.org_Deployments

6. “council: Community Council Charter”. OpenOffice.org. Retrieved 2009-12-17. http://council.openoffice.org/councilcharter12.html

7. “council: OpenOffice.org Community Council”. OpenOffice.org. Retrieved 2009-12-17. http://council.openoffice.org/

8. “Oracle and Sun Overview and FAQ”. Oracle. Retrieved 2009-12-17. http://www.oracle.com/ocom/groups/public/documents/webcontent/038563.pdf

9. “Quality Assurance - OpenOffice.org Wiki”. OpenOffice.org Wiki. Retrieved 2009-12-17. http://wiki.services.openoffice.org/wiki/QA

10. “C4 Software Technology Reference Guide: A Prototype”. Carnegie Mellon University / Software Engineering Institute. CMU/SEI-97-HB-0011997. http://www.sei.cmu.edu/library/abstracts/reports/97hb001.cfm

11. “Cpp Coding Standards - OpenOffice Wiki”. OpenOffice.org Wiki. Retrieved 2009-12-17. http://wiki.services.openoffice.org/wiki/Cpp_Coding_Standards