Long-Term Sustainability of Open Source Software Communities beyond a Fork: A Case Study of LibreOffice

Jonas Gamalielsson and Björn Lundell

University of Skövde, Skövde, Sweden
{jonas.gamalielsson,bjorn.lundell}@his.se

Abstract. Many organisations have requirements for long-term sustainable software systems and associated communities. In this paper we consider long-term sustainability of Open Source software communities in Open Source projects involving a fork. There is currently a lack of studies in the literature that address how specific Open Source software communities are affected by a fork. We report from a case study aiming to investigate the developer community around the LibreOffice project, which is a fork from the OpenOffice.org project. The results strongly suggest a long-term sustainable community and that there are no signs of stagnation in the project 15 months after the fork. Our analysis provides details on the LibreOffice developer community and how it has evolved from the OpenOffice.org community with respect to project activity, long-term involvement of committers, and organisational influence over time. The findings from our analysis of the LibreOffice project make an important contribution towards a deeper understanding of challenges regarding long-term sustainability of Open Source software communities.

1 Introduction

Many organisations have requirements for long-term sustainable software systems and associated digital assets. Open Source software (OSS) has been identified as a strategy for implementing long-term sustainable software systems (Blondelle et al., 2012; Lundell et al., 2011; Müller, 2008). For any OSS project, the sustainability of its communities is fundamental to its long-term success. In this paper, we consider long-term sustainability of OSS communities in OSS projects involving a fork. In so doing, we undertake an investigation of how the LibreOffice (LO) project community has evolved from the OpenOffice.org (OO) project community with respect to project activity, long-term involvement of committers, and organisational influence over time.

Many companies need to preserve their systems and associated digital assets for more than 30 years (Lundell et al., 2011), and in some industrial sectors (such as avionics) even more than 70 years (Robert, 2006, 2007). In such usage scenarios "there will be problems if the commercial vendor of adopted proprietary software leaves the market" with increased risks for long-term availability of both software and digital

I. Hammouda et al. (Eds.): OSS 2012, IFIP AICT 378, pp. 29–47, 2012.

assets (Lundell et al., 2011). Similarly, for organisations in the public sector, many systems and digital assets need to be maintained for several decades, which cause organisations to vary concerning different types of lock-in and inability to provide long-term maintenance of critical systems and digital assets (Lundell, 2011). For this reason, sustainability of OSS communities has been identified as essential for long-term sustainability of OSS.

There are many different aspects of an OSS project that can affect community sustainability. Good project management practice includes to consider different incentives for contributing to OSS communities, which in turn may affect the future sustainability of communities (Bonaccorsi and Rossi, 2006). Earlier research has also suggested that an effective structure of governance is a basis for healthy and sustainable OSS communities (de Laat, 2007), and also that a community manager plays a key role for achieving this (Michlmayr, 2009). Further, the licensing of OSS may affect the community and it has been claimed that "fair licensing of all contributions adds a strong sense of confidence to the security of the community" (Bacon, 2009). It has also been claimed that the choice of an OSS license "can positively or negatively influence the growth of your community." (Engelfriet, 2010)

The issue of forking OSS projects has been an ongoing issue of debate amongst practitioners and researchers. It has been claimed that "Indeed, the cardinal sin of OSS, that of project forking (whereby a project is divided in two or more streams, each evolving the product in a different direction), is a strong community norm that acts against developer turnover on projects" (Ågerfalk et al., 2008), and it has been noted that few forks are successful (Ven and Maennert, 2008). It is therefore, perhaps, not surprising that it has been claimed that "there must be a strong reason for developers to consider switching to a competing project" (Wheeler, 2007). However, it has also been claimed that "forking has the capability of serving as an invisible hand of sustainability that helps open source projects to survive extreme events such as commercial acquisitions" (Nyman et al., 2011). Clearly, there is a need for increased knowledge about how OSS communities are affected by a fork.

The overarching goal of this study is to investigate how long-term sustainability of OSS communities is affected by a fork . We investigate this in the context of the LO project. The paper makes three principle contributions. Firstly, we establish a characterisation of project activity for the LO developer community before and after the fork from OO. In so doing, we specifically focus on contributions to the Software Configuration Management System (SCM). Secondly, we present findings regarding long-term involvement of contributors in the LO project. Thirdly, we present findings regarding the organisational influence in the LO project over time. Further, besides providing these three principle contributions, we also contribute approaches and metrics for analysing long-term sustainability of OSS communities (with or without forks) in OSS projects, and illustrate their use on the LO project.

There are a number of reasons which motivate a case study on the LO project. Firstly, LO is one of few OSS projects which have had an active community for more than 10 years (including the development in OO), with significant commercial interest. Secondly, there has been tensions within the OO project which finally led to the creation of the Document Foundation and the LO project (Byfield, 2010; documentfoundation.org, 2012). Thirdly, the project has reached a certain quality in that it has been adopted for professional use in a variety of private and public sector organisations (Lundell, 2011; Lundell and Gamalielsson, 2011). Therefore, its community is likely to attract a certain level of attention from organisations and individuals. Fourthly, previous studies of the parent project OO (Ven et al., 2007) and more recent studies of LO (Gamalielsson and Lundell, 2011) show that there is widespread deployment in many organisations in a number of countries. This in turn impose significant challenges from a geographically distributed user community. Further, previous results (Gamalielsson and Lundell, 2011) and anecdotal evidence from an official spokesperson for the LO project (Nouws, 2011) suggest significant activity in the LO community, which motivates more in-depth studies of the project.

In this paper we position our exploration of sustainability of OSS communities in the broader context of previous research on OSS communities (section 2). We then clarify our research approach (section 3), and report on our results (section 4). Thereafter, we analyse our results (section 5) followed by conclusions and discussion (section 6).

2 Background

In the context of OSS projects, it has been shown that "little research has been conducted on social processes related to conflict management and team maintenance" and that there are several open questions related to this, such as "How is team maintenance created and sustained over time?" (Crowston et al., 2012). Further, we note that there is a lack of reported insights on specific projects, and in particular a lack of research on OSS community involvement in projects involving a fork. One notable exception is a study on motivations for forking SourceForge.net hosted OSS projects (Nyman and Mikkonen, 2011). However, this study did not focus on community involvement over time.

Studies on the evolution of OSS projects over time do not always have a community focus and are not always targeted at specific projects. Examples include a study on the total growth rate of OSS projects (Deshpande and Riehle, 2008), and work on the evolution of social interactions for a large number of projects on SourceForge.net over time (Madey et al., 2004). Another example is a study on survival analysis of OSS projects involving the application of different metrics based on the duration of thousands of projects in the FLOSSMETRICS database (Samoladas et al., 2010). There are also studies which focus on the evolution of software over time for specific OSS projects but which do not consider the community aspect. An example is a study on the Linux kernel based on Lehman's laws of software evolution, which involved the application of code oriented metrics over time (Israeli and Feitelson, 2010). A similar approach was used in a case study on the evolution of Eclipse (Mens et al., 2008).

There are other studies that do have a focus on the evolution of communities for specific OSS projects, but do not address the effects of a fork. For example, case studies have been conducted on the Debian project involving quantitative investigations of evolution of maintainership and volunteer contributions over time (Robles et al., 2005; Michlmayr et al., 2007). Another study involved an investigation of developer community interaction over time for Apache webserver, GNOME and KDE using social network analysis (Lopez-Fernandez et al., 2006). A similar study involved the projects Evolution and Mono (Martinez-Romo et al., 2008). Case studies on the Nagios project (Gamalielsson et al., 2010), and TopCased & Papyrus (Gamalielsson et al., 2011) addressed community sustainability and evolution over time with a special focus on organisational influence. Other research partially focusing on community evolution are early case studies on the Linux kernel (Moon and Sproull, 2000), GNOME (German, 2003), Apache webserver (Mockus et al., 2002), and Mozilla (Mockus et al., 2002).

3 Research Approach

We undertook a case study of the LO project as our approach for investigating how long-term sustainability of OSS communities is affected by a fork.

First, to establish a characterisation of project activity for the LO developer community we undertook an analysis of release history, commits to the SCM and contributing committers over time. Second, to investigate long-term involvement of contributors we used different metrics that consider how long period of time committers have been active, recruitment of new committers over time, and to what extent committers contribute before and after the fork from OO. Third, to investigate organisational influence in the LO project over time we undertook an analysis of the use of different email affiliations in the SCM over time and in particular studied the difference in influence in connection with the fork from OO.

To investigate the sustainability of OSS communities, we adopt and extend approaches from earlier studies (Gamalielsson et al., 2011; Gamalielsson and Lundell, 2011) in order to analyse the contributions in terms of committed SCM artefacts of the OSS projects over time. The approaches used in this study for analysis of long-term involvement are the principal extensions to the approaches used in the earlier studies. The data for the LO project was collected from the LO website[1], where all listed Git subrepositories were used in the analysis. Git logs were extracted for the repositories and thereafter analysed using custom made scripts. More specifically, the date and committer email address for each commit was extracted and stored for subsequent analysis over time. The affiliation of a committer at the time of a commit was established by using the domain of the email address of the commit. We also used additional information regarding the affiliation of contributors[2] to further analyse the results on organisational influence. Further, a semi-automated approach involving manual inspection was used to associate email address aliases with the same actual committer.

4 Results

4.1 Project Activity

The combined version history of LO and OO is shown in table 1. It can be observed that there has been a continuous flow of new releases for more than 10 years. On 25

[1] `http://www.libreoffice.org/developers-2/`, accessed 3 March 2012

[2] `http://cgit.freedesktop.org/libreoffice/contrib/gitdm-config/`, accessed 3 March 2012

January 2011 the Document Foundation released its first stable version of LO, which constitutes a fork from the OO project (documentfoundation.org, 2012). Further, the version history is divided into 14 project intervals, where most intervals span between second-level releases. The start date of an interval is the date of the associated release, and the end date is the day before the next release in the table (except for the last interval where the end date is 2011-12-31). The intervals are used in the analysis of long-term involvement of contributors in section 4.2. Working towards major and second level releases demands a significant effort, and we therefore found it appropriate to report on long-term involvement at this level of abstraction.

Table 1. The combined version history of OpenOffice.org (OO) and LibreOffice (LO)

Releases	Date (YYYY-MM-DD)	Interval
OO Initial release	2001-10-01	I1
OO 1.0	2002-05-01	I2
OO 1.1	2003-09-02	I3
OO 2.0	2005-10-20	I4
OO 2.1	2006-12-12	I5
OO 2.2	2007-03-28	I6
OO 2.3	2007-09-17	I7
OO 2.4	2008-03-28	I8
OO 3.0	2008-10-13	I9
OO 3.1	2009-05-07	I10
OO 3.2	2010-02-11	I11
LO 3.3 B1	2010-09-28	I12
OO 3.3, LO 3.3	2011-01-25	I13
LO 3.4.0	2011-06-03	I14

The developer activity in LO is presented in Figure 1, which shows the number of commits for each month from September 2000 to December 2011. Our SCM analysis of the LO project includes the development in OO before the fork on 28 September 2010 (indicated with a vertical red line in Figure 1). We note that activity in the LO project varies, with distinct peaks in connection with the OO 2.0 (September 2005) and OO 2.4 (March 2008) releases (each peak with more than 50000 commits, which are not shown in the diagram for scaling reasons). Since October 2008 (with the release of OO 3.0) there have been 2700 commits each month on average.

Figure 2 illustrates the number of active committers during each month of the LO project. It can be observed that there is a large number of committers active early in the project, and that the activity decreases considerably shortly after the release of the first stable version of the software (OO 1.0) in May 2002. Further, the number of committers increases to a higher level after the release of OO 3.1 in May 2009. It can also be noticed that there is a significant peak in October 2010 in connection with the fork from OO (see vertical red line in Figure 2). In total, 665 unique committers (distributed over 1009 committer identifiers) have contributed to the LibreOffice Git repository from September 2000 until December 2011.

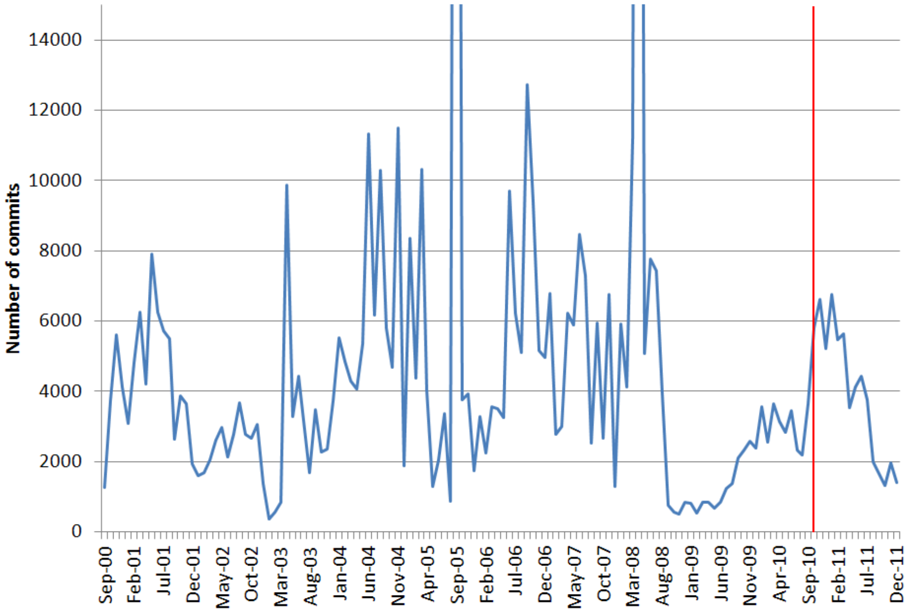

Fig. 1. Number of monthly commits for the LibreOffice project

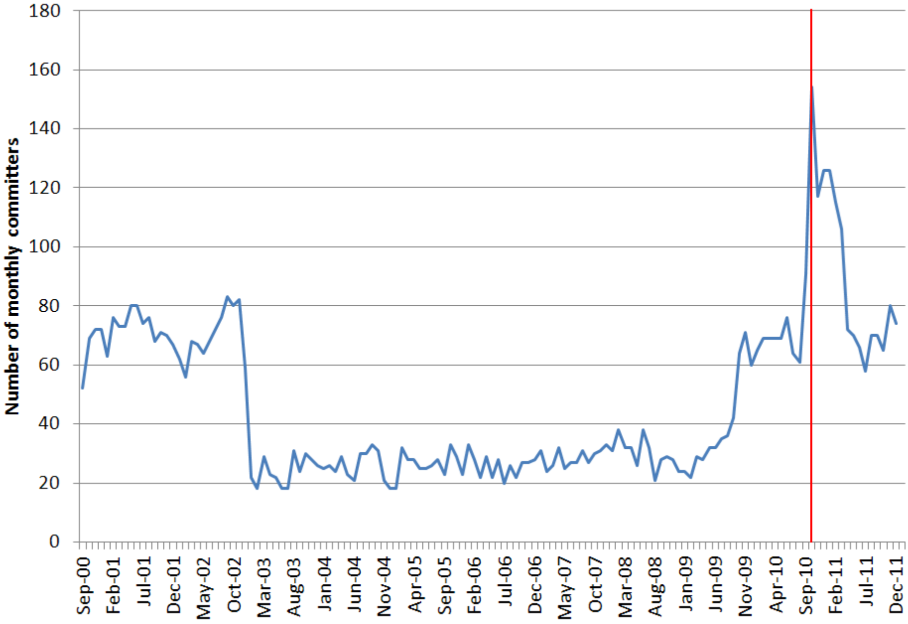

Fig. 2. Number of monthly committers for the LibreOffice project

4.2 Long-Term Involvement

Figure 3 provides an overall impression of the endurance and total activity of LO committers over time. The elapsed number of project intervals between the first and the most recent commit for each committer is shown. The committers are sorted in descending order from the bottom and upwards based on elapsed number of project intervals from the interval for the first commit until the interval for the most recent

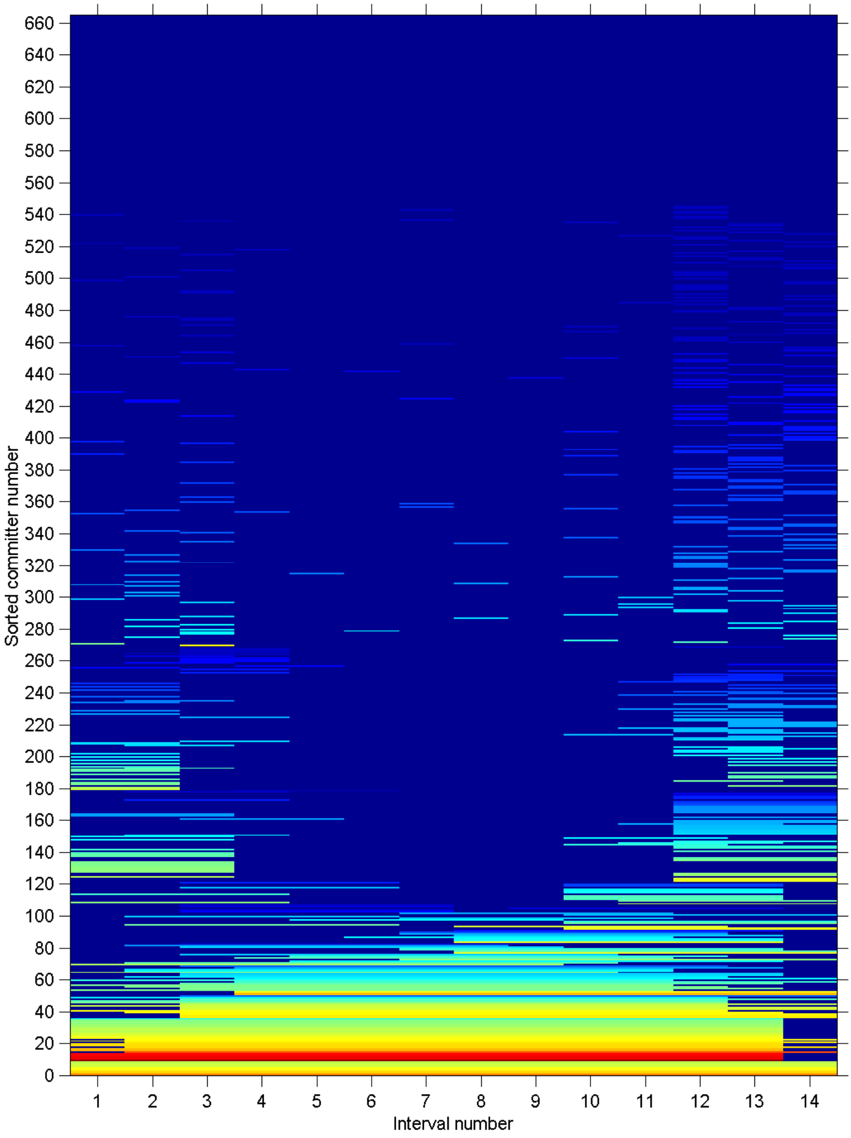

Fig. 3. Endurance and total activity of LibreOffice committers over time

commit. Committers with the same elapsed number of project intervals are secondarily sorted in descending order based on their total activity over all project intervals. For each combination of committer (along the Y-axis) and project interval (along the X-axis), the colour represents the total activity over all project months (dark blue represents low activity, whereas dark red represents high activity using a rainbow colour scale). The figure includes all committers who only provided a single commit, and for those the elapsed time is presented as one interval.

Figure 4 is based on the data visualised in Figure 3 and illustrates the elapsed number of project intervals between the interval for the first commit and the interval for the most recent commit as a function of proportion of committers. For example, the graph shows that 5% of the committers contribute over a period of at least 13 intervals. Further, 27% of the committers have contributed over a period of at least three intervals. Nine committers (1,4%) have committed over the longest observed period of 14 intervals. It can also be noted that 59% of the committers have contributed (one or several commits) during one single interval.

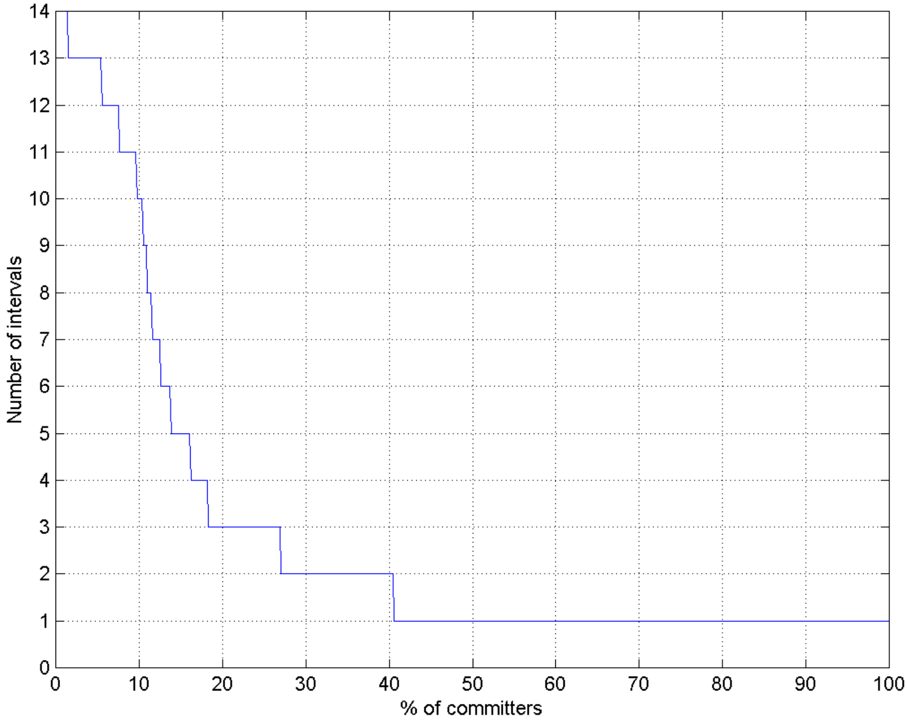

Fig. 4. Number of commit intervals from first to last commit as a function of proportion of committers for the LibreOffice project

Figure 5 shows the accumulated number of committers as a function of project interval. An observation is that 96 initial committers contribute during the first interval. Further, the growth rate in terms of new committers has varied during the project. There was for example a fast growth rate initially (during intervals 1-3), and also at the time of the fork and onwards (from interval 12). The other intervals are characterised by a slower growth in number of new committers.

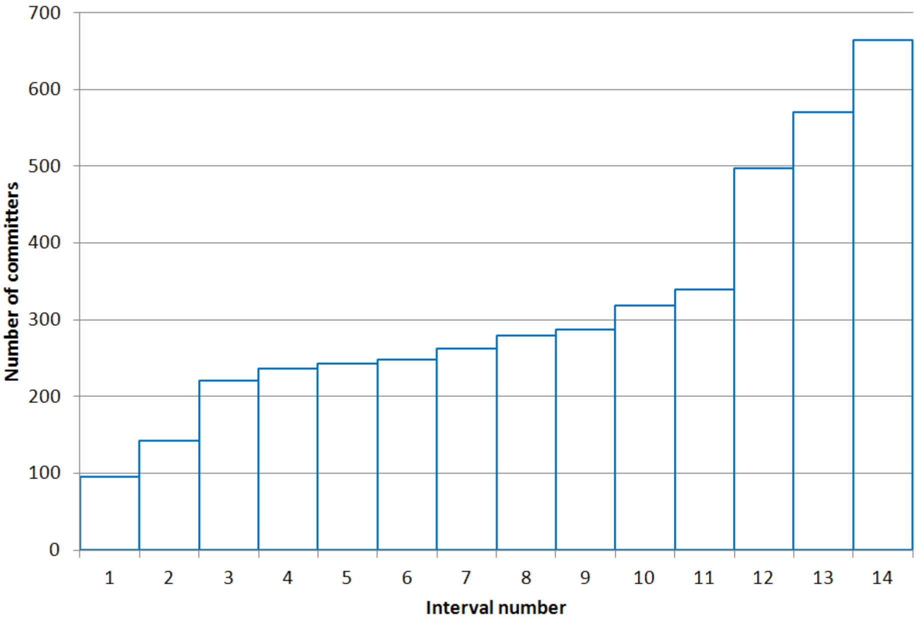

Fig. 5. Accumulated number of committers as a function of interval number

Figure 6 illustrates the involvement of committers over time in terms of number of active intervals. The left graph shows the number of active intervals as a function of proportion of committers only active before the fork (red trace), and length of the longest sequence of consecutive active intervals as a function of proportion of committers only active before the fork (blue trace). It can for example be observed that 38% of the committers contribute during at least two intervals and that 35% contribute during at least two consecutive intervals. Further, it can be noted that no committer is active during all 11 intervals before the fork. Similarly, the right graph illustrates the involvement of committers only active after the fork (using the same trace legend). For example, 23% of the committers contribute during at least two intervals and 20% contribute during at least two consecutive intervals.

Figure 7 is similar to Figure 6, and the left graph shows the number of active intervals as a function of proportion of committers active both before and after the fork (red trace), and length of the longest sequence of consecutive active intervals extending

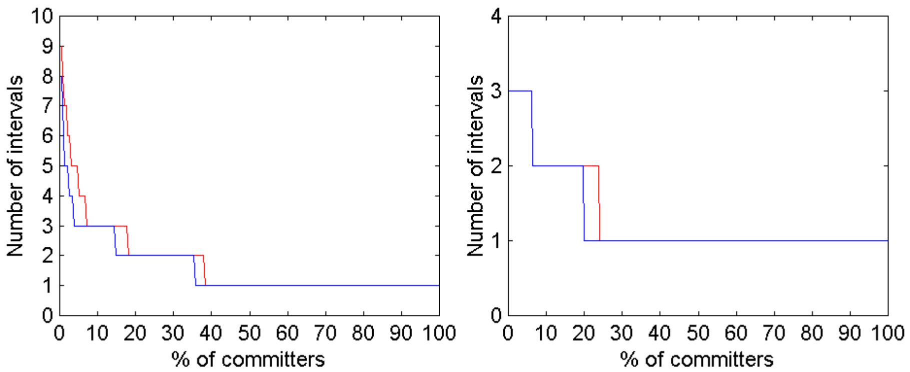

Fig. 6. Involvement of committers in terms of number of active intervals (left: committers active only before the fork, right: committers active only after the fork)

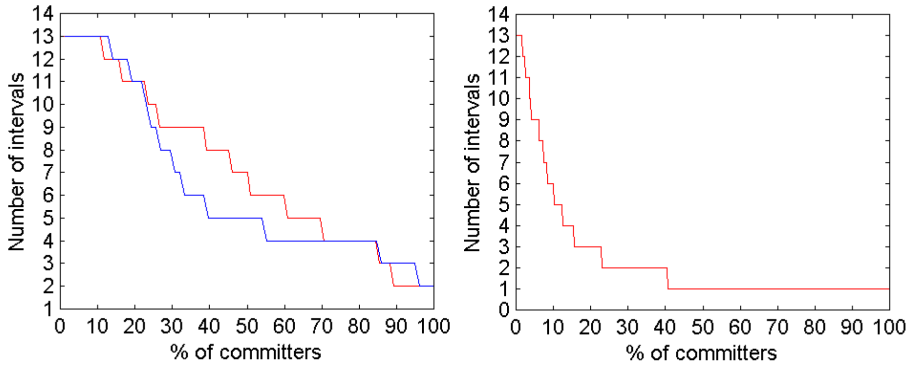

Fig. 7. Involvement of committers in terms of number of active intervals (left: committers active both before and after the fork, right: all committers)

upwards and downwards from the interval of the fork (blue trace). It can for example be observed that 38% of the committers contribute during at least nine intervals and that 38% contribute during at least six consecutive intervals The right graph illustrates the number of active intervals as a function of proportion of all committers. For example, 22% of all committers contribute during at least three intervals. One important observation from Figures 6 and 7 is that committers active both before and after the fork contribute during more intervals and during longer consecutive periods of intervals compared to committers who are only active either before or after the fork. This is a clear indication of endurance for contributors that are committed to the LO branch.

Figure 8 illustrates the proportion of commits as a function of proportion of committers. Specifically, the left graph shows the proportion of all commits during the project as a function of proportion of all committers (black trace), committers who contribute only before the fork (red trace), committers who contribute only after the fork (green trace), and committers who contribute both before and after the fork (blue trace). The total number of commits contributed by all 665 committers during the project is 587026, where 552620 (94%) of these commits are contributed by 102 committers (15% of all committers) that are active both before and after the fork. 20321 (3,5%) of the commits are contributed by 238 committers (36% of all committers) active only before the fork, and 14085 (2,5%) of the commits are contributed by 325 committers (49% of all committers) active only after the fork. From the graph it can for example be observed that 10% of all committers contribute 95% of all commits, and that 10% of committers that are active both before and after the fork contribute 80% of all commits. The same proportion of committers only contributing either before or after the fork contribute 2,4% and 1,9% of all commits, respectively. One important observation in the left graph of Figure 8 is that committers that are active both before and after the fork contribute the majority of the commits, which indicates that the most influential committers are committed to the LO branch. Further, committers only active either before or after the fork contribute a small proportion of the commits. One possible explanation for the limited influence of new committers since the fork is that only 15 months have passed after the fork. It should also be mentioned that a large proportion of all committers contribute few commits (18% only make a single commit, and 44% contribute 5 commits or less).

The right graph in Figure 8 is similar to the left graph, but shows the proportion of subgroup commits during the project as a function of proportion of committers. The subgroups are all committers (black trace), committers only contributing before the fork (red trace), committers only contributing after the fork (green trace), and committers contributing both before and after the fork (blue trace). It can for example be observed that for the set of commits for each committer subgroup, 10% of all committers contribute 95% of the commits (same as in left graph), and that 10% of committers that are active both before and after the fork contribute 85% of the commits. The same proportion of committers only contributing either before or after the fork contribute 69% and 80% of the subgroup commits, respectively. Hence, a relatively small proportion of committers contribute a major proportion of the commits within each committer subgroup.

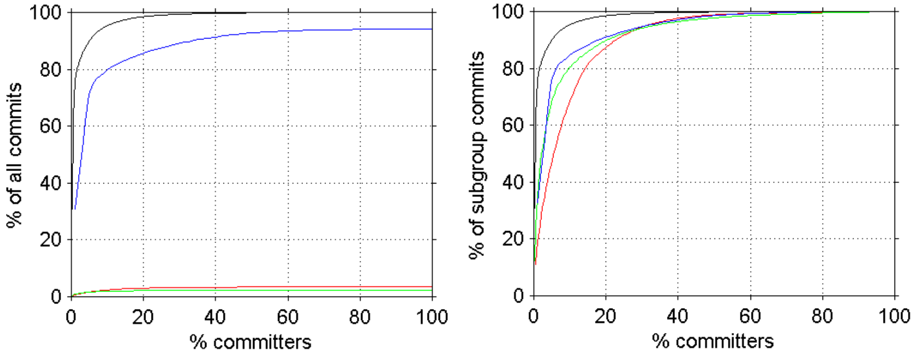

Fig. 8. Proportion of commits as a function of proportion of committers (left: for all commits, right: for commits in different committer subgroups)

4.3 Organisational Influence

The proportion of commits for the 10 most active affiliations over the time (from January 2007 to December 2011) in the LO project is shown in Figure 9 (like in Figure 1, the peak in April 2008 is not shown for scaling reasons). It can be observed that "openoffice" is dominating until August 2010, and that other affiliations break the dominance from September 2010 (the month of the fork) and onwards. It is also noted that "sun" is most active in the period from October 2009 to July 2010, and that "oracle" is most active from August 2010 to March 2011. Further observations are that "novell" and "suse" have been active for the entire four year period with an increased activity after the fork, and that "redhat" has become the major contributor ever since the fork.

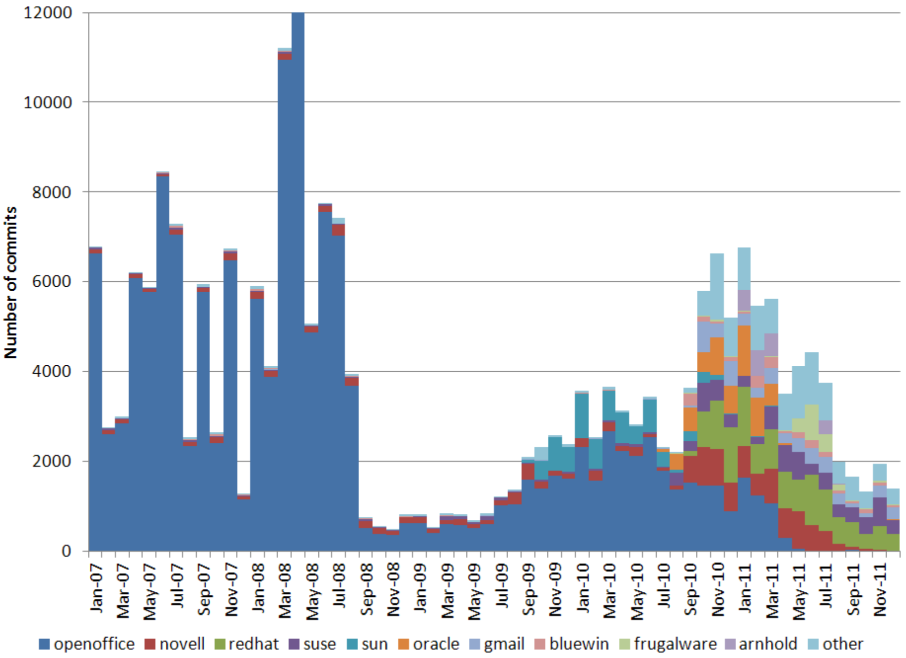

Fig. 9. Proportion of commits per affiliation over time for the LibreOffice project

Figure 10 illustrates the total affiliation commit influence for LO 15 months before and after the fork on 28 September 2010, and further emphasises the shift from "openoffice" domination to a more diversified developer community after the fork. This is especially evidenced by the increased proportion of "other" affiliations 15 months after the fork. The additional information[3] regarding the use of the

[3] http://cgit.freedesktop.org/libreoffice/contrib/gitdm-config/, accessed 3 March 2012

"openoffice" affiliation revealed that this affiliation is clearly dominantly used by committers employed by either Oracle or Sun. Further, we found that there were 148 different committers with 52 different affiliations contributing during the time period 15 months before the fork, whereas there were 424 different committers with 194 different affiliations contributing during the time period 15 months after the fork. In fact, there have been 377 different committers with 116 affiliations contributing from the start (September 2000) until the fork, which is less than the number of committers and affiliations contributing during 15 months after the fork. This together further strengthens the impression of a more diversified developer community after the fork.

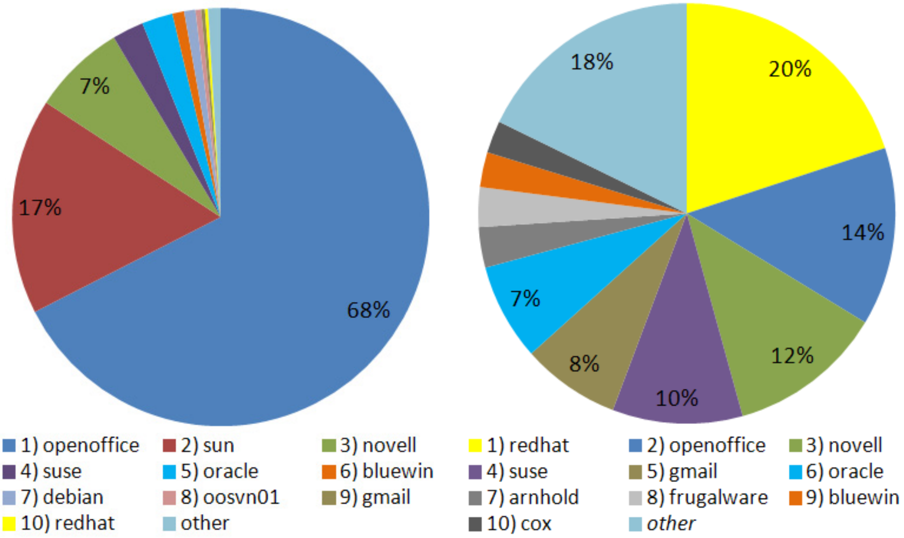

Fig. 10. Total affiliation commit influence (left pie: during 15 months before the fork, right pie: during 15 months after the fork) for the LibreOffice project

5 Analysis

From our results we make a number of observations related to our results on project activity. Firstly, there have been regular and frequent releases of stable versions of the software for a time period of more than ten years. Other examples of well known OSS projects with release histories extending over many years are Apache webserver[4] and the Linux kernel[5], which have had frequent releases since 1995 and 1991, respectively. We note that, as for the LO project, both these projects are governed by a

[4] http://httpd.apache.org/

[5] http://www.kernel.org/

foundation[6]. Secondly, despite a relatively limited time period (15 months) after the LO fork from OO, the transition and formation of the LO community seems to be successful. However, we acknowledge the short time period after the fork and that our early indications of a successful LO community after transition from OO need to be confirmed by an analysis over a longer time period at a later stage. As a comparison, a well-known fork with significant uptake and a long-term sustainable community is OpenBSD[7], which was forked from NetBSD in 1995 and still has an active developer community (Gmane.org, 2012). Thirdly, there has been substantial activity for more than ten years. Despite some variation between stable releases, our findings suggest a long-term trend towards a sustainable community as we have not observed any signs of a lasting decline in the community activity. As a comparison, there has been stable OSS community activity over many years in the aforementioned Apache webserver and Linux kernel projects.

Based on our results on long-term involvement we observe that a large proportion of the most influential committers have been involved for long periods of time both before and after the fork from OO, which indicates that the developer community has a strong commitment with the LO branch. A strong commitment of contributors over long periods of time has been observed earlier in a study on the Debian project where it was found that maintainers "tend to commit to the project for long periods of time" and that "the mean life of volunteers in the project is probably larger than in many software companies, which would have a clear impact on the maintenance of the software" (Robles et al., 2005).

Our results on organisational influence show that a number of committers representing different organisations provide substantial amounts of contributions to the project over a number of years. This may be considered a risk for the project in the long term unless new contributors join the project on a regular basis. However, this does not seem to be a significant risk for this project since our findings suggest that the community has successfully recruited new committers over time. Diversified developer communities with respect to affiliation have also been observed in earlier case studies on the Linux kernel project (Aaltonen and Jokinen, 2007), and on the Topcased and Papyrus projects (Gamalielsson et al., 2011).

Further, when considering OSS products in long-term maintenance scenarios for potential adoption, it is critical to understand and engage in communities related to the OSS project. For the base project analysed (OO), a governance structure has been established and the OO community is governed by its community council (openoffice.org, 2012). Similarly, the investigated branch after the fork (LO) has also established a governance structure referred to as the Document Foundation (documentfoundation.org, 2012). Despite such explicitly documented governance structures, project participants may decide to fork a project, which happened when the Document Foundation established the LO project as a fork from OO on 28 September 2010. In acknowledging a short time period after the fork (15 months), our results suggest that this fork may actually be successful. We note that our observation indicates that the

[6] The Apache Software Foundation (`http://www.apache.org/`) and the Linux Foundation (`http://www.linuxfoundation.org/`)

[7] `http://www.openbsd.org/`

LO project may be an exception to the norm since previous research claims that there has been "few successful forks in the past" (Ven and Mannaert, 2008).

The importance of engagement in OSS communities has been amplified by Shaikh and Cornford (2010) "Large companies understand that if they want to preserve a long term relationship with open source communities and harness the expertise and products they offer, then they must loosen up and relax, avoiding to much concern about their level of control.". Further, for the longevity of an OSS community, it is essential to be able to attract active contributors long-term. Our study shows that there is no single commercial company dominating the contributions to the LO project by the most influential contributors to the SCM. In fact, the LO community involves a large number of contributors from many different organisations, and the fork was not initiated by the organisation behind the base project. This is in contrast with a fork initiated by a company that controls the code base, something which occurred when Red Hat "forked their own codebase into Red Hat Enterprise Linux and Fedora Core Linux" (Gary et al., 2009).

There is a complex inter-relationship between community and company values which impacts on opportunities for long-term maintenance and support for OSS projects. In fact, for the investigated base project (OO) in this study, concerns for the long-term sustainability of its community have been raised (Noyes, 2010). In previous research, it has been claimed that companies "have valid concerns about the survival of and continued support for F/OSS products. The traditional telephone hotline and maintenance contract offer a comfort factor that a voluntary bulletin board—which is the main support for many F/OSS products—cannot provide." (Fitzgerald, 2004) On the contrary, practitioners in the embedded systems area experience that support from large OSS communities is "considered superior, compared to proprietary alternatives in some cases" (Lundell et al., 2011). Further, for the investigated OSS project, there are a number of companies offering support on a commercial basis, something which should be considered before potential adoption in addition to engagement in voluntary OSS communities.

To successfully master the art of establishing a long-term sustainable volunteer community is a huge challenge. As in all organisations, there are "times in every community when repetition, housekeeping, and conflict play a role in an otherwise enjoyable merry-go-round. When the community begins to see more bureaucracy and repetition than useful and enjoyable contributions, something is wrong." (Bacon, 2009) From our investigation in this study, our results indicate that the LO project seems to be successful in keeping old and recruiting new committers that contribute to the SCM over time. For software systems with long life-cycles, the success by which an OSS project manages to recruit new contributors to its community is critical for its long term sustainability. Hence, good practice with respect to governance of OSS projects is a fundamental challenge.

6 Conclusions and Discussion

In this paper we have reported from a case study on the LO project. From our analysis we make a number of observations which strongly suggest a long-term sustainable community, something which is a fundamental prerequisite for long-term maintenance of software systems and digital assets.

The findings from our analysis of the LO project make an important contribution towards a deeper understanding of challenges regarding long-term sustainability of OSS communities. In many usage scenarios, it is often the case that digital assets outlive the software systems that were used to generate them. For this reason, both software systems and the organisational structure in which such are developed, need to be maintained over very long life-cycles. It is therefore essential that there exist software systems with associated long-term sustainable communities, and our findings indicate that LO has a developer community that is sustainable in the long term.

For future work we intend to extend our study of the LO community to allow analysis over a longer time period after the fork from OO. In so doing, we plan to include data from the OO branch in our analysis. Such a study would also include an exploration of the relationships between different organisations governing the LO and the OO projects, and an exploration of how contributors are committed to the two different branches. Previous research has shown that the extent to which there is a commercial drive in OSS projects may impact on community values and contributor engagement (Lundell et al., 2010). To extend our analysis of the LO project we plan to undertake further investigations on developer experiences with respect to this issue in order to enrich our findings.

References

Behlendorf, B.: How Open Source Can Still Save the World. In: Boldyreff, C., Crowston, K., Lundell, B., Wasserman, A.I. (eds.) OSS 2009. IFIP AICT, vol. 299, pp. 2–2. Springer, Heidelberg (2009)

Aaltonen, T., Jokinen, J.: Influence in the Linux Kernel Community. In: Feller, et al. (eds.) Open Source Development, Adoption and Innovation, pp. 203–208. Springer, Berlin (2007)

Ågerfalk, P., Fitzgerald, B.: Outsourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy. MIS Quarterly 32(2), 385–410 (2008)

Bacon, J.: The Art of Community. O'Reilly, Sebastopol (2009)

Blondelle, G., Arberet, P., Rossignol, A., Lundell, B., Labezin, C., Berrendonner, R., Gaufillet, P., Faudou, R., Langlois, B., Maisonobe, L., Moro, P., Rodriguez, J., Puerta Peña, J.M., Bonnafous, E., Mueller, R.: Polarsys towards Long-Term Availability of Engineering Tools for Embedded Systems. In: Proceedings of the sixth European Conference on Embedded Real Time Software and Systems (ERTS 2012), Toulouse, France, February 1-2 (2012)

Bonaccorsi, A., Rossi, C.: Comparing Motivations of Individual Programmers and Firms to Take Part in the Open Source Movement: From Community to Business. Knowledge, Technology & Policy 18(4), 40–64 (2006)

Byfield, B.: The Cold War Between OpenOffice.org and LibreOffice. Linux magazine (October 22, 2010), http://www.linux-magazine.com/Online/Blogs/Off-the-Beat-Bruce-Byfield-s-Blog/The-Cold-War-Between-OpenOffice.org-and-LibreOffice (accessed March 3, 2012)

Crowston, K., Kangning, W., Howison, J., Wiggins, A.: Free/Libre open-source software development: What we know and what we do not know. ACM Computing Surveys 44(2), Article 7 (2012)

Deshpande, A., Riehle, D.: The Total Growth of Open Source. In: Russo, B., et al. (eds.) Open Source Development, Communities and Quality. IFIP, vol. 275, pp. 197–209. Springer, Boston (2008)

documentfoundation.org: The Document Foundation (2012), `http://www.documentfoundation.org/` (accessed March 3, 2012)

Engelfriet, A.: Choosing an Open Source License. IEEE Software 27(1), 48–49 (2010)

Fitzgerald, B.: A Critical Look at Open Source. IEEE Computer 37(7), 92–94 (2004)

Gamalielsson, J., Lundell, B., Lings, B.: The Nagios Community: An Extended Quantitative Analysis. In: Ågerfalk, P., Boldyreff, C., González-Barahona, J.M., Madey, G.R., Noll, J., et al. (eds.) OSS 2010. IFIP AICT, vol. 319, pp. 85–96. Springer, Heidelberg (2010)

Gamalielsson, J., Lundell, B.: Open Source communities for long-term maintenance of digital assets: what is offered for ODF & OOXML? In: Hammouda, I., Lundell, B. (eds.) Proceedings of SOS 2011: Towards Sustainable Open Source, Tampere University of Technology, Tampere, pp. 19–24 (2011) ISBN 978-952-15-2411-0, ISSN 1797-836X

Gamalielsson, J., Lundell, B., Mattsson, A.: Open Source Software for Model Driven Development: A Case Study. In: Hissam, S.A., Russo, B., de Mendonça Neto, M.G., Kon, F. (eds.) OSS 2011. IFIP AICT, vol. 365, pp. 348–367. Springer, Heidelberg (2011)

Gary, K., Koehnemann, H., Blakley, J., Groar, C., Mann, H., Kagain, A.: A Case Study: Open Source Community and the Commercial Enterprise. In: 2009 Sixth International Conference on Information Technology: New Generations, pp. 940–945. IEEE Computer Society, Los Alamitos (2009)

German, D.: The GNOME project: a case study of open source global software development. Journal of Software Process: Improvement and Practice 8(4), 201–215 (2003)

Gmane.org: Information about gmane.os.openbsd.cvs (2012), `http://dir.gmane.org/gmane.os.openbsd.cvs` (accessed March 3, 2012)

Israeli, A., Feitelson, D.G.: The Linux kernel as a case study in software evolution. Journal of Systems and Software 83(3), 485–501 (2010)

de Laat, P.: Governance of open source software: state of the art. Journal of Management and Governance 11(2), 165–177 (2007)

Lopez-Fernandez, L., Robles, G., Gonzalez-Barahona, J.M., Herraiz, I.: Applying Social Network Analysis Techniques to Community-driven Libre Software Projects. International Journal of Information Technology and Web Engineering 1(3), 27–48 (2006)

Lundell, B.: e-Governance in public sector ICT-procurement: what is shaping practice in Sweden? European Journal of ePractice 12(6) (2011), `http://www.epractice.eu/en/document/5290101`

Lundell, B., Gamalielsson, J.: Towards a Sustainable Swedish e-Government Practice: Observations from unlocking digital assets. In: Proceedings of the IFIP e-Government Conference 2011(EGOV 2011), Delft, The Netherlands, August 28-September 2 (2011)

Lundell, B., Lings, B., Lindqvist, E.: Open Source in Swedish companies: where are we? Information Systems Journal 20(6), 519–535 (2010)

Lundell, B., Lings, B., Syberfeldt, A.: Practitioner Perceptions of Open Source Software in the Embedded Systems Area. Journal of Systems and Software 84(9), 1540–1549 (2011)

Madey, G., Freeh, V., Tynan, R.: Modeling the F/OSS community: A quantitative investigation. In: Koch, S. (ed.) Free/Open Source Software Development, pp. 203–221. Idea Group Publishing, Hershey (2004)

Martinez-Romo, J., Robles, G., Ortuño-Perez, M., Gonzalez-Barahona, J.M.: Using Social Network Analysis Techniques to Study Collaboration between a FLOSS Community and a Company. In: Russo, B., et al. (eds.) Open Source Development, Communities and Quality. IFIP, vol. 275, pp. 171–186. Springer, Boston (2008)

Mens, T., Fernández-Ramil, J., Degrandsart, S.: The evolution of Eclipse. In: Proceedings of the 24th IEEE International Conference on Software Maintenance (ICSM 2008), pp. 386–395 (2008)

Michlmayr, M.: Community Management in Open Source Projects. The European Journal for the Informatics Professional X(3), 22–26 (2009)

Michlmayr, M., Robles, G., Gonzalez-Barahona, J.M.: Volunteers in Large Libre Software Projects: A Quantitative Analysis. In: Sowe, S.K., et al. (eds.) Emerging Free and Open Source Software Practices, pp. 1–24. IGI Publishing, Hershey (2007)

Mockus, A., Fielding, R.T., Herbsleb, J.D.: Two case studies of open source software development: Apache and Mozilla. ACM Transactions on Software Engineering and Methodology 11(3), 309–346 (2002)

Moon, Y.J., Sproull, L.: Essence of distributed work: The case of the Linux kernel. First Monday 5(11) (2000)

Müller, R.: Open Source – Value Creation and Consumption, Open Expo, Zürich (September 24-25, 2008), http://www.openexpo.ch/fileadmin/documents/2008Zuerich/Slides/35_Mueller.pdf (accessed March 3, 2012)

Nouws, C.: LibreOffice– the first year and looking forward! presented at ODF plugfest, Gouda, Netherlands (November 18, 2011), http://odfplugfest.org/2011-gouda/Nouws-LibreOffice.pdf (accessed March 3, 2012)

Noyes, K.: Don't Count on Oracle to Keep OpenOffice.org Alive, PC World (August 23, 2010), http://www.pcworld.com/printable/article/id,203910/printable.html

Nyman, L., Mikkonen, T.: To Fork or Not to Fork: Fork Motivations in SourceForge Projects. In: Hissam, S.A., Russo, B., de Mendonça Neto, M.G., Kon, F. (eds.) OSS 2011. IFIP AICT, vol. 365, pp. 259–268. Springer, Heidelberg (2011)

Nyman, L., Mikkonen, T., Lindman, J., Fougère, M.: Forking: the Invisible Hand of Sustainability in Open Source Software. In: Hammouda, I., Lundell, B. (eds.) Proceedings of SOS 2011: Towards Sustainable Open Source, pp. 1–5. Tampere University of Technology, Tampere (2011) ISBN 978-952-15-2411-0, ISSN 1797-836X

openoffice.org: Community Council (2012), http://wiki.services.openoffice.org/wiki/Community_Council (accessed March 3, 2012)

Robert, S.: On-board software development - The open-source way. In: IST/ARTEMIS Workshop, Helsinki, November 22 (2006)

Robert, S.: New trends and needs for Avionics Systems. In: ARTEMIS Conference, June 4-5, Berlin (2007)

Robles, G., Gonzalez-Barahona, J.M., Michlmayr, M.: Evolution of volunteer participation in libre software projects: evidence from Debian. In: Proceedings of the First International Conference on Open Source Systems, pp. 100–107 (2005)

Samoladas, I., Stamelos, I., Angelis, L.: Survival analysis on the duration of open source projects. Information and Software Technology 52(9), 902–922 (2010)

Shaikh, M., Cornford, T.: 'Letting go of Control' to Embrace Open Source: Implications for Company and Community. In: Proceedings of the 43rd Hawaii International Conference on System Sciences - 2010, pp. 1–10. IEEE Computer Society, Los Alamitos (2010)

Ven, K., Huysmans, P., Verelst, J.: The adoption of open source desktop software in a large public administration. In: Proceedings of the 13th Americas Conference on Information Systems (AMCIS 2007), Keystone, Colorado, August 9-12, pp. 9–12 (2007)

Ven, K., Mannaert, H.: Challenges and strategies in the use of Open Source Software by Independent Software Vendors. Information and Software Technology 50(9-10), 991–1002 (2008)

Wheeler, D.A.: Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers (2007), `http://www.dwheeler.com/oss_fs_why.html` (revised April 16, 2007) (accessed March 3, 2012)