Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 64 (2015) 924 – 931

Procedia Computer Science

Conference on ENTERprise Information Systems / International Conference on Project MANagement / Conference on Health and Social Care Information Systems and Technologies, CENTERIS / ProjMAN / HCist 2015 October 7-9, 2015

Understanding Enterprise Open Source Software Evolution

Lerina Aversano*, Marco Di Brino, Daniela Guardabascio, Marcello Salerno and Maria Tortorella

Department of Engineering University of Sannio
P.zza Roma 21, 82100 Benevento, Italy

Abstract

Enterprise Open Source Software is continuously gaining acceptance in business organizations. This is essentially due to the understanding of the potential benefits deriving from the adoption of OSS project solution. Indeed, Open Source Software solutions offer great opportunities for cost reduction and quality improvement, especially for small and medium enterprises that typically have to address major difficulties due to the limited resources. In this direction it is relevant understand and gain knowledge regarding the evolution of such software over systems the time. This paper report results of an empirical study aimed at analyzing the evolution of most relevant ERP open source system during their lifetime.

Peer-review under responsibility of SciKA - Association for Promotion and Dissemination of Scientific Knowledge

Keywords: Software evolution; Empirical study; Open Source Software; Enterprise Resource Planning.

1. Introduction

Enterprise Resource Planning Open Source software promises significant benefits to organizations. They include lowering costs, reducing inventories, increasing productivity, improving operational efficiency and achieving competitive advantages. In this context, it is relevant to gain a deep knowledge regarding the way these software

* Corresponding author. Tel.: +39 0824 305551.
E-mail address: aversano@unisannio.it

Peer-review under responsibility of SciKA - Association for Promotion and Dissemination of Scientific Knowledge
doi:10.1016/j.procs.2015.08.609

systems are evolving during their lifecycle. Specifically, this issue has been more generally addressed in the software engineering research area but it still needs a deep analysis with reference to the ERP Open Source Software.

It is widely recognized that the software systems continue to evolve after the first version has been distributed. Numerous studies indicate that the costs associated with software maintenance and evolution are at least 50%, and sometimes more than 90% of the total costs associated with a software system. As a consequence, both managers and developers must understand the factors that drive software evolution and take proactive steps that facilitate changes and ensure that the software systems do not decay. On the other hand, even adopters need to understand if the systems are adequately maintained during their evolution.

Currently there is the availability of repositories of large open source applications with lifetimes that exceed 20 years. Then, the empirical study presented in this paper aims to leverage software evolution data contained in historic program versions, and to obtain a clearer image of the software evolution process. To this end, the complete release histories of the major ERP open source systems, such as OpenBravo, Odoo, Dolibarr, Adempiere and so on, have been analyzed. The remainder of this paper is organized as follows. Section 2 analyses existing research work related to software evolution. Section 3 provides a description of the design of the empirical study. Section 4 describes the results of the evaluation of ERP OSS projects. Concluding remarks are given in the last section.

2. Related work on software evolution

The development of a theory of software evolution is a relevent research and several empirical studies of software evolution have been proposed in the literature. However, it is encouraging to see that the empirical study on the Open Source evolution has recently become a topic of interest.

M. Lehman and colleagues have proposed the main reference studies on software evolution over a 30-year period dating back to the mid-1970. The studies have given rise to eight laws of software evolution, as formulated and refined by Lehman and colleagues[1,2,3]. These laws are the result of careful and challenging empirical studies of the evolution of large-scale software systems found in a variety of corporate-based settings.

Other studies have also been conducted and found to yield consistent growth models. The data is summarized as a set of growth curves[3]. In plotting these growth curves as graphs, the size of the system (e.g., measured in terms of number of modules) after the first release is analyzed with reference to the sequence number of the software release. The analysis suggests that, during its evolution, a system tracks a growth curve that can be approximated either as linear or inverse-square model[8] within phases of the lifetime of an application, separated by transitions. The combined patterns can be interpreted as "S" curves, with phases of growth in size followed by less rapid growth (even stagnation) and transitions to another growth phase. Thus, these data/curves explicate conformity to the first (continual change), second (increasing complexity) and sixth (continual growth) laws, in that they suggest continual adaptation via incremental growth, and the system complexity controls the growth rate in a constant/bounded (linear or inverse-square) manner. The third, fourth and fifth laws have been refined to take into account the new observations. The seventh law addressing quality could not be directly observed within the data, but may conform to the observations made by Lehman and colleagues regarding these systems. The eighth law (feedback system) is a synthesis of the other laws. The support to each of the other laws strengthens the eight. In general, in the most recent studies by Lehman and colleagues, their data set and diversity of data substantiates and supports the original or refined versions of the laws.

However, it is unclear whether such a data set is a representative sample of the different kinds of software systems, or whether the laws can be interpreted as providing theoretical guidance for what kinds of software systems to study.

Other empirical studies have been conducted and published[4,6,7]. Here, the attention is directed to the Enterprise software systems category that has been analyzed. This is mainly intended to see if this study of ERP software category can confirm, refute, or otherwise extend and refine the laws and theory of software evolution.

3. Design of the study

The study described in this paper followed the same procedure for each analyzed system. First of all, the publicly available official releases have been downloaded, starting from the most recent one and going back as far as it was possible. Then, the metrics to be considered for the analysis have been identified. Specifically it has been decided to observe the evolution of: SLOC- Source Lines of Code, commits, contributors and number of download. These metrics, indeed, allow a comparison of different point of views, such as: the developers point of view and final user point of view. Therefore, the metrics evaluated for each system have been collected. The proposed empirical study concerned several ERP Open Source Software Systems without considering the programming language. In selecting the software systems, few criteria have been adopted. First, since we are interested in long- term software evolution, the systems had to have a long release history. Second, the ERP systems had to be sizable, so we could understand the issues appearing in the evolution of realistic, multi-developer software. Third, the ERP systems had to be actively maintained. Finally, the ERP systems had to have a high number of download.

Table 1 presents the download data for ERP Open Source Systems. Specifically, the second and third columns present the number of download, and data of the last updated release, respectively.

Table 1. ERP Open Source Software selected.

Project	Download 2014	Last Update
Viger CRM	370.869	19-02-2015
SugarCRM	214.612	15-12-2014
Dolibarr ERP – CRM	188.635	27-12-2014
Openbravo	101.708	24-02-2015
xTuple ERP: Postbooks	96.092	23-03-2015
Adempiere ERP	83.720	02-03-2015
Project Open	60.475	07-03-2015
CiviCRM	55.491	17-03-2015
Web ERP	41.342	06-02-2015
EPESI	25.914	16-03-2015
Opentaps	24.606	01-11-2013
Opencrx	20.116	24-02-2015
Jfire	17.628	25-04-2013
Vienna Advantage	15.554	05-03-2015
LedgerSMB	12.847	17-01-2015
MyIT CRM	8.119	21-01-2012

We aimed to analyze complete lifespans for each application, from the first publicly available release to the latest release as of March 2015.

A first aspect observed in the study regards the fact that the selected projects are all recently updated, with few exceptions among the less popular projects. This data confirms the point of view that the most successful products in open source software are those that have an active community behind, following the project and frequently updating it.

A second interesting finding concerns the fact that the two most common systems are purely CRM solutions, which have a very significant number of downloads, also in relation to the most popular ERP solutions.

For gathering the data from the project repository, the Openhub.net has been used. It is an online platform that deals with the automated retrieval of information from the repository of all the most important open source projects. These data were integrated with data from two extraction metrics tools, which are Statsvn and Gitstats.

Open HUB, formerly known as Ohloh, is a suite of web services providing an online community platform, with the goal of keeping track of the development of open source software systems. It was founded in 2004 by two former Microsoft manager, and more than 650,000 projects have been tracked in September 2014. Open HUB provides statistics about the projects, their licenses and software metrics: such information is made available thanks to a preliminary work of automatic retrieval of information from the versioning systems.

StatSVN is a tool written entirely in Java that allows the easy retrieval of information from the Subversion repository. It generates a set of charts and tables that describe the progress of the development process within the repository, such as the activities of the various contributors, changes to files and other useful metrics to analyze the work done on the repository in the course of his life.

Gitstats is a generator of statistics for Git repository. It analyses the repository and produces as output a series of tables and graphs available as HTML files.

4. Results

The data in Openhub allowed to easily retrieve a lot of information of interest to the selected projects. Wishing to proceed to a first analysis of the sample, the data collected in Table 2 allow to make important considerations, especially if integrated with those already presented in Table 1. First of all, it can be observed as most popular ERP software systems, with the exception of Dolibarr, are large systems which have a significantly high number of SLOC (> 2.000.000). Consistent with this data, such systems are those that tend to exhibit greater activity on the repositories with a number of contributors greater than 80 and a number of commits greater than 10.000. However, there are some anomalous data, such as that relating to commits SugarCRM, far lower than that of most of the projects analyzed and the already mentioned Dolibarr ERP solution that appears to be the most downloaded but it presents a number of SLOC less than 500.000 . Figure 1 depict the trend of the lines of code of most popular ERP systems.

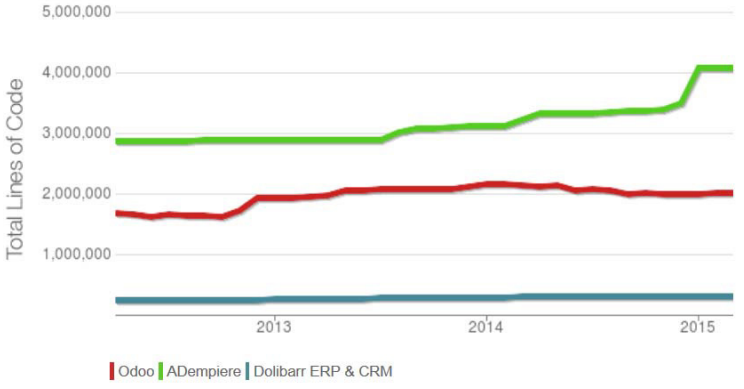


Fig. 1. Total lines of code excluding comments and blank lines

Regarding the number of contributors, the study aimed at understanding if the activities is evenly distributed among all contributors or whether there are contributors who perform a significant amount of work, more than others on a given repository. To explore this aspect, a fuller analysis of some projects with tools StatSVN and GitStats has been executed, depending on the VCS used, and it has been seen that, for many analyzed projects, the number of developers significantly active on the repository was a very small fraction of the total of the developers. In particular, the number of commits were analyzed, together with the minimum number of contributors who had made at least 80% of the number of commits in the repository. The obtained results are shown in Table 3. The analysis involved two projects not indexed on SourceForge but fairly popular in the field of open source applications Enterprise: ErpNExt and Apache OFBiz.

Table 2. Data related to all the repository history for the ERP and CRM

Project	SLOC	Contributors	Commits
		CRM Systems	
Vtiger CRM	2.007.507	23	8195
Sugar CRM	903.670	11	53
CiviCRM	2.190.005	511	49.384
Myit CRM	132.344	4	85
		ERP Systems	
Dolibarr	320.194	139	45.126
Openbravo	7.195.468	129	29.337
Postbooks	2.305.663	90	28.627
Adempiere	4.090.334	84	8.098
Project Open	2.450.200	41	14.523
webERP	677.877	28	18.503
EPESI	610.358	14	7.184
opentaps	1.352.015	29	10.788
opencrx	3.648.970	8	6.981
Jfire	870.254	27	19.419
Vienna Advantage	1.057.690	2	106
LedgerSMB	124.009	20	6.426

Table 3. Main contributors of the projects

Project	Total Number of contributors	Main contributors (80% of commits)
Adempiere	66	6
Dolibarr	60	3
SugarCRM	7	2
EPESI	12	3
ErpNext	26	3
LedgerSB	21	2
CiviCRM	183	16
Postbooks	34	4
OFBiz	28	8
VTiger CRM	29	9
WebERP	26	3

Tables 2 and 3 show that in all the analyzed cases, the number of developers that have carried out 80% of commits on the repository is a very small fraction of the total number, often below 10%. This allows us to assert that in the analyzed Enterprise Open Source Projects the work is done mainly by few developers, who are also the ones that tend to work more on the ongoing project, while a significant number of contributors gives a limited support to the repository.

For a more complete analysis and to understand which of the projects presented an ongoing activity over time, data relating to activities executed in the past 12 months have recovered from OpenHub.net. Table 4 shows these data. The projects with a greater popularity are those that have a higher activity in the last 12 months, with a

significant number of contributors and commits: such projects are stable in terms of annual trend or increasing. Conversely, little or no activity exists in the last 12 months for the less popular projects.

These data are also made explicit by the trend that almost always commits a stable trend or growth projects and popular and in decline for those less downloaded. Generally, projects prevail with a decreasing number of commits.

Table 4. Activities in the various repositories over the past 12 months

Project	Contributors last 12months	Commits last 12 months	Changed Files	Added Lines	Removed Lines	Commits trends
CRM						
Vtiger CRM	1	13	8.861	998.308	705.824	Increasing
Sugar CRM	1	1	192	3.373	1.636	Decreasing
CiviCRM	266	9.518	8.613	1.056.75 3	799.878	Stable
ERP						
Dolibarr	78	5.089	5.911	822.700	600.261	Stable
OpenBravo	41	3.664	3.295	3.051.76 6	877467	Increasing
Postbooks	39	3.745	7.580	7.198.54 3	1.481.96 5	Stable
Adempiere	10	483	8.663	7.393.06 8	3.364.02 7	Stable
Project Open	6	1.087	661	39.604	19.396	Stable
webERP	9	1.029	1.274	130.246	85.057	Decreasing
EPESI	5	534	904	61.665	35.309	Increasing
opentaps	0	0	0	0	0	Decreasing
opencrx	1	4	1.454	73.144	49.826	Decreasing
Jfire	0	0	0	0	0	Decreasing
Vienna Advantage	0	0	0	0	0	Decreasing
LedgerSMB	8	1.356	571	56.003	64.533	Increasing

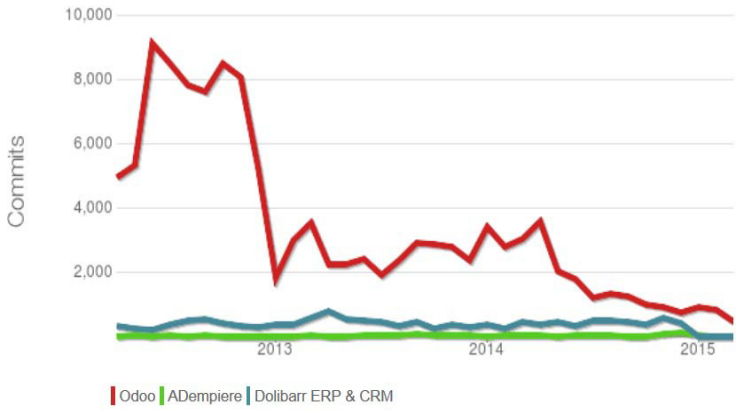


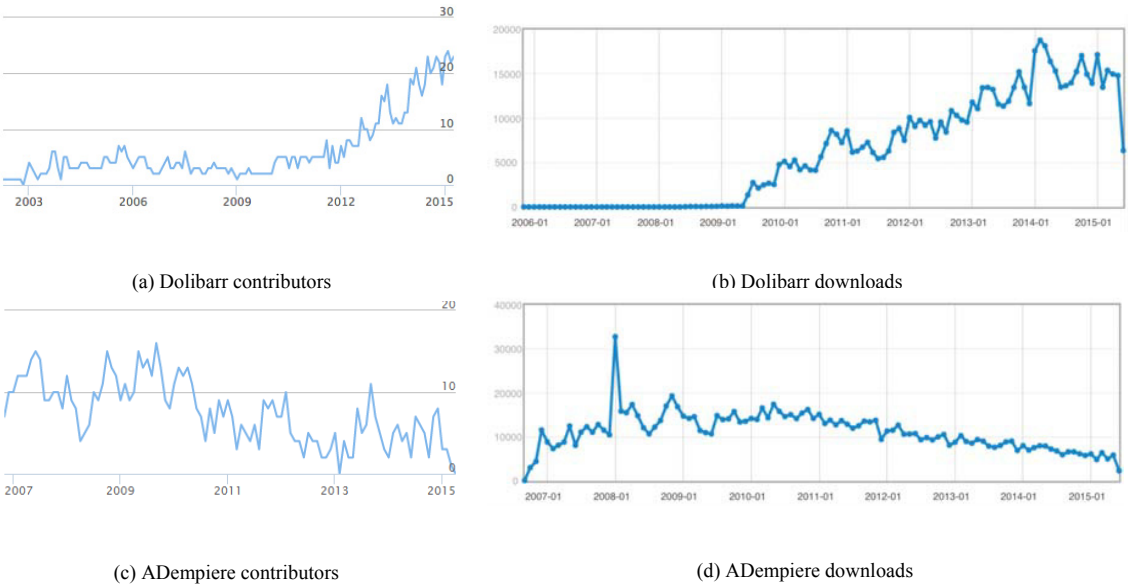
Fig. 2 Number of commits made to the project source code each month.

The number of contributors, and its evolution in time, differs significantly between the different projects: the only project that maintain a number of contributors constant in time is OpenBravo, which is also the largest system in terms of SLOC among those analyzed: this could highlight how large projects are more easily followed by a number

of people who tends to remain constant over time. Instead, other projects, such as Vtiger and PostBooks, present a decreasing number of contributors.

In the case of projects like Dolibarr, WebERP and CiviCRM, the number of contributors presents a constant increasing. In the case of Dolibarr, it is interesting to note that, although the system is among the smallest in terms of SLOC, it presents a growing number of contributors. Considering the fact that this system is extremely popular, it can be assumed that the project evolves, probably supported by the positive feedback from the high number of users who downloaded this system (see Table 1).

To explore this aspect, Figure 3 shows the number of downloads for Dolibarr over the years. It is interesting to note that the trend of contributors reflects in a certain way the download trend. This indicates that the two phenomena are somehow connected. Indeed, in the software development, open source community of developers and final users have a strong mutual influence and this confirms in some way the previous observation.
Dually, the number of contributors of projects such as Adempiere is decreasing over time and the number of downloads follows a similar trend, as shown in Figure 3 which highlights the evolution of contributors and downloads of Adempiere.



(a) Dolibarr contributors

(b) Dolibarr downloads

(c) ADempiere contributors

(d) ADempiere downloads

Fig. 3. Dolibarr and ADempiere: number of contributors and download

Analyzing the results obtained with StatSVN, it has been possible to observe a direct analogy with the research on software evolution of Godfrey[5]. Indeed, the analyzed systems tend to concentrate the greatest number of changes to a few files. Specifically, the number of files with a particularly high number of revision, is limited, and the number of revisions for files rapidly decreases. In addition, these changes are often related to the management of orders, that would lead to think that these files are the ones that contribute most significantly to the activities on the repository, undergoing frequent changes.

The order management functionality is definitely reference as a core business for the ERP systems and consequently it is likely that it is more susceptible to changes.

5. Conclusions

This paper proposes an analysis and comparison of some popular open source ERP and CRM solutions. The results of this study highlighted some important aspects summarized in the following.

The most successful projects are those that have a more intense activity in the last 12 months.

The development work is not uniformly distributed among the various contributors, as few contributors tend to make the large part of the work.

The study also showed that for some projects the number of contributors is connected to the popularity of the project, confirming that in the open source community the final users represents a motivation to concrete improvement of the software product, thanks to the direct interaction with the developers.

There is a limited set of files that are subject to significant changes made over time on a project, specifically these are the files belonging to the core business functions of Enterprise systems such as order management.

Only a small subset of the analyzed projects presents a growing number of commits in the last year analyzed.

References

1. Lehman, M. M., "Programs, Life Cycles, and Laws of Software Evolution". Proc. IEEE 68 (9): 1060–1076, 1980.
2. Lehman, M. M.; Ramil, J. F.; Wernick, P. D.; Perry, D. E.; Turski, W. M. (1997). "Metrics and laws of software evolution—the nineties view". Proc. 4th International Software Metrics Symposium (METRICS '97). pp. 20–32.
3. Lehman, M. M.,"On Understanding Laws, Evolution, and Conservation in the Large-Program Life Cycle". Journal of Systems and Software 1980, 1: 213–221.
4. Carbone, P., "Value Derived from Open Source is a Function of Maturity Levels, " in OCRI Conference – Alchemy of open source business, 2007.
5. Godfrey, M. W. and Tu, Q., "Evolution in Open Source software: A case study" in ICSM '00 Proceedings of the International Conference on Software Maintenance, San Josè, California, 2000, pp. 131-142.
6. Amor, J. J., Gonzalez-Barahona J. M., and Herraiz I. Robles, G., "Evolution and growth in large software projects," in Proceedings of the 8th International Workshop on Principles of Software Evolution, Lisbona, Portugal, 2005, pp. 165-174.
7. Mens, T., Demeyer, S., (Eds.), Software Evolution, Springer, 2008.
8. Turski., W.M., Reference model for smooth growth of software systems. IEEE Trans. on Software Engineering, 22(8), Aug 1996.