Electronic Communications of the EASST
Volume 33 (2010)

Proceedings of the Fourth International Workshop on Foundations and Techniques for Open Source Software Certification (OpenCert 2010)

Integrating Data from Multiple Repositories to Analyze Patterns of Contribution in FOSS Projects.

Sulayman K. Sowe and Antonio Cerone

17 pages

Guest Editors: Luis S. Barbosa, Antonio Cerone, Siraj A. Shaikh
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Integrating Data from Multiple Repositories to Analyze Patterns of Contribution in FOSS Projects.

Sulayman K. Sowe[1]* **and Antonio Cerone**[2]

[1] sowe@ias.unu.edu,
UNU-IAS, Yokohama, Japan.

[2] antonio@iist.unu.edu,
UNU-IIST, Macau SAR China.

Abstract: The majority of Free and Open Source Software (FOSS) developers are mobile and often use different identities in the projects or communities they participate in. These characteristics not only poses challenges for researchers studying the presence (where) and contributions (how much) of developers across multiple repositories, but may also require special attention when formulating appropriate metrics or indicators for the certification of both the FOSS product and process. In this paper, we present a methodology to study the patterns of contribution of 502 developers in both SVN and mailing lists in 20 GNOME projects. Our findings shows that only a small percentage of developers are contributing to both repositories and this cohort are making more commits than they are posting messages to mailing lists. The implications of these findings for our understanding of the patterns of contribution in FOSS projects and on the quality of the final product are discussed.

Keywords: Open Source Software developers, Open Source Software projects, Software repositories, Concurrent Versions System, Mailing lists, Linking data, Software Quality.

1 Introduction

Free and Open Source Software (FOSS) developers are like nomads; freely moving from one project to another. They commit bits and pieces of code, report and fix bugs, take part in discussions in various mailing lists, forums, and IRC channels, document coding ethics and guidelines, and help new entrants. Along the way they create and archive a wealth of knowledge and experience associated with their art [SAS06]. Participants in various projects use tools (Versioning Systems, mailing lists, bug tracking systems, etc.) to enable the distributed and collaborative software development process to proceed. These tools serve as repositories which can be data mined to understand *who* is involved, who is talking to *whom*, *what* is talked about, *how much* someone contributes in terms of code commits or email postings. Thus, by applying cyber-archeology [SII07] to these repositories, we can learn and better understand the patterns of contribution [SFF+06, GKS08] of FOSS developers in the projects concerned.

* *Correspondence Author*: Sulayman K. Sowe. *Email*: sowe@ias.unu.edu. *Address*: UNU-IAS, Yokohama 220-8502, Japan. Tel: +81-45-221-2300, Fax: +81-45-221-2302.

An important aspect of software engineering research, and the certification of FOSS products in particular, is understanding and measuring the contribution of individuals, particularly developers, who work on a project [SO09a, SAS06]. A host of factors which have both empirical and industrial implications motivates this kind of research. Factors include, but not limited to;

- helping practitioners understand and monitor the rate of project development,
- characterizing FOSS projects in terms of developers turnover and extent of contribution [CLM03, GKS08, SSSA08],
- identifying bottlenecks and isolate exceptional cases in terms of projects and individuals contributions [SO09a], and
- using the research results to develop new metrics or evaluate an existing taxonomy [SO09b] of metrics (Process, Product, and Resources) for FOSS quality attributes [SAOB02] and the certification process.

Furthermore, as argued by [SC09], communication and patterns of contribution are factors that contribute to measure the efficiency of the development process, a measure that the authors called "quality by development". Indeed, the patterns of [code] contribution in FOSS projects has emerged as an important measure in assessing the quality of FOSS products [SO09b, SAOB02].

A lot of research utilizes data from a single repository to analyze code contribution of developers [RG06, GKS08], trends and inequality in posting and replying activities in Apache and Mozilla [MFH02], KDE [Kuk06], Debian [SSL08], and FreeBSD [DB05]. Most of these researches use data from CVS or mailing lists as these are *de facto* repositories in FOSS projects. Source configuration management (SCM), of which CVS or SVN[1] is part, is mainly used to coordinate the coding activities of software developers and manage software builds and releases. Mailing lists, on the other hand, are the main communication channels [SSL08]. Many important aspects of a project are negotiated in [developer] lists: software configuration details, the way forward and how to deal with future requests, how tasks are distributed, issues concerning package dependencies, scheduling online and off-line meetings, etc. Thus, for a developer to keep abreast with developments in a project, committing code to SVN alone is not sufficient. S/he needs to participate in the respective lists, communicate his ideas, and engage with colleagues. To bolster this view, [Bro75] pointed out the essence of communication as a means to foster long term success of software projects. This may take the form of a bi-directional developer to developer, developer to user, and developer to community communication.

Even though a strong linkage exist between the information in FOSS repositories (e.g. bug reports and source code repositories [DB07, ZPZ07]), few researchers strive to understand how developers' contributions varies across repositories. In this research we tired to fill this niche by establishing links between SVN and mailing lists to locate developers who are present in both repositories and quantify their contribution in terms of commits and posts.

The rest of the paper is organized as follows. First, in section 2, we discuss the rationale behind this research and construct two hypothesis which will guide us for the rest of the paper.

[1] Note: SVN is our software code repository (see Subsection 3.1). Reference is made to CVS when other researchers mentioned using data from that repository.

In section 3, we outline the methodology and data used in this research and present our algorithm for identifying and quantifying developers contributions to both SVN and mailing lists. This is followed by an analysis and discussion of our results in section 4. Our concluding remarks and future work are presented in 5 section.

2 Research Rationale and Hypothesis

For software projects to evolve, it is by design that developers must continuously commit and review the codebase. In the eyes of the developer, user, and business community an active mailing list is a proxy of project success. The presence of project's leads, core and active developers in mailing lists has a profound effect on the way individuals within and outside the project see the commitment of the most influential members in the project. For software companies and private enterprises, developers presence in lists may indicate that software support activities are not only available from ordinary users, but also comes form individuals behind the software and project. Thus, developers should strive to balance their coding activity with their involvement in mailing lists. This raises a number of questions which may be of great interest to both FOSS project administrators and researchers. For instance;

- *How many developers are willing to commit code and patches and at the same time participate in discussions in mailing lists and other project's fora?*
- *If developers are coding more than they are participating in mailing lists, what does this tell us about the maintenance and dynamics of the software and project?*
- *How much effort can a developer allocate to one activity and at what stage in the project's life-cycle?*
- *If attaining a balance activity is much required in a project, how can project administrators schedule and assign or dedicate one activity at the expense of another?*
- *What is the impact on the performance the project of having developers specializing in one activity?*

In this research, we used data provided by the FLOSSMetrics project[2] to proposed a methodology to help us answer some of the above questions. FOSS researchers (e.g. [MFH02, Kuk06, DB05]) study and report developers coding activities separately from their mailing lists activities. However, research on the FOSS development process [Mas05, SFF+06] informs us that in many projects, a small number of talented core developers or "cod gods" [RG06] are busily (as if in a *software beehive*) submitting patches and tinkering with code to produce good and usable software for the rest of the community. This cohort also contribute to discussions in mailing lists; interacting with other software developers and users, keeping abreast with project activities and monitoring what goes on in there projects or packages [SSL08]. Nevertheless, we conjecture that not all the developers who commit or make changes to a project's source repository also participate in [developer] mailing lists. This study investigates the contributions of FOSS developers to

[2] http://flossmetrics.org/; Last visited: Monday, November 29, 2010.

both SVN and developer mailing lists and presents a methodology to overcome the empirical research challenges associated with integrating or linking data from multiple repositories. That is, we find out if developers are coding through commits in SVN as much as they are participating in mailing lists. This involves correlating developers commits activities with their corresponding mailing lists activities within the same project. Hypothesis put forward in this research are the following;

- **Hypothesis** [H1]: Since developers must code and commit, *ad infinitum*, for the software and project to evolve, we hypothesize that *FOSS developers make more commits to a project's code (SVN) repository than they are posting messages to mailing lists.*
- **Hypothesis** [H2]: However, we posit that developers must strike a balance between their coding and mailing lists activities. Thus, *FOSS developers contribute equally to code repository and mailing lists*.

3 Research Methodology

The methodology employed in this research investigates the simultaneous occurrence of developers in SVN and mailing lists. That is, identifying developers who make both commits and postings, and ensuring that the developer making the SVN commit is the same individual posting to the developer mailing list(s) of the same project. The methodology also ensures that developers with multiple identities are only counted once. The methodology, as represented in figure 1, shows the FLOSSMetrics database as the data source from which we extracted SVN and mailing lists data for the 20 projects in our study. In our data acquisition, a fundamental question is always asked; "Is this the same developer we have in both repositories ?" Figure 1 also shows the MYSQL database tables and fields from which we extracted commits and posts which are used in our analysis to identify developers (see subsection 3.3) in the projects. The links between the tables as indicated by the arrows (with "IS") shows the path taken to locate a developer and counting his contribution to both SVN and mailing lists.

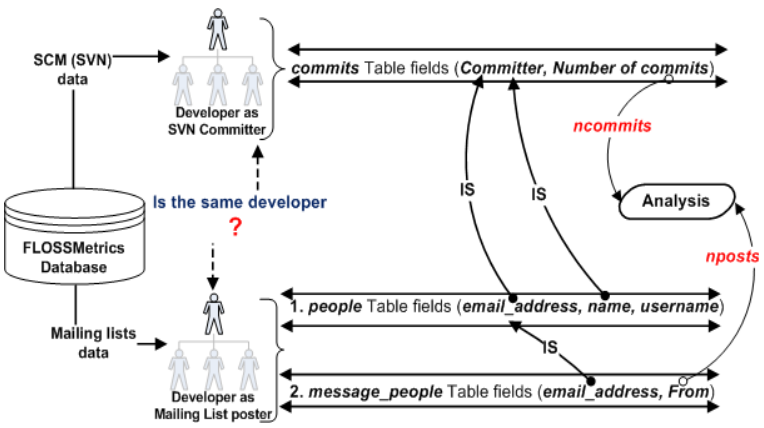

Figure 1: Methodology to Identify developers from multiple repositories.

3.1 Data

The data for this research consists of the 20 GNOME projects shown in table 1. The FLOSS-

Table 1: List of GNOME projects studied

No.	Projects	No.	Projects
1	Balsa	11	GNOME Control Center
2	Brasero	12	GNOME Games
3	Deskbar Applet	13	GNOME Media
4	Ekiga	14	GNOME Power Manager
5	Eog	15	GNOME Screensaver
6	Epiphany	16	GNOME System Tools
7	Evince	17	Libsoup
8	Evolution	18	Metacity
9	GDM	19	Nautilus
10	gedit	20	Seahorse

Metrics database retrieval system uses a combination of tools[3] to retrieve data from projects (e.g. GNOME and Apache) and forges (e.g. SourceForge) and computes various code and community metrics. The *CVSAnalY2* [ARG06, RGCH09, SSSA08] tool retrieves Source Content Management systems (SCM) data and stores committers attributes into various tables. The *MLStats* [SSSA08] tool extracts one or more mailing lists archives of a particular project. For each of the 20 projects, committers SVN identifications (commit ID) and the total number of commits each committer made is extracted. For the mailing list data, for each project, data was extracted from two FLOSSMetrics database tables: Two fields (type_of_recipient and **email_address**) from the *"messages_people"* table. The type_of_recipient field has the format "**From**", "To", and "Cc". The "From" email header is used to identify lists posters [QJ04] and counting their contribution to mailing lists [SAS06]. And three fields (**email_address**, **name**, and **username**) from the *"people"* table.

3.2 Data Cleaning

Having identified fields needed to analyze developers participation in SVN and mailing lists, we proceeded with data cleaning. For the mailing lists data, since we need both the "name" and "username", all posters without recognizable names and/or usernames were removed. Some of the names contained unrecognizable characters such as "*=?ISO88591?Q?g=FCrkan_g=FCr?=*". Some of the posts with null posters/developere were also removed. Furthermore, since the full name (first +last) is needed to identify a developer, all posters with a single name were deleted from the mailing lists data. That is, delete developer "*Foo*" but retain developer "*Foo Bar*". For the SVN data, all commits without committers or authors were removed. Aggregate number of items deleted in each of the above categories were; Unrecognizable characters = 28, Posts with null posters = 30, Posters with a single name = 14, and commits without authors = 5093.

[3] http://tools.libresoft.es/; Last visited: Monday, November 29, 2010

3.3 Identification of developers across repositories

As depicted in figure 1, a poster in the mailing lists can be identified in two ways. In the *messages_people* table, a poster is identified by his email address. By using the "*From*" field, all the emails posted by a particular person can be aggregated . The *people* table is used to identify a poster through his *"email address"*, poster *"name"* in the form of first name + last name (eg. Pawel Salek), and *"username"* (eg. pawsa). For the SVN data, the committer field from the *commits* table was used to identify a committer or author of a commit. In SVN, an individual is simply identified as a "Committer" or an "Author" of one or more commits. Mailing lists participants, on the other hand, can be identified by means of message identifiers like "From:" in email headers [SAS06]. The identification process proceeds thus;

1. For each project in the *commits* table, LIST all the committers and for each committer (unique commit ID or *commit_id*) SUM all his commits and store the value as *ncommits* variable.

2. For each project in the *people* table, LIST ("email address" + "name" + "username" or *poster_id*) WHERE both name and username is the same for this committer as in the *commits* table. And

3. From the *messages_people* table, LIST developers "email address", WHERE *people*.email address = *messages_people*.email_address. For each developer, COUNT all the posts and store the value as *nposts* variable.

The results of a typical query is shown in figure 2, with developers emails anonymized. From the query, it can be seen that a developer may appear many times. This is because, while a developer has only one identification in SVN, his commit id, the same developer may use many email addresses when posting messages to developer mailing lists.

email	full_name	poster_id	commit_id	nposts	ncommits
[redacted]@ximian.com	Federico Mena Quintero	federico	federico	28	100
[redacted]@gnu.org	Federico Mena-Quintero	federico	federico	1	100
[redacted]@bentspoon.com	Darin Adler	darin	darin	7	5
[redacted]@redhat.com	Havoc Pennington	hp	hp	2	2
[redacted]@gnome.org	Christian Rose	menthos	menthos	1	58
[redacted]@menthos.com	Christian Rose	menthos	menthos	1	58
[redacted]@home-of-linux.org	Martin Baulig	martin	martin	6	90
[redacted]@ximian.com	Michael Meeks	michael	michael	3	17
[redacted]@triq.net	Jens Finke	jens	jens	67	581
[redacted]@eknif.de	Jens Finke	jens	jens	11	581
[redacted]@gnome-db.org	Carlos	carlos	carlos	1	8
[redacted]@gnome.org	Jody Goldberg	jody	jody	3	1
[redacted]@inkstain.net	John Fleck	jfleck	jfleck	1	3
[redacted]@redhat.com	Alexander Larsson	alexl	alexl	2	1
[redacted]@linuxrising.org	Christian Schaller	Uraeus	uraeus	1	2
[redacted]@gnome.org	Lucas Rocha	lucasr	lucasr	33	479
[redacted]@svn.gnome.org	Felix Riemann	friemann	friemann	42	460
[redacted]@gnome.org	Felix Riemann	friemann	friemann	11	460
[redacted]@cvs.gnome.org	Felix Riemann	friemann	friemann	2	460
[redacted]@alumnos.utalca.cl	Claudio Saavedra	csaavedra	csaavedra	41	202
[redacted]@gnome.org	Claudio Saavedra	csaavedra	csaavedra	23	202
[redacted]@igalia.com	Claudio Saavedra	csaavedra	csaavedra	3	202

Figure 2: Query showing the identification of FOSS developers from SVN and Mailing lists.

3.3.1 Unmasking Aliases and removing duplicates

The volunteering nature of the FOSS development process and participation in public repositories means that participants may use different emails. For example, as shown in figure 2, a developer (e.g. Felix Riemann) has his identity masked in three email aliases; foo@svn.gnome.org, bar@gnome.org, foo.bar@cvs.gnome.org. The fundamental problem in email alias unmasking [BGD+06, SSL08] is finding out that those aliases all belong to one developer. The algorithm for checking duplicate records and unmasking aliases in the mailing lists data proceeded thus;

```
------Begin Algorithm-------------------------------

For ALL records in project X
IF ncommits OCCURS MORE THAN ONCE for THIS developer
AND poster_id = commit_id
RETURN ''THIS is a duplicate''
RECORD ONLY 1 value of ncommits for THIS developer
TOTAL posts for this developer = SUM of nposts.

------End Algorithm---------------------------------
```

The query scenario in figure 3 shows the result when the algorithm is applied to the dataset. This literally means; a developer (e.g. *Federico Mena Quintero* in figure 2) with a unique commit_id (federico) made 100 commits to the project's SVN. However, he contributed to mailing lists using two emails (foo@ximian.com and foo.bar@gnu.org). He posted 28 messages using the first email and 1 message using the second email. The developer's overall email postings is the sum of the two posts he made using the different emails, i.e. 28 + 1 = 29. All duplicate

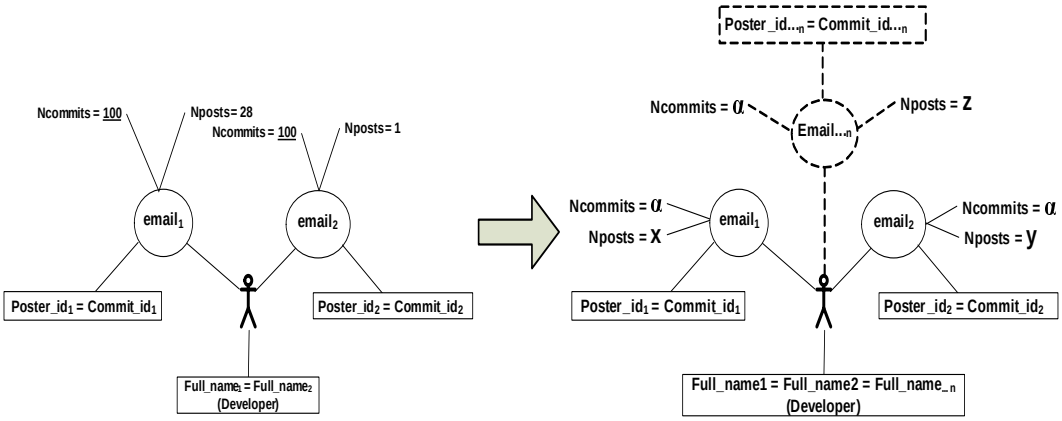

Figure 3: Query scenario to identify developers in SVN and mailing lists

records are identified and developers *nposts* and *ncommits* are calculated in a similar manner. There were an average of 115 duplicate records of this nature per project in our dataset. This means that many developers are using multiple email addresses. Generally, as shown on the right hand side of figure 3, a developer contribution to mailing list (nposts) will be counted as X + Y + Z, whilst his SVN contribution (ncommits) will be counted as α.

4 Analysis and discussion

According to [Sek06], an exploratory study is undertaken "when not much is known about the situation at hand or no information is available on how similar problem or research issues have been solved in the past". Thus, we begin our analysis using what we call an *exploratory data analysis (*EDA*) technique* to help us examine the distribution, the nature of the commits and posts, and prepare the ground for what may be the appropriate analysis technique to be used to answer the research hypothesis. Tables 2 and 3 shows the descriptive statistics of the developers posting and committing activities after data cleaning.

Table 2: Descriptive statistics of Posts

Projects	N Posters	Mean	Median	Std. Dev.	Skewness	Std. Err. of Skewness	Max.	Sum
Balsa **	1088	12.98	2.00	67.273	13.942	.074	1465	14125
Brasero	63	4.13	1.00	8.071	3.498	.302	45	260
Deskbar Applet	97	7.00	2.00	19.187	5.200	.245	137	679
Ekiga **	729	9.24	3.00	59.999	22.103	.091	1509	6734
Eog	134	4.17	1.50	8.914	4.533	.209	67	559
Epiphany **	889	5.91	1.00	23.795	12.657	.082	470	5250
Evince **	451	3.46	1.00	13.093	14.013	.115	238	1562
Evolution **	4769	7.44	2.00	46.274	25.619	.035	1760	35478
GDM **	658	3.99	1.00	25.597	20.006	.095	595	2628
gedit **	571	3.95	1.00	15.653	14.252	.102	306	2253
GNOME Power Manager **	203	5.58	2.00	33.046	13.881	.171	470	1133
GNOME Control Center	174	8.36	2.00	20.936	5.261	.184	186	1455
GNOME Games	173	8.79	2.00	25.146	5.909	.185	224	1521
GNOME Media	289	5.39	2.00	12.270	5.884	.143	115	1557
GNOME Screensaver	27	5.59	3.00	7.846	3.322	.448	39	151
GNOME System Tools **	297	4.51	1.00	11.019	6.076	.141	112	1339
Libsoup **	52	3.73	1.00	8.761	6.326	.330	63	194
Metacity	60	4.82	2.00	11.029	5.301	.309	77	289
Nautilus **	2065	8.61	2.00	61.402	32.822	.054	2475	17782
Seahorse	62	6.16	2.00	18.382	5.390	.304	122	382
Total	**128,512**							**95,331**

Table 3: Descriptive statistics of Commits

Projects	N Committers	Mean	Median	Std. Dev.	Skewness	Std. Err. of Skewness	Max.	Sum
Balsa	181	44.09	4.00	241.309	9.233	.181	2688	7981
Brasero ++	86	26.05	5.00	137.976	8.869	.260	1271	2240
Deskbar Applet ++	133	19.67	5.00	84.751	8.413	.210	834	2616
Ekiga	186	41.99	5.00	286.865	12.130	.178	3757	7810
Eog	298	16.59	4.00	53.660	8.231	.141	581	4944
Epiphany	252	34.84	6.00	217.618	14.340	.153	3352	8780
Evince	203	17.30	4.00	59.881	7.494	.171	535	3511
Evolution	430	81.11	10.00	309.253	8.099	.118	4061	34877
gdm	282	23.63	5.00	103.297	9.653	.145	1266	6663
gedit	329	20.68	5.00	81.699	10.704	.134	1153	6804
GNOME Power Manager	148	22.75	5.00	161.952	12.060	.199	1974	3367
GNOME Control Center ++	423	21.39	6.00	55.908	6.917	.119	634	9049
GNOME Games ++	321	27.68	7.00	89.618	8.559	.136	1164	8884
GNOME Media ++	324	13.05	4.00	31.803	6.804	.135	345	4228
GNOME Screensaver ++	126	12.79	4.00	74.388	11.097	.216	838	1611
GNOME System Tools ++	207	20.55	5.00	82.615	10.079	.169	1043	4254
Libsoup	37	32.49	1.00	111.261	5.067	.388	647	1202
Metacity ++	264	15.50	4.00	61.675	8.547	.150	600	4091
Nautilus	395	37.52	7.00	126.202	8.529	.123	1712	14822
Seahorse ++	137	21.39	5.00	99.481	9.603	.207	1087	2931
Total	**4,762**							**140,665**

As shown in table 2, for each project the total number of posters (N posters), the mean post per poster, the median, standard deviation, skewness, the maximum posts made by one individual, and the total or sum of postings for that project are shown. For all the projects, the mode and

minimum numbers of posts made equals 1. A total of 12,851 posters contributed 95,331 email messages. Table 3 shows the same descriptive statistics for the committers (N Committers) in each project. A total of 4,762 developers made 140,665 commits. Evident from the statistics is that each project has its unique characteristics [CLM03] in terms of developers' postings and committing activities, as well as the number of developers involved in each activity. For instance, 45% (N = 9) of the projects (marked with ++ in table 3) have more committers than posters. The other 55% (N = 11) of the projects (marked with ** in table 2) have more posters than committers.

Furthermore, figures 4 (both *Y*-axis in logarithmic scale) shows the distribution of posts and commits in the respective projects. From the boxplots it can be seen that the contributions of the developers to mailing lists is characterized by smaller means (post per poster). However, the posting data has many outliers; with many developers posting few emails and a few making large numbers of posts. On the contrary, the commits are characterized by larger means (commits per committer). These characteristics are reminiscent of power distributions observed in FOSS participants' contributions to mailing lists [SSL08] and CVS [MFT02] activities.

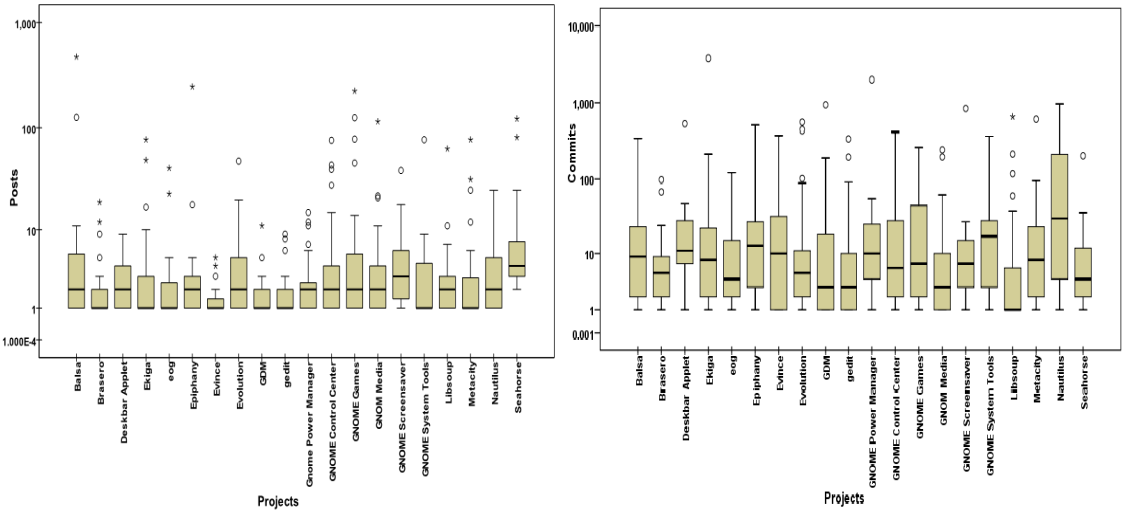

Figure 4: Box-plots showing the distributions of Posts and Commits.

4.1 Developers in both SVN and Mailing lists

In order to analyze the simultaneous occurrence of the developers in both repositories, we queried the SVN and mailing lists data for each project and computed developers contributions in terms of the *ncommits* and *nposts* variables discussed in subsection 3.3. Table 4 shows the number of developers (N_dev) in each project who contributed to both SVN and mailing lists. For the 20 projects, 502 developers made more commits (mean = 152.1; Std. deviation = 431.171) than posts (mean = 43.19; Std. deviation = 164.353).

Furthermore, as shown in figure 5, our identification technique and algorithm revealed a rel-

Table 4: Developers contribution to both SVN and Mailing Lists

		nposts					ncommits				
Projects	*N_dev.*	Mean	Median	Std.dev.	Max.	Sum	Mean	Median	Std.dev.	Max.	Sum
balsa	40	37.23	5.5	133.76	851	1489	112.53	25	206.33	751	4501
brasero	6	19.33	2	30.936	77	116	69.17	8.5	98.3	196	415
deskbar_applet	8	20.13	6	35.64	106	161	120.25	5.5	289.5	834	962
ekiga	4	438.25	121.50	722.49	1509	1753	2170.25	2417	1876.01	3757	8681
eog	16	18.81	4.5	25.95	78	301	129.38	37.5	196.16	581	2070
epiphany	40	55.73	7	116.69	470	2229	146.82	16.5	536.03	3352	5873
evince	18	27.17	2	58.61	238	489	100.89	10.5	180.31	535	1816
evolution	92	56.47	4.5	172.68	1481	5195	283.29	46	622.01	4061	26063
gdm	21	26.38	2	56.63	227	554	112.9	17	242.76	939	2371
gedit	19	20.84	2	69.34	306	396	103	4	267.13	1153	1957
gnome_control_center	35	21	4.00	51.753	296	735	69.54	19	125.13	527	2434
gnome_games	14	43.57	7	83.15	304	610	178.93	13.5	341.49	1164	2505
gnome_media	23	21.87	5	36.67	130	503	39.22	6	84.03	345	902
gnome_power_manager	7	3.43	4	1.51	6	24	8	2	11.4	32	56
gnome_screensaver	4	15.75	10	15.84	39	63	211.5	3.5	417.67	838	846
gnome_system_tools	22	17.14	3	33.42	154	377	92.59	24	228.27	1043	2037
ibsoup	3	28.33	8	39.63	74	85	219.67	8	370.09	647	659
metacity	10	14.8	6	23.85	77	148	184.2	7	270.12	600	1842
nautilus	136	49.95	8.5	225.14	2475	6793	86.63	13	220.1	1712	11782
seahorse	2	73.5	73.5	101.12	145	147	198	198	275.77	393	396

atively small, but varying, percentage[4] of developers who are involved in both activities. The

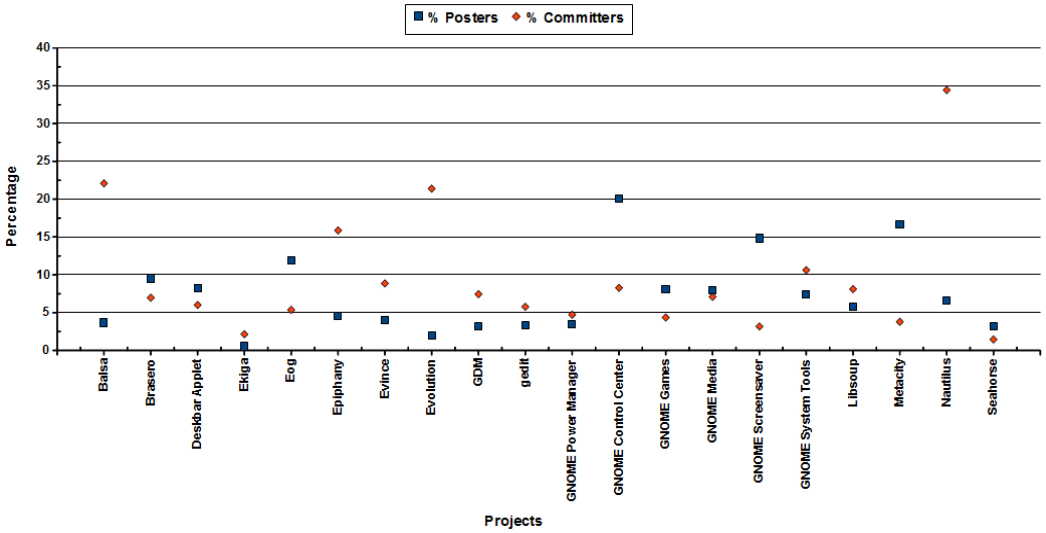

Figure 5: Percentage of developers involved in posting and committing.

percentage of developers in each activity varies across the projects. For example, the Ekiga, Gnome Power Manager, and Seahorse projects having less than 5% of their developers committing to SVN and at the same time posting messages to their respective projects' mailing lists. Projects such as Balsa and Nautilus have few poster (3.68% and 3.23%), but higher percentage of committers (22.1% and 34.43%).

[4] Calculated as: *% posters = (N_dev/N Poster)*100* and *% committers = (N_dev/N committers)*100*

4.1.1 Are developers making more commits than posts?

Hypothesis [H1]: *FOSS developers make more commits to a project's code (SVN) repository than they are posting messages to mailing lists.*

In our investigation of **H1**, for each project, we compared the total number of commits made to SVN with the total number of messages developers posted to the mailing lists. The pattern of contribution for all the 502 developers in the 20 projects is shown in the boxplots in figure 6. In the boxplots, the median line and error T-bar widths for each set of project data (nposts and ncommits) are shown. The domination of SVN commits, with larger means of commits per developer (mean = 150.32, Std. deviation = 424.986) over posts (mean = 42.63, Std. deviation = 161.852) is evident in all the projects.

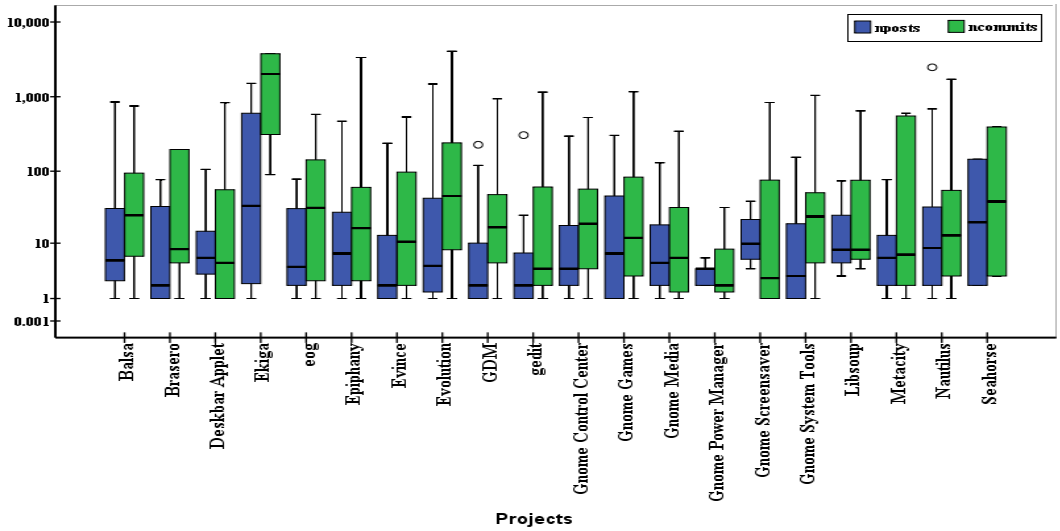

Figure 6: Distributions of posts and commits for all projects (y-axis in log scale).

4.1.2 Are developers contributing equally to SVN and mailing lists?

Hypothesis [H2]: *FOSS developers contribute equally to code repository and mailing lists.*

We used correlation between commits and posts to study how developers activities in SVN and mailing lists are related. The scatter plot in figure 7 shows the correlation between commits and posts in all projects. In the plot, data points are fitted to a line to show the trend in the commits and posting activities of the developers. Previous research ([SSL08]; page 414) showed that FOSS developers and users mailing lists activities have *fractal* or self-similarities properties and could best be explained by a polynomial model of third order, i.e. a cubic relation of the type

$$LogN = b0 + b1 * logr + b2 * (logr)^2 + b3 * (logr)^3 \quad (1)$$

As shown in figure 7, our fit method could explain 30.4% (r^3 = 0.304) of the variability in commits and posting activities. This translates to 26.5% or r^2 = 0.265 in linear terms. The linear

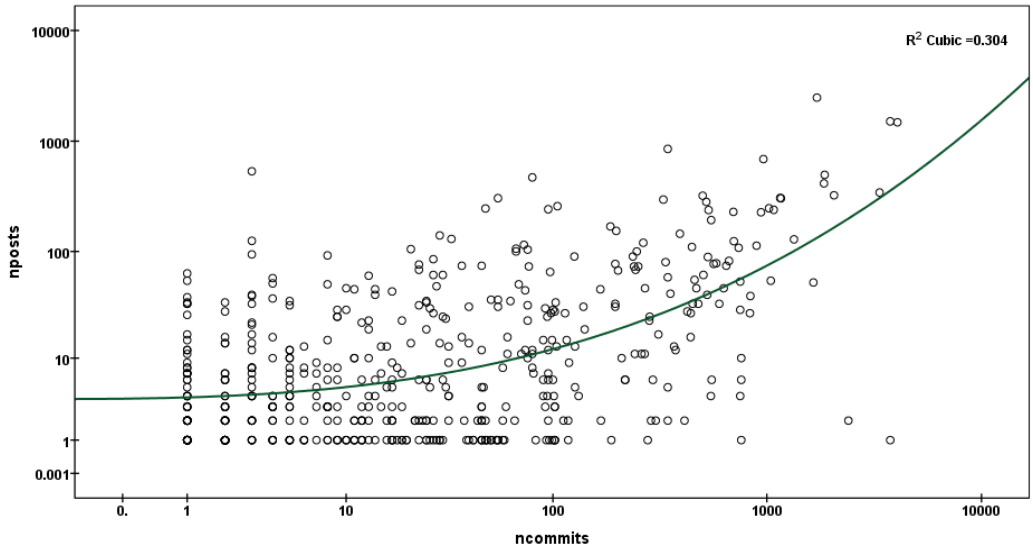

Figure 7: Relationship between posts and commits. *both axis in a log scale*.

association between *nposts* and *ncommits* as measured by Pearson correlation = 0.594, and this is significant at the 0.01 level (2-tailed) with $\rho = 0.000$. However, the *nposts* and *ncommits* data are not normally distributed and have outliers. Thus, nonparametric correlations using Spearman's *rho* and Kendall's *tau_b* statistics, which work regardless of the distribution of the variables [Nor04], are used to report the association between posts and commits. Table 5 shows that, overall, there is a low correlation between commits and posts, with Spearman's coefficient (ρ) = 0.426 (p =1.000).

Table 5: Correlations between posts and commits

			nposts	ncommits
Kendall's tau_b	nposts	Correlation Coefficient	1.000	.308
		Sig. (2-tailed)	.	.000
		N	502	502
	ncommits	Correlation Coefficient	**.308**	1.000
		Sig. (2-tailed)	.000	.
		N	502	502
Spearman's rho	nposts	Correlation Coefficient	1.000	**.426**
		Sig. (2-tailed)	.	.000
		N	502	502
	ncommits	Correlation Coefficient	.426	1.000
		Sig. (2-tailed)	.000	.
		N	502	502

Furthermore, Wilcoxon signed ranks for the Two-Related-Samples Tests procedure was used

to compare the distributions of two variables. The results of the test in table 6 shows that for the 502 developers in the 20 projects; for 140 = ncommits < nposts, for 327 developer ncommits > nposts, and 35 developers had a balanced activity with ncommits = nposts.

Table 6: Ranks of developers contribution

Variable		$N_{dev.}$	Mean Rank	Sum of Ranks
ncommits -nposts	Negative Ranks	140	175.46	24565.00
	Positive Ranks	327	259.06	84713.00
	Ties	35		
	Total	502		

5 Concluding remarks

In this paper we have put forward research questions to investigate whether FOSS developers are making more commits to code repositories (SVN) than they are posting messages to mailing lists (Hypothesis 1), and whether developers should aim at a balanced activity by contributing equally there repositories (Hypothesis 2). Despite the fact that FOSS data is widely available and can be easily extracted [SASM07], this kind of research is made difficult because of the problem associated with integrating data from and the subsequent identification of developers from multiple repositories (SVN and mailing lists). We have presented and discuss a methodology which alleviates these empirical research obstacles. The methodology and algorithm enabled us to locate and count the quantitative contribution of FOSS developers in 20 GNOME projects.

An exploratory data analysis or EDA technique was used to show that each project has its unique characteristics and developers contribution to either coding or mailing lists can vary tremendously. In our data consisting of 12,851 posters and 4,762 committers who, respectively posted 95,331 email messages and made 140,665 commits, we found out that in 55% (N = 11) of the projects there are more developers as posters, with smaller means (post per poster), than committers, with larger means (commits per committer).

From this sample of posters and committers we are able to extract 502 developers who simultaneously contribute to both SVN and mailing lists. This cohort made more commits (mean = 152.1; Std. deviation = 431.171) than posts (mean = 43.19; Std. deviation = 164.353). However, this group accounts for a relatively small percentage of the overall developer community in each project. But a close examination of the percentage of developers involved in posting and committing shows that projects with small number of posters will also have a small number of committers. This is valid in 60% (N = 12) of the projects studied. There is a 50-50 split (20%; N=4 on either side) between projects with small percentage of posters but large percentage of committers (Balsa, Epiphany, Evolution, and Nautilus) and those with large percentage of posters but a smaller percentage of committers.

The analysis supports our first hypothesis (H1) that developers are making more commits to SVN (mean = 150.32, Std. deviation = 424.986) than they are posting messages to the developers mailing lists (mean = 42.63, Std. deviation = 161.852). Furthermore, a low but significant correlation (ρ = .0426; p = 1.000) between developers commits and posting were observed.

This moderately supports our second hypothesis (H2) that developers are contributing equally to code repositories and mailing lists. Wilcoxon signed ranks for the Two-Related-Samples Tests revealed that only 35 developers (less than 10%) had a balanced or tie activity.

The implications of these findings may provide assurance that FOSS developers, apart from coding and committing bits of code to a project's SCM, they are also involved in knowledge brokage [SAS06] in mailing lists. We can conjecture from earlier findings [SAS06, ARG06, Lon06] and our experience in both the FLOSSMetrics[5] and SQO-OSS[6] projects that this serendipity has implications for the quality of code since a large number of developers are externalizing and discussing their coding activities with other community members in the mailing lists. This kind of engagement may enable the developers to improve the quality of their code base, do more refactoring and learn about how the quality of the produced code may be improved.

Future work and research directions: As a follow up to this research, our future work aims at consolidating understanding developer dynamics and the development of appropriate community metrics [SC09] or indicators for the certification of both the FOSS product and process. Thus, narrowing the gap which exist in FOSS certification and formal methods [CS08]. Specifically, we plan to add a qualitative element to our research by interviewing some of the α [VTG+06] or star [SSL08] or key developers. This future work may also incorporate content analysis of the postings, new metrics like posts/commits and how such metrics vary overtime. This kind of data, metrics and commits analysis may help us better understand the quality of developers contribution, reveal any bottlenecks which may hinder the incorporation of developers code into the release product, and further reveal what kinds of metrics may be most appropriate when characterizing FOSS developers and projects.

Furthermore, in addition to SVN and mailing lists, developers also contribute intensively to the bug reporting and fixing process. Therefore, there exist an avenue of extending the methodology presented in this research to incorporate data from bug tracking systems data. This will provide a more comprehensive view of the pattern of developers contribution in open source projects. While the conclusion drawn from this study points out certain trends in Gnome projects, we are working on extracting a more heterogenous sample of projects and apply the same methodology to see if the patterns observed here can be generalized to other FOSS projects, not specifically Gnome based.

Acknowledgements: We are grateful to all partners in the FLOSSMETRICS project for providing access to the data and tools used in this study. We also wish to acknowledge the excellent suggestions we received from anonymous reviewers and participants at the 4th International Workshop on Foundations and Techniques for Open Source Software Certification (OpenCert2010). Their comments have helped us improve the paper greatly. The correspondence author wishes to acknowledge the Japan Society for the Promotion of Science (JSPS) who are currently sponsoring his research under Grant ID: P10807 and UNU-MERIT, Maastricht, Netherlands where this research began.

[5] http://www.flossmetrics.org/; Last visited: Tuesday, November 30, 2010
[6] http://www.sqo-oss.org/home; Last visited: Tuesday, November 30, 2010

Bibliography

[ARG06] J. Amor, G. Robles, J. Gonzalez-Barahona. Discriminating Development Activities in Versioning Systems: A Case Study. In *Proceedings PROMISE 2006: 2nd. International Workshop on Predictor Models in Software Engineering co-located at the 22th International Conference on Software Maintenance (Philadelphia, Pennsilvanya, USA)*. 2006.

[BGD+06] C. Bird, A. Gourley, P. Devanbu, M. Gertz, A. Swaminathan. Mining email social networks. In *MSR '06: Proceedings of the 2006 international workshop on Mining software repositories*. Pp. 137–143. ACM Press, New York, NY, USA, 2006.

[Bro75] F. Brooks. *The Mythical Man-Month. Essays on Software Engineering.* Addison-Welsey Publishing, 1975.

[CLM03] A. Capiluppi, P. Lago, M. Morisio. Characteristics of Open Source Projects. In *CSMR '03: Proceedings of the Seventh European Conference on Software Maintenance and Reengineering*. P. 317. IEEE Computer Society, Washington, DC, USA, 2003.

[CS08] A. Cerone, S. A. Shaikh. Incorporating Formal Methods in the Open Source Software Development Process. In *2nd International Workshop on Foundations and Techniques for Open Source Software Certification*. Milan, Italy, 10 September 2008 2008.

[DB05] T. T. Dinh-Trong, J. M. Bieman. The FreeBSD Project: A Replication Case Study of Open Source Development. *IEEE Transactions on Software Engineering* 31(6):481–494, 2005.

[DB07] J. M. Dalle, M. den Besten. Different Bug Fixing Regimes? A Preliminary Case for Superbugs. In Feller et al. (eds.), *Open Source Development, Adoption and Innovation*. IFIP International Federation for Information Processing 234, pp. 247–252. Springer, September 7-10 2007.

[GKS08] G. Gousios, E. Kalliamvakou, D. Spinellis. Measuring developer contribution from software repository data. In *MSR '08: Proceedings of the 2008 international workshop on Mining software repositories*. Pp. 129–132. ACM, 2008.

[Kuk06] G. Kuk. Strategic Interaction and Knowledge Sharing in the KDE Developer Mailing List. *MANAGEMENT SCIENCE 2006 52: 1031-1042.* 52:1031–1042, 2006.

[Lon06] J. Long. Understanding the Role of Core Developers in Open Source Development. *Journal of Information, Information Technology, and Organizations* 1:75–85, 2006.

[Mas05] B. Massey. Longitudinal analysis of long-timescale open source repository data. In *PROMISE '05: Proceedings of the 2005 workshop on Predictor models in software engineering*. Pp. 1–5. ACM, New York, NY, USA, 2005.

[MFH02] A. Mockus, R. Fielding, J. Herbsleb. Two case studies of open source software development: Apache and Mozilla. *Transactions on Software Engineering and Methodology.* 11(3):1–38, 2002.

[MFT02] G. Madey, V. Freeh, R. Tynan. The open source software development phenomenon: An analysis based on social network theory. In *Americas conf. on Information Systems (AMCIS2002)*. Pp. 1806–1813. 2002.

[Nor04] M. Norusis. *Statistical Procedures Companion*. Prentice Hall, Inc., 2004.

[QJ04] S. R. Q. Jones, G. Ravid. Information overload and the message dynamics of online interaction spaces: a theoretical model and empirical exploration. *Information System Research* 15 (2):194210, 2004.

[RG06] G. Robles, J. Gonzalez-Barahona. Contributor Turnover in Libre Software Projects. In Damiani et al. (eds.), *IFIP International Federation for Information Processing, Open Source Systems*. Volume 203, pp. 273–286. Springer,Boston, 2006.

[RGCH09] G. Robles, J. Gonzalez-Barahona, D. Cortazar, I. Herraiz. Tools for the study of the usual data sources found in libre software projects. *International Journal of Open Source Software and Processes* 1(1):24–45, Jan-March 2009.

[SAOB02] I. Stamelos, L. Angelis, A. Oikonomou, G. Bleris. Code Quality Analysis in Open-Source Software Development. *Information Systems Journal, 2nd Special Issue on Open-Source, Blackwell Science* 12 (1):43–60, 2002.

[SAS06] S. K. Sowe, L. Angelis, I. Stamelos. Identifying Knowledge Brokers that Yield Software Engineering Knowledge in OSS Projects. *Information and Software Technology* 48:1025–1033., 2006.

[SASM07] S. K. Sowe, L. Angelis, I. Stamelos, Y. Manolopoulos. Using Repository of Repositories (RoRs) to Study the Growth of F/OSS Projects: A Meta-Analysis Research Approach. In *Open Source Development, Adoption and Innovation*. IFIP International Federation for Information Processing 234/2007(978-0-387-72485-0), pp. 147–160. Springer Boston, August 2007.

[SC09] S. A. Shaikh, A. Cerone. Towards a metric for Open Source Software Quality. *Electronic Communications of the EASST* Volume 20: Foundations and Techniques for Open Source Certification 2009, 2009.

[Sek06] U. Sekaran. *Research Methods for Business: A Skill Building Approach*. Wiley, 4th edition, 2006.

[SFF+06] W. Scacchi, J. Feller, B. Fitzgerald, S. A. Hissam, K. Lakhani. Understanding Free/Open Source Software Development Processes. *Software Process: Improvement and Practice* 11(2):95–105, 2006.

[SII07] S. K. Sowe, G. S. Ioannis, M. S. Ioannis (eds.). *Emerging Free and Open Source Software Practices.* IGI Global, 2007.

[SO09a] SQO-OSS. Novel Quality Assessment Techniques. Deliverable Report-D7. Technical report, Software Quality Observatory for Open Source Software. Project Number: IST-2005-33331, 29 June 2009.
http://www.sqo-oss.eu/research/reports/SQO-OSS_D_7_final.pdf

[SO09b] SQO-OSS. Overview of the state of the art. Deliverable Report-D2. Technical report, Software Quality Observatory for Open Source Software. Project Number: IST-2005-33331, 29 June 2009.
http://www.sqo-oss.eu/research/reports/SQO-OSS_D_2_final.pdf

[SSL08] S. K. Sowe, I. Stamelos, A. Lefteris. Understanding Knowledge Sharing Activities in Free/Open Source Software Projects: An Empirical Study. *Journal of Systems and Software* 81(3):431–446., 2008.

[SSSA08] S. K. Sowe, I. Samoladas, I. Stamelos, L. Angelis. Are FLOSS developers committing to CVS/SVN as much as they are talking in mailing lists? Challenges for Integrating data from Multiple Repositories. In *3rd International Workshop on Public Data about Software Development (WoPDaSD). September 7th - 10th 2008, Milan, Italy.* 2008.

[VTG+06] S. Valverde, G. Theraulaz, J. Gautrais, V. Fourcassie, R. V. Sole. Self-Organization Patterns in Wasp and Open Source Communities. *IEEE Intelligent Systems* 21(2):36–40, 2006.

[ZPZ07] T. Zimmermann, R. Premraj, A. Zeller. Predicting Defects for Eclipse. In *PROMISE '07: Proceedings of the Third International Workshop on Predictor Models in Software Engineering*. P. 9. IEEE Computer Society, Washington, DC, USA, 2007.