

Impact of Stakeholder Type and Collaboration on Issue Resolution Time in OSS Projects

Anh Nguyen Duc¹, Daniela S. Cruzes¹, Claudia Ayala², and Reidar Conradi¹

¹ Norwegian University of Science and Technology, Department of
Computer and Information Science,
Trondheim, Norway

{anhn, dcruzes, conradi@idi.ntnu.no}

² Technical University of Catalunya, Department of Service Engineering
and Information Systems,
Barcelona, Spain

{cayala@essi.upc.edu}

Abstract. Initialized by a collective contribution of volunteer developers, Open source software (OSS) attracts an increasing involvement of commercial firms. Many OSS projects are composed of a mix group of firm-paid and volunteer developers, with different motivations, collaboration practices and working styles. As OSS development consists of collaborative works in nature, it is important to know whether these differences have an impact on collaboration between different types of stakeholders, which lead to an influence in the project outcomes. In this paper, we empirically investigate the firm-paid participation in resolving OSS evolution issues, the stakeholder collaboration and its impact on OSS issue resolution time. The results suggest that though a firm-paid assigned developer resolves much more issues than a volunteer developer does, there is no difference in issue resolution time between them. Besides, the more important factor that influences the issue resolution time comes from the collaboration among stakeholders rather than from individual characteristics.

1 Introduction

Open source software (OSS) development is a highly distributed and collaborative activity. In OSS projects, stakeholders, who are people involved in software development project such as developers, project leader, tester and end-users, collaborate with each other in various ways to accomplish development tasks. Although OSS was born as a movement mainly based on contributions of volunteer stakeholders, an increasing number of firms are getting involved in OSS projects [21][31]. Lakhani et al. found that around 40% of programmers are paid by companies to contribute to OSS projects [24]. Hars and Ou obtained similar results in a survey on the developers of the Linux kernel [29]. Consequently, many open source projects contain both types of stakeholder (firm-paid and volunteer), which have different motivations, collaboration practices and working styles. For instance, firm-paid developers contribute to the OSS community as part of their jobs, which provide

them a financial motivation. In addition, they often also work on proprietary software since it constitutes a part of the business model of their sponsor firm [2][9][25]. Therefore, they have to learn the community working style and adjust to the rhythms and the demands of OSS development [2]. In contrast, volunteer developers are usually motivated by social or technical reasons to demonstrate or improve their technical skills [9][25].

Several studies have investigated the potential differences among firm-paid and volunteer developers in OSS projects [2][21][24][29][31]. However, these studies did not address whether these differences actually have an impact on the OSS project outcomes such as quality of the source code, productivity of developers, activeness of the community and time to accomplish a software evolution task.

A software evolution task (or software issue) is normally referred as a unit of work to accomplish an improvement in the system. Dealing with a software issue includes fixing defects, implementing new feature requests and enhancing current system features. With a large amount of issues that occur from time to time, resolving them in a cost-effective manner is essential to achieve a high user satisfaction with less working effort.

Besides the impact of some special characteristics of stakeholders (in the issue resolving process, they are usually reporters and assignees), the issue resolution time can be influenced by a collaborative working process between reporters and assignees. Pinzger et al. mention the Coordination theory in OSS, which state that the interaction among stakeholders can impact software quality (such as mean time between failure) and work performance (such as defect removal effectiveness and problem fixing time) [30]. In the issue resolving process, stakeholders often use electronic media such as mailing list, IRC and issue tracking system to discuss, comment and clarify about an assigned task [23][26]. The collaboration among stakeholders, such as discussion, instruction and clarification on an issue, is important to the completion of the issue-resolving task.

This study has three main objectives. First, we characterize the difference in the average amount of resolved issues and issue resolution time between a volunteer assignee and a firm-paid assignee. To best of our knowledge, there is no study that empirically investigates the influence of volunteers versus firm-paid developers on issue resolution time. Second, we investigate collaboration among stakeholders in OSS projects by using Social network metrics and analysis. Last, we explore the impact of the collaboration measures on issue resolution time. While there are several studies using Social network metrics investigating software quality (as described in Section 2.1), this is among the first attempts to apply these metrics on studying issue resolution time.

The rest of the paper is organized as follows. Section 2 presents a construction of stakeholder collaboration measure using Social network analysis (SNA). While Section 3 states our hypotheses, Section 4 describes our case study and data collection procedure. Section 5 provides the hypotheses testing results. Section 6 discusses the findings and Section 7 identifies the threats to validity. The paper ends with a conclusion and future works.

2 Stakeholder Collaboration Measure by Social Network Analysis (SNA)

2.1 Impact of Collaboration on Software Development

Table 1 presents several studies exploring the impact of collaboration on software development outcomes. Bettenburg et al. studied the impact of social structure on software quality and find a statistical relation between a communication flow between

Table 1. Studies about collaboration

Studies	Dependent Variable	Collaboration Variable	Exploring Method	Test Results
Bettenburg et al. [6]	Number of post-released defects	Participant reputation (number of contributed messages)	Multiple linear regression model	Increase a predictive power of prediction model 11.66%
Abreu et al. [1]	Number of code changes	Number of messages in mailing list Number of messages from high-centrality-degree developers	Spearman's correlation	R = 0.1 to 0.45 p < 0.001 R = 0.06 to 0.16 p < 0.05
Bird et al. [7]	Post-released defect proneness	Developer-component network measures, e.g.: centrality degree	Release-cross Multiple Logistic regression	Recall: 0.705 to 0.859. Precision: 0.747 to 0.827
Wolf et al. [32]	Build failure likelihood	Developer-developer network measures, e.g.: density, centrality, betweenness and structural holes	Bayesian classifier	Recall:0.62, Precision: 0.75
Pinzger et al. [30]	Number of failure	Number of authors, number of commits, networks measures e.g.: Freeman centrality degree and betweenness	Spearman correlation Multiple linear regression model	R= 0.503 to 0.747, p<0.01 R ² = 0.698 to 0.746
Andrew et al. [5]	Vulnerability of software files	Betweenness measures, number of developers and number of commits	Mann-Whitney-Wilcoxon (MWW) test	Higher values for vulnerable file, p<0.0001
Feczak et al. [14]	Bug fixing time	Stakeholder network measures, e.g: Freeman centrality degree	Spearman correlation	R = 0.13 to 0.35 p <0.05
Anbalagan et al. [2]	Defect resolution time	Number of unique participants	Spearman correlation	R = 0.22 p < 0.0001
Guo et al. [16]	Likelihood of fixed defect	Defect opener reputation, number of defect report editors and assignees	Chi square test Correlation test	p < 0.0001 Not reported

developers and users and post-release defects [6]. Abreu et al. investigated Eclipse sub-projects and found a significantly positive correlation between communication frequency between developers and number of injected defects in the software [1]. Bird et al. showed that a socio-technical network of software modules and developers is able to predict software failure proneness with greater accuracy than other prediction methods [7]. Wolf et al. formed a developer-task network to explore the impact of developer communication on software build integration fail [32]. Pinzger et al. constructed a developer-module network to predict the software failures [30].

More relevant to our focus are studies about relationship between developer collaboration and defect fixing time. Feczak et al. empirically validated the Coordination theory in open source projects and found that collaboration among stakeholders, measured by social network metrics, has a positive influence on software defect fixing time [14]. Anbalagan et al. also found a significant correlation between number of participants in editing a defect report and median time taken to correct it [2]. Guo et al. used collaboration measures to predict which defect will get fixed in Windows 7 and concluded that the defects that have more people involved in defect report editing will be more likely to be fixed [16]. While these studies show that developers collaboration, measured by a developer-artifact network metrics is useful for predicting software defects and fixing time, a similar approach can be applied to discover the impact of developers collaboration on issue resolution time.

2.2 Issue-Stakeholder Network Measures

Social network analysis (SNA) considers social relationships in term of network theories, which focus on social nodes, such as people, groups, organizations and measures relationships and information flows among them [15]. In this study, we construct an undirected graph to represent a network of issue-stakeholders. The graph employs two types of nodes: stakeholders and issues. Stakeholders include a *reporter* (who reports the issue), an *assignee* (who is assigned to resolve the issue) or a *commenter* (who comments or discuss about the issue). A link occurs only between a stakeholder and an issue, which represents for a stakeholder's action on the issue, such as an issue report, a report update, a comment on the issue and an issue assignment.

To establish the issue-stakeholder network, we use a social network analysis tool, namely ORA¹. The most common measure in SNA is centrality, which denotes the structural power position of a node in a given network. There are three centrality measures in SNA, namely Freeman Degree Centrality, Closeness and Betweenness. In the scope of this study, we investigate Freeman Centrality Degree since this metric is successfully applied in relevant studies [14][30][32]. In our network, the Freeman Degree Centrality of an issue represents the number of unique stakeholders that involve in the issue. For each issue, the high value of a centrality degree shows a large number of stakeholders working on it (reporting, commenting or resolving it). The centrality degree of an issue is calculated as in Formula 1:

¹ <http://www.casos.cs.cmu.edu/projects/ora/>

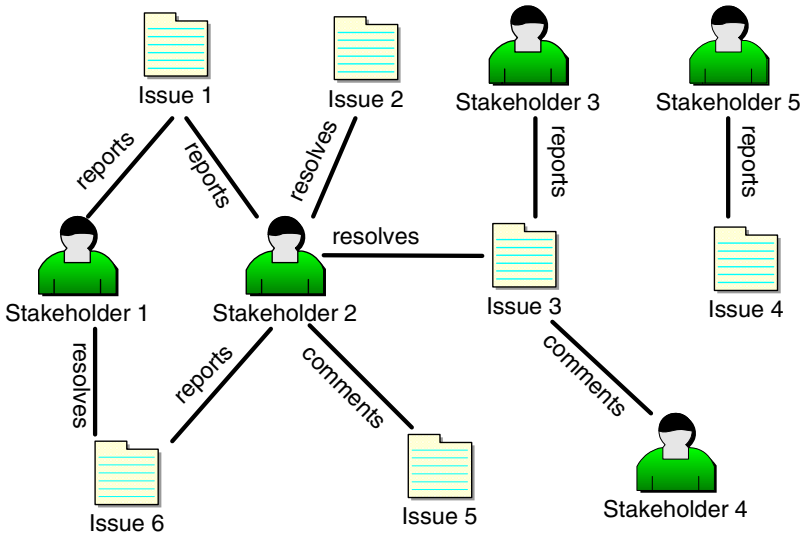


Fig. 1. Issue-stakeholder network in issue resolution

$$Gd(i) = \frac{d(i)}{n-1} \quad (1)$$

with $d(i)$ is the node degree of a issue,
 n is the total number of stakeholders and issues

Similarly, the Freeman Degree Centrality of a stakeholder is the number of issues directly linked to the stakeholder. We also want to explore whether stakeholder centrality has an impact on issue resolution time. For each issue, we calculate the accumulative stakeholder centrality degree (C_s) as a sum of centrality degrees of all involved stakeholders, as in Formula 2:

$$C_s(i) = Gd_{ass}(i) + Gd_{rep}(i) + \sum Gd_{com}(i) \quad (2)$$

with $Gd_{ass}(i)$, $Gd_{rep}(i)$ and Gd_{com} is the centrality degree of assignee, reporter and commenter correspondently.

The meaning of $C_s(i)$ is that the issue is important when they are resolved by many stakeholders and by important stakeholders, who involved in many other issues. Illustrated by Figure 1, the Freeman centrality degree of Stakeholder 2 is $5/11$ and the degree of Stakeholder 3 is $1/11$, which shows that Stakeholder 2 involves in more issues than Stakeholder 3 does. Issue 3's centrality degree is $3/11$ and Issue 4's centrality degree is $1/11$, which shows that Issue 3 is involved by more stakeholders than Issue 4 is. The accumulative stakeholder centrality degree of Issue 3 is $7/11$.

3 Research Hypotheses

In our context, a firm-paid stakeholder is an assignee or a reporter who works for a commercial company that uses and contributes to the development of an OSS project. We observe that many firm-paid assignees are also core contributors in developing the OSS product. While these core project members have significant contributions in developing the software [12], it is interested to know whether they also significantly contribute to resolving issues in the software evolution phase. Therefore, our first hypothesis is that:

***H1:** The stakeholder's centrality degree of a firm-paid assignee is higher than those of volunteer assignee. (Null hypothesis: there is no difference in distribution of stakeholder centrality degrees between firm-paid and volunteer assignees).*

Since firm-paid assignees also include the core members of the projects, they are supposed to have more knowledge and experience in developing the OSS product than peripheral members do [12]. Therefore the resolution time should be different between the group of volunteer assignees and the group of firm-paid assignees. Our second hypothesis is that:

***H2:** There is a difference in mean issue resolution time between a firm-paid assignee and a volunteer assignee. (Null hypothesis: there is no difference in mean issue resolution time between firm-paid and volunteer assignees).*

An issue with many stakeholders involved might relate to many different software modules or different development tasks. Therefore, the complexity of such issues is higher and thus, it takes the assignee longer time to resolve. Our third hypothesis is that:

***H3:** The larger number of stakeholders involve in an issue is, the longer the issue resolution time is. (Null hypothesis: there is no correlation between the number of stakeholders involved in an issue and the issue resolution time).*

A large number of comments and discussions on an issue may be caused by problems on the issue description (which leads to confusion or dissensus among stakeholders) or by the complexity of the resolving task and could lead to longer resolution time. Our last hypothesis is that:

***H4:** The larger number of exchanged messages on an issue is, the longer the issue resolution time is. (Null hypothesis: there is no correlation between the number of message exchanged in an issue and the issue resolution time).*

4 The Case Study

4.1 Projects Context and Selection

Three OSS projects were selected for our study, namely Qt, Qpid and Geronimo. The reasons for selecting these projects were: (1) these projects are active and ongoing for at least 4 years, which ensure the scale of the datasets; (2) there are similar issue

tracking system used in these projects, which facilitate the data collection; (3) these projects are similar in business domain and technical level, thus reducing the variability of the results, and, (4) these projects are significantly influenced by firm-paid developers, which enable the investigation of the impact of different stakeholder types.

Qt is an Open Source cross-platform framework developed by Qt Development Framework (Nokia) based on the programming language C++. The framework offers common components such as networking, OpenGL, multimedia and a widget toolkit². **Qpid** is an cross-platform Open Source enterprise messaging system developed around the open standard Advanced Message Queuing Protocol (AMQP). It is implemented in many programming languages, such as: C++, C#, Python, Ruby and Java³. The project originated from a joint venture mostly consisting of code by Red Hat, Iona and JP Morgan. **Geronimo** is a server runtime framework that pulls together the Open Source alternatives to create runtime instances that meet the needs of developers and system administrators and open-source, Apache-licensed⁴. The project originated from IBM developers.

4.2 Data Collection and Preprocessing

All software issues were collected from JIRA repositories⁵ of the respective projects. The summary of datasets was described in Table 2, with the main, owner firm of each project, the time frame of the issues collected for analysis, the total number of issues, number of stakeholders (assigned developers and issue reporters, who collaborated with the project during this period), the total number of issues in the repository and the total number of issues that we used for our analysis.

Table 2. Issue collection from cases study

Info.\ Projects	Qt	Qpid	Geronimo
Main Firms	Qt (Nokia)	Red Hat, JP Morgan	IBM
Time Frame	11/03-12/10 (85 months)	9/06-12/10 (51 Months)	8/03-12/10 (87 Months)
Number of Stakeholders	1568	126	405
Number of issues	16818	3016	5697
Number of selected issues	9921	2278	4787

Issue resolution time was computed by using the *created time* field and the *issue resolved time* field. We excluded 3514 issues that are not possible to calculate the issue resolution time. We removed 2171 issues that have the state OPEN, DUPLICATE or INVALID. We also deleted 2838 issues that do not have *reporter* or *assignee* information (stated as *unassigned* or *unknown*), and issues with invalid

² Qt project - <http://qt.nokia.com/>

³ Qpid project - <http://qpid.apache.org/>

⁴ Geronimo project - <http://geronimo.apache.org/>

⁵ JIRA—bug, issue and project tracking system, <http://www.atlassian.com/software/jira/>

stakeholder information (as described below). Twenty-two data points were also taken out by an outlier detection function implemented in the R⁶ package. At the end of the data preprocessing procedure, 16986 issues were selected for further analyses, which consumes 67% of total number of issues.

The classification of stakeholder type (firm-paid or volunteer) was manually executed by searching stakeholder name and professional information in the Internet. The first information source is the list of contributor and mailing list from the project repository. We found these stakeholders with explicit company information, either as project initiators or main contributors of the open projects. With stakeholders that company information was not given in the project site, we determined the affiliation by: (1) the stakeholder's profile from social networking site such as Facebook, LinkedIn and personal blogs, and (2) the stakeholder's email with a private company domain. The stakeholder company information were extracted by the time when the stakeholder worked in the OSS project. We assumed that the group of stakeholders (more than three) come from the same company participate in the OSS project as a company representative and are paid by the company. The stakeholders without any identified company information were classified as volunteers.

After collecting stakeholder information, we synchronized the stakeholder name and alias to avoid replicated data. Table 2 describes the total number of stakeholders that involve in the OSS projects in the time period that data are collected. Collaboration information was extracted from issue tracking systems and the mailing lists of OSS projects using a Perl script. For each issue, we collected comments, edits on the issue report and issue-related messages from the project mailing list.

4.3 Descriptive Statistics

Table 3 presents the distribution of reported issues by stakeholder types in Qpid, Geronimo and Qt correspondingly. As our expectation, stakeholders from Redhat and JP Morgan in Qpid (53.6% of reported issues) and stakeholders from IBM in Geronimo (60.8% of reported issues) are the main contributors in reporting issues. However, the largest amount of reported issues in Qt comes from volunteer reporters (44.9% of reported issues). This observation can be explained by the large amount of end-users involved in the Qt project, who directly report their problem, in the issue project tracking system. Table 4 shows the distribution of resolved issues by different stakeholder types. Not surprisingly, most of the issues are resolved by developers from the main firms such as Redhat and JP Morgan (62.4% of resolved issues) in Qpid, IBM (71.6% of resolved issues) in Geronimo and Nokia (62% of resolved issues) in Qt.

Figure 4 presents box plot charts of issue centrality and issue-based messages in the three projects. In Figure 4a shows that most of issues are touched by one to three stakeholders, other than the reporter. In Figure 4b, the average number of issue-based messages is similar among three projects. We see that number of message exchanged around an issue in three projects is from none to four messages, slightly vary among projects.

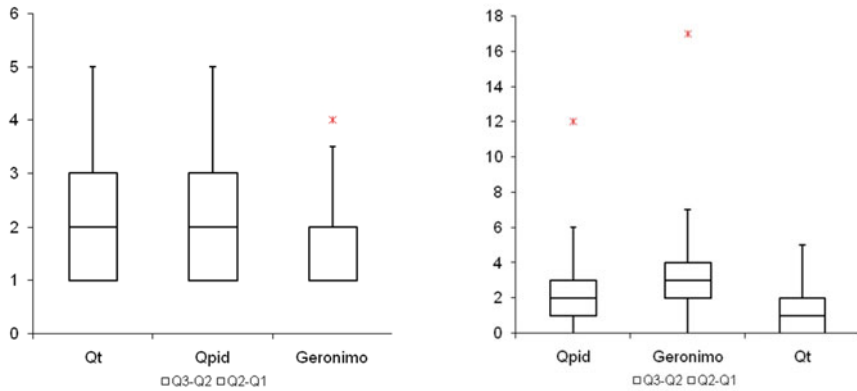
⁶ The R Project for Statistical Computing - <http://www.r-project.org/>

Table 3. Distribution of contribution in reporting issue

Issues from	Qpid	Geronimo	Qt
Individual	453 (19.9%)	1205 (25.0%)	4452 (44.9%)
Other company	605 (26.5%)	683 (14.2%)	1124 (11.3%)
Main Firms	1220 (53.6%)	2919 (60.8%)	4345 (43.8%)
Total	2278 (100%)	4787 (100%)	9921 (100%)

Table 4. Distribution of contribution in resolving issue

Issues from	Qpid	Geronimo	Qt
Individual	252 (11.1%)	401 (8.4%)	2463 (24.8%)
Other company	604 (26.5%)	956 (20.0%)	1315 (13.2%)
Main Firms	1422 (62.4%)	3420 (71.6%)	6143 (62.0%)
Total	2278 (100%)	4787 (100%)	9921 (100%)

**Fig. 4a, b.** Descriptive of issue centrality and issue-based messages

5 Hypotheses Testing Results

5.1 H1: *The stakeholder's centrality degree of a firm-paid assignee is higher than those of a volunteer assignee.*

Due to the fact that stakeholder centrality degrees are not normally distributed as observed from histogram and descriptive statistics, we used Wilcoxon rank-sum test [13].

Table 5. Resolution time by volunteer vs. firm-paid assignees

Projects	Median centrality of Firm-paid	Median centrality of Volunteer	Significance level
Geronimo	0.0169	0.0049	p=0.0014
Qpid	0.0114	0.0057	p=0.0251
Qt	0.0131	0.0024	p=0.0014

All the tests are performed using the statistic package R with $\alpha = 0.05$. The null hypothesis H1, which stated that there is no difference in stakeholder centrality degree between firm-paid and volunteer assignee was investigated with a one-tail test. The results are shown in Table 5. In all cases, the median values of centrality degree in the firm-paid groups are significantly higher than those in the volunteer groups. In particular, the number of issues involved by a firm-paid stakeholder is at least two times higher than ones involved by volunteer stakeholder in all projects. The p-values in all tests allow us to reject the null hypotheses in all projects. We accept the alternative hypothesis that the centrality degree of firm-paid stakeholders is higher than one of volunteer stakeholders.

5.2 H2: There is a difference in mean issue resolution time between a firm-paid assignee and a volunteer assignee

The distribution of issue resolution time between firm-paid assignee and volunteer assignee is shown in Figure 5. From the graph, we notice that the difference between these two groups in Qt and Qpid is very small. In Geronimo, there is a slightly higher difference in distribution of issue resolution time between firm-paid and volunteer assignee, but the high standard deviation could make this insignificant. To test whether there is a difference in issue resolution time between firm-paid and volunteer assignees, we also used the Wilcoxon rank-sum test.

The null hypothesis H2, which stated that there is no difference in issue resolution time between firm-paid and volunteer assignee was investigated with a two-tail test. The results are shown in Table 6. We observed that in three cases, the test with Geronimo data revealed a significant difference in resolution time between two groups while those with Qt and Qpid data did not. Therefore, the null hypothesis was rejected only in Geronimo dataset at significance level 95%. In Qpid and Qt, we accept the assumption of the null hypothesis.

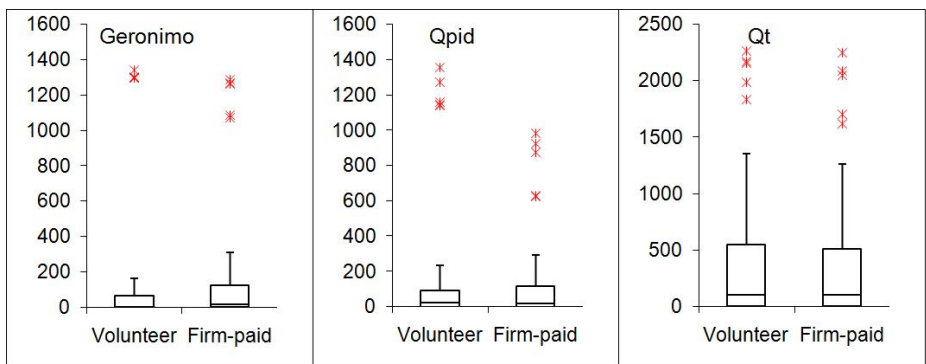


Fig. 5. Issue resolution time (days) between volunteer and firm-paid stakeholder

Table 6. Resolution time of volunteer vs. firm-paid assignees

Projects	Median resolution time by Firm-paid	Median resolution time by Volunteer	Significance level
Geronimo	10	18	p= 0.0000
Qpid	23	17	p= 0.1653
Qt	102	101	p= 0.4911

5.3 H3: The larger number of stakeholders involve in an issue is, the longer the issue resolution time is, and H4: The larger number of exchanged message on an issue is, the longer the issue resolution time is

We performed a pair-wise correlation analysis among number of message, issue centrality degree, sum of stakeholder centrality and issue resolution time. The correlation matrixes for Qt, Qpid and Geronimo projects are shown in Table 7, Table 8 and Table 9 respectively. The mark “**” represents a significance level at 0.01. Referring to Hopskin interpretation of value of correlation coefficient, which classify the value of correlation coefficient as trivial (<0.1), minor (0.1 – 0.3), moderate (0.3-0.5), large (0.5 – 0.7), very large (0.7 – 0.9) and almost perfect (0.9 - 1.0) [22], the correlation between number of task-based messages and issue resolution time is significant at minor level in Qt, Qpid while it is at moderate level in Geronimo. The correlation between issue centrality and its resolution time is at a minor level for Qt and at a moderate level for Qpid and Geronimo. Besides, the correlation coefficient between stakeholder accumulative centrality and resolution time is slightly higher than the one of issue centrality. All of these correlation coefficients are significant at level 0.01, which allow us reject the null hypotheses for H3, H4 and accept the alternative ones. It is noticed that among three variables, the accumulative stakeholder centrality degree has the largest correlation coefficient with issue resolution time in all projects.

Table 7. Pairwise correlation for Qt

	No of message	Issue centrality	Sum. Stak. centrality	Resolution time
Number of message	1	0.413**	0.460**	0.125**
Issue centrality		1	0.213**	0.172**
Sum. Stak. centrality			1	0.262**
Resolution time				1

Table 8. Pairwise correlation for Qpid

	No of message	Issue centrality	Sum. Stak. centrality	Resolution time
Number of message	1	0.569**	0.423**	0.243**
Issue centrality		1	0.199**	0.310**
Sum. Stak. centrality			1	0.331**
Resolution time				1

Table 9. Pairwise correlation for Geronimo

	No of message	Issue centrality	Sum. Stak. centrality	Resolution time
Number of message	1	0.491**	0.382**	0.416**
Issue centrality		1	0.251**	0.303**
Sum. Stak. centrality			1	0.409**
Resolution time				1

6 Discussion of Results

Table 10 summarizes the testing results for each hypothesis. Concerning **hypothesis H1**, the statistical test results reject the null hypotheses in all cases, which show the centrality degree of an average firm-paid assignee are significantly higher than that of an average volunteer assignee. This finding infers the distribution of labor between firm-paid and volunteer assignees. It indicates that in the issue-resolving process, a firm-paid assignee involves in much more issues than a volunteer assignee does.

On testing **hypothesis H2**, the issue resolution time significantly varies between firm-paid assignees and volunteer assignees in only one out of three investigated projects. Therefore, we can conclude that the stakeholder type is unlikely an influenced factor on issue resolution time. The data suggests that while volunteer and firm-paid assignees participate in OSS projects with different motivation and working approaches, these differences do not have an impact on their issue resolution time.

In the result for **H3 and H4**, the correlation tests reveal a positive correlation between collaboration measures, such as number of message, number of involved stakeholder and issue resolution time. It implies that the high collaboration level in an issue, e.g. high number of messages exchanged or high number of involved stakeholders indicates a longer resolution time. This may be due to the complexity of the task that relates other issues or software modules; or the poor quality of the issue description leads to demands of explanation and discussion. However, we are aware that the result of correlation analysis doesn't imply cause-effect relationship due to the effect of compounding factors. To validate the provided hypothesis, a further regression analysis is necessary. From the results, we also observe that there is significant positive correlation between issue centrality and number of messages exchanged. This observation was expected as the larger number of stakeholders involved in an issue (i.e. editing the reports or commenting on the issue) clearly leads to the increasing of number of comments or report edits. Therefore, these two variables should be checked for compounding factors if they are both used in regression models.

Table 10. Results of Hypotheses testing

Hypotheses	H1	H2	H3	H4
Test	Mann Whitney U	Mann Whitney U	Spearman correlation	Spearman correlation
Geronimo	Accept	Accept	Accept	Accept
Qt	Accept	Reject	Accept	Accept
Qpid	Accept	Reject	Accept	Accept

7 Threats to Validity

First, a major threat of the study validity lies in the division of stakeholders as volunteer or firm-paid. Although a major amount of stakeholder's affiliation is identified, there are still some stakeholders with no company information. However this group of unidentified stakeholders is responsible for a very small portion of issues in general. Since the major portion of the issues comes from identified stakeholders, the comparison of resolution time between groups of stakeholders would not be significantly influenced.

Second, a main concept investigated in this study is collaboration, which is measured by the number of comments, messages and number of issue-involved stakeholders. Although collaboration between stakeholders can be done via other channels, such as IRC, Skype and face-to-face discussion, issue tracking system and mailing list are the most common discussion means. The most relevant discussion about an issue should be found here. The other concern in the data collection process is the quality of the issue report since the data can be randomly filled in and the occurrence of duplicated reports. However, the quality of report is also an included factor in this study since it might influence the issue resolution time.

Third, another threat to validity comes from the generality of the research findings. As in many empirical studies of OSS projects, few case studies are definitely not significant enough to generalize what we found to the population of OSS projects. In this study, the cases were thoroughly selected to represent for an active, medium-size and on-going OSS projects.

Last but not least, compounding factors is an unavoidable threat in a correlational study. The high correlation between number of messages, number of stakeholders and issue resolution time can be caused by a latent variable, not investigated in this study, such as complexity of the issue, or dependencies among issues. Therefore, this concern could be a subject for a future investigation.

8 Conclusion and Future Work

In this study, we investigated the impact of different stakeholder types and their collaboration on issue resolution time in three medium-size and ongoing OSS projects. The statistic test result provides some interesting findings for OSS practitioners as well as OSS researchers. First, we found that in firm-involved OSS projects, there is not only a large portion of firm-paid labor contributed to the projects, but also a higher workload on an average firm-paid assignee than on a average volunteer assignee. However, we did not find a difference in issue resolution time between volunteer and firm-paid assignees. The result contributes to the understanding of distribution of workload and resolving time between volunteer and firm-paid assignees. Second, we found a significant impact of stakeholder collaboration on issue resolution time. Particularly, the issue with fewer stakeholders is resolved faster than the one with more stakeholders. The issue with fewer comments is also resolved faster than the ones with more comments. For practitioners, these metrics can be integrated in the issue tracking system or defect repository to provide a recommendation for issue resolving process. Particularly, the

collaboration information collected overtime will help developers being aware of which issue is going to take longer time to resolve. For researchers who want to integrate collaboration measures in software quality or productivity prediction models, they should consider of not only the usefulness of number of involved stakeholders, number of exchanged messages but also the compounding effect between them.

The paper contributes to fill in a gap in the literature gap by providing an empirical investigation of firm-paid stakeholders and their cooperation with others in OSS projects. The findings were supported by descriptive statistic and correlation analysis and further work should employ regression analysis to validate these findings. The study is also limited in using simple SNA metrics, such as the Freeman Centrality Degree. In future, we will explore more SNA metrics to investigate other aspects of stakeholder collaboration. Besides, the findings are based on only three projects, so the analysis should be replicated with more datasets to generalize conclusions on OSS community.

Acknowledgements. The authors would like to thank Tor Stålhane for his valuable comments and help with checking the statistical procedures.

References

- [1] Abreu, R., Premraj, R.: How developer communication frequency relates to bug introducing changes. In: Proceedings of the Joint International and Annual ERCIM Workshops on Principles of Software Evolution (IWPSE) and Software Evolution (Evol) Workshops, pp. 153–158. ACM, Amsterdam (2009)
- [2] Anbalagan, P., Vouk, M.: On predicting the time taken to correct bug reports in open source projects. In: Proceedings of IEEE International Conference on Software Maintenance (ICSM 2009), pp. 523–526 (2009)
- [3] Ayala, C.P., Cruzes, D., Hauge, Ø., Conradi, R.: Five Facts on the Adoption of Open Source Software. *IEEE Software* 28, 95–99 (2011)
- [4] Berdou, E.: Insiders and outsiders: paid contributors and the dynamics of cooperation in community led F/OS projects. In: *Open Source Systems*, pp. 201–208. Springer, Boston (2006)
- [5] Bettenburg, N., Hassan, A.: Studying the Impact of Social Structures on Software Quality. In: Proceedings of IEEE 18th International Conference on Program Comprehension (ICPC 2010), pp. 124–133 (2010)
- [6] Bettenburg, N., Just, S., Schroter, A., Weiss, C., Premraj, R., Zimmermann, T.: What makes a good bug report? In: Proceedings of 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESE 2008), pp. 308–318. ACM, Atlanta (2008)
- [7] Bird, C., Nagappan, N., Gall, H., Murphy, B., Devanbu, P.: Putting It All Together: Using Socio-technical Networks to Predict Failures. In: Proceedings of 20th IEEE International Symposium on Software Reliability Engineering (ISSRE 2009), pp. 109–119 (2009)
- [8] 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, 40–64 (2006)

- [9] Bonaccorsi, A., Lorenzi, D., Merito, M., Rossi, C.: Business firms' engagement in community projects. Empirical evidence and further developments of the research. In: Proceedings of 1st International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS 2007), May 21, pp. 1–5. IEEE Computer Society, Minneapolis (2007)
- [10] Capra, E., Francalanci, C., Merlo, F., Rossi-Lamastra, C.: Firms' involvement in Open Source projects: A trade-off between software structural quality and popularity. *Journal of System and Software* 84, 144–161 (2011)
- [11] Crowston, K., Scozzi, B.: Bug Fixing Practices within Free/Libre Open Source Software Development Teams. *Journal of Database Management (JDM)* 19(2), 1–30 (2008)
- [12] Crowston, K., Wei, K., Li, Q., Howison, J.: Core and Periphery in Free/Libre and Open Source Software Team Communications. In: Proceedings of 39th Annual Hawaii International Conference on System Sciences (HICSS 2006), vol. 06, p. 118. IEEE Computer Society, Los Alamitos (2006)
- [13] Devore, J.L.: Probability and Statistics for Engineering and the Sciences. *Technometrics* 46(4), 497
- [14] Feczak, S., Hossain, L.: Measuring Coordination Gaps of Open Source Groups through Social Networks. In: Proceedings of 11th International Conference on Enterprise Information Systems (ICEIS 2009), pp. 84–90 (2009)
- [15] Freeman, L.: The Development of Social Network Analysis. Empirical Press, Vancouver (2006)
- [16] Guo, P.J., Zimmermann, T., Nagappan, N., Murphy, B.: Characterizing and predicting which bugs get fixed: an empirical study of Microsoft Windows. In: Proceedings of 32nd ACM/IEEE International Conference on Software Engineering (ICSE), Cape Town, South Africa, vol. 1, pp. 495–504 (2010)
- [17] Hars, A., Ou, S.: Working for Free? - Motivations of Participating in Open Source Projects. *International Journal of Electronic Commerce* 6, 25–39 (2002)
- [18] Hauge, Ø., Ayala, C., Conradi, R.: Adoption of open source software in software-intensive organizations - A systematic literature review. *Journal of Information Software and Technology* 52, 1133–1154 (2010)
- [19] Hauge, Ø., Sørensen, C., Conradi, R.: Adoption of Open Source in the Software Industry. In: *Open Source Development, Communities and Quality*, pp. 211–221. Springer, Boston (2008)
- [20] Henkel, J.: Champions of revealing the role of open source developers in commercial firms. *Journal of Industrial and Corporate Change* 18(3), 435–471 (2009)
- [21] Herbsleb, J.D.: Global Software Engineering: The Future of Socio-technical Coordination. In: Proceedings of 29th International Conference on Software Engineering- Future of Software Engineering (ICSE), pp. 188–198 (2007)
- [22] Hopkins, W.G.: A scale of magnitudes for the effect statistics. *A New View of Statistics* (June 2002)
- [23] Kagan, S.L.: *United we stand: Collaboration for child care and early education services*. Teachers College Press, New York (1991)
- [24] Lakhani, K.R., Wolf, R.: *Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects*. In: *Perspectives on Free and Open Source Software*. MIT Press, Cambridge (2005)
- [25] Martinez-Romo, J., Robles, G., Gonzalez-Barahona, J., Perez, M.: Using social network analysis techniques to study collaboration between a FLOSS community and a company. *Open Source Development, Communities and Quality* 275, 171–186 (2008)

- [26] Mattessich, P.W., Murray-Close, M., Monsey, B.R.: Collaboration: What Makes it Work: A review of Research and Literature on Factors Influencing Successful Collaboration. Amherst H. Wilder Foundation (2001)
- [27] Meneely, A., Williams, L.: Secure open source collaboration: an empirical study of linus' law. In: Proceedings of 16th ACM Conference on Computer and Communications Security (CCS 2009), Illinois, USA, pp. 453–462 (2009)
- [28] Meneely, A., Williams, L., Snipes, W., Osborne, J.: Predicting failures with developer networks and social network analysis. In: Proceedings of 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2008), Atlanta, Georgia, USA, pp. 13–23 (2008)
- [29] Nguyen, T., Adams, B., Hassan, A.: Studying the impact of dependency network measures on software quality. In: Proceedings of IEEE International Conference on Software Maintenance (ICSM 2010), pp. 1–10 (2010)
- [30] Pinzger, M., Nagappan, N., Murphy, B.: Can developer-module networks predict failures?. In: 16th ACM SIGSOFT International Symposium on Foundations of software Engineering (FSE), Atlanta, Georgia, pp. 2–12 (2008)
- [31] Rahman, M., Ruhe, G.: Resource allocation and activity scheduling: bug fixing perspective, Technical Report, Software engineering decision support laboratory, University of Calgary (2010)
- [32] Wolf, T., Schroter, A., Damian, D., Nguyen, T.: Predicting build failures using social network analysis on developer communication. In: Proceeding of 31st International Conference on Software Engineering (ICSE), pp. 1–11 (2009)