The Emergence of Quality Assurance Practices in Free/Libre Open Source Software: A Case Study

Adina Barham

The Emergence of Quality Assurance Practices in Free/Libre Open Source Softaware: A Case Study

Adina Barham
Hitotsubashi University, Graduate School of Social Sciences, 2-1 Naka,
Kunitachi, Tokyo, 186-8601, Japan
adina.barham@yahoo.com

Abstract. As the user base of Free/Libre Open Source Software (FLOSS) diversifies, the need for higher quality is becoming more evident. This implies a more complex development model that includes various steps which were previously associated exclusively with proprietary development such as a formal quality assurance step (QA). However, little research has been done on how implementing formal quality assurance impacts the structure of FLOSS communities. This study aims to start filling this gap by analyzing interactions within such a community. Plone is just one among many FLOSS projects that acknowledged the importance of verification by implementing a quality assurance step.

Keywords: quality assurance, test, social network analysis, information flow

1 Introduction

A previous preliminary study [1] established that almost one third of the top 50 FLOSS software products ranked by number of downloads on www.ohloh.net had implemented explicit QA procedures. Furthermore, more than a quarter of the top 100 products ranked by number of user have some kind of QA. Verification, and more specifically quality procedures under the FLOSS development model has attracted a lot of interest within the academic community [2-9]. The structure of communities behind FLOSS has also been extensively researched. Studies focus on many community aspects such as structure and dynamics [10], communication patterns between core and periphery [11-12], or migration within the hierarchy of FLOSS projects [13]. However, little research has been done on how implementing formal QA affects the community. This research aims to start filling that gap by improving our understanding of how QA fits into the organizational structure of FLOSS communities.

adfa, p. 1, 2011.

A single open source project was chosen as a pilot case study in order to develop research questions that can then be applied in a wider comparative study of QA in open source projects. The Python-based content management system Plone was selected because it is a mature project (began in 1999) and because its development process includes a QA step [14]. The QA team has a dedicated webpage where one can find basic information such as activity description, communication channels and team leaders [15]. QA activities include triaging new bugs, validating submitted patches, ensuring that new releases are usable and generally help in the release process.

2 Research Questions

Q1: How is the QA layer included in the Plone community structure? We aim to find out how much contributors work only on QA and how much they work on other aspects of the project. Also, we ask how much peripheral members perform QA tasks. Previous research has approached the latter issue for Firefox and it has been shown that the percentage of periphery contributions is 20-25% [12].

Q2: What are the characteristics of activities performed by members of the Plone QA team? It is logical to draw the conclusion that some members will be more active than others but it would be interesting to investigate if members are equally active on all communication channels or if their tasks are limited to certain areas of the project.

Q3: How does the QA team communicate with other teams? Previous research has shown, that participants who have better access to information are able to contribute more efficiently [16] therefore interrupting the information flow might affect negatively the project's evolution. For this reason it is important to determine if there are any members that control the information flow. Due to the fact that social networks are in a continuous change [17], it might be useful to also establish the stages that the community went through before reaching its current state.

3 Data and Research Method

In order to measure QA activity levels, issue tracker data as well as mailing list data were taken into account. Data was retrieved in December 2012 – January 2013 and stored locally. The issue tracker data contained 13026 bugs with 55883 associated comments, and was downloaded using a web crawler. 29525 e-mails were downloaded from all the Plone mailing list archives that were parsable using MailingListStats [18]. In addition, a list of Plone contributors containing names and nicknames used in code repositories was downloaded from Ohloh.net [19].

Data from the QA mailing list which started in 2011 included 41 members of whom approximately 70% had sent only one e-mail at the time of data collection. Of the remaining 12 members only 1 had sent more than 10 e-mails, 4 were not active on other mailing lists and 7 were not listed as code contributors. The 4 members who

were not active on other mailing lists were also not listed as code contributors. However, their activity on the QA mailing list was low: none sent more than 5 e-mails. Therefore QA does not constitute a separate layer in the Plone community.

An interesting fact that can be observed from Fig.1 is that there is no correlation between time progression and activity levels. In addition, there seems to be no correlation between the number of bugs posted and the number of associated comments, which leads to the question the reasons behind these spikes in activity levels.

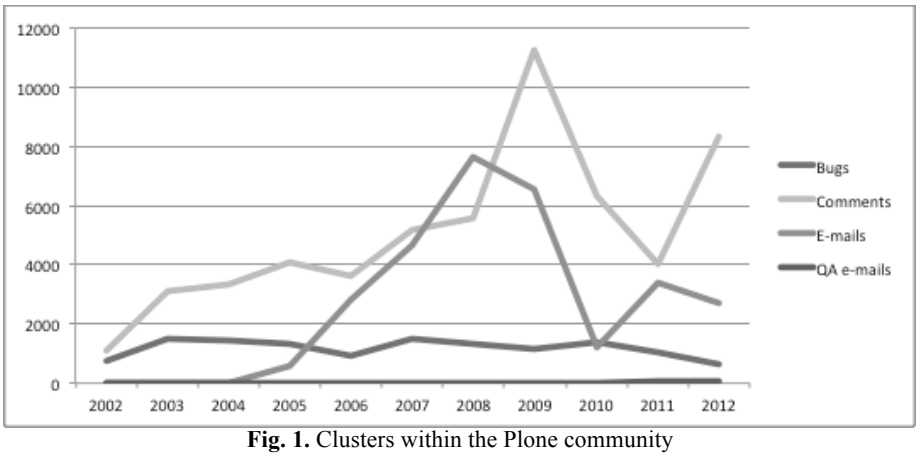

Fig. 1. Clusters within the Plone community

Social network analysis was used to analyse communication patterns within the Plone community. Project participants were represented as nodes (vertices) while interactions were represented as edges (arcs). The weight (value) of the graph's arcs represent the number of interactions between two members.

To create the network graph only authors who had replied to someone were taken into consideration. The next step consisted of eliminating loops or arcs starting and pointing to the same vertex. An additional reduction was performed in order to remove vertices that had no connections with other vertices. The resulted network contains 3414 vertices connected by a total of 16042 arcs of which 5093 have a value greater than 1. This means that 10949 connections (68%) are created by only one interaction. These members are occasional or peripheral contributors. From the remaining arcs 31% have values between 2 and 79 which means that arcs with values between 1 and 79 account for almost 99% of all arcs. The average degree is 9.39, which means that on average, a person interacts with approximately 9 other people. The network was then processed by transforming arcs into edges by summing up their values. Members who interacted with only one other member represented 35% of the whole community. 86% of community members interacted with a lower than average number of members (i.e. < 9).

To assess task distribution among community members the graph was divided into clusters. The cluster that contains QA mailing list members represents 1.06%

of the whole community, and contains a sub-cluster of members who do not contribute code accounting for 0.46% of the community.

Social networks are dynamic as they change their structure over time and for this reason it is important to consider time frames [20-21]. Social network analysis methods were applied using 6-month time frames to analyze the states through which the community went after dedicating a communication channel to the QA team. The size of the network varied between 226 and 746 vertices. The variation in size was expected considering the fact that many arcs were created after only one interaction. In addition the highest degree was 7.52 while the lowest was 4.29.

4 Conclusions

Q1: How is the QA layer included in the Plone community structure? Considering that the most active QA members also contributed code and were active on other mailing lists, QA is not a separate layer in the community. However, non-QA tasks might have been performed in different time frames than the ones in which members were part of the QA team; comparing time frames would allow us to clarify this. Furthermore, approximately 70% of QA mailing list members have sent only one 1 e-mail which might suggest that these are periphery members performing QA tasks. Only 30% of QA mailing list participants have not been active on other channels. In addition, members active on the QA mailing list account for only 1% of the whole community.

Q2: What are the characteristics of activities performed by members of the Plone QA team? An interesting phenomenon that occurs within the Plone community is the increase in activity levels that seems not to be linked to time progression. In addition it seems that the number of bugs opened does directly relate to the increase in comment activity levels. This suggests that there are other variables that influence activity levels.

Q3: How does the QA team communicate with other teams? The community seems to form a large component that spans both issue tracker and mailing list with the exception of few small sub-networks. This means that there is a lower risk for some members to control the information flow and to jeopardize the communication flow. However, there is a small group of people that are highly engaged in communicating with other members (86% of community members interacted with less than 9 other members – 9 being the average number of interactions a member has). In line with this conclusion, a large percentage of community members (68%) create links defined by only one interaction. This means that the rest of the community has somewhat stronger connections whereas a small percentage of users (1%) have very strong connections defined by more than 79 interactions. In addition, after analyzing the networks created using time frames, one could reach the conclusion that due to the drastic decrease in community size many participants were occasional contributors or in other words members of the periphery.

5 Limitations and Further Research

A number of limitations should be noted. First, only limited data cleaning was carried out. Second, it is possible that community members have used other communication channels than those listed on the relevant Plone websites. Third, it is possible that some members of the QA team did not actively participate in the mailing list. For these reasons, it would be desirable to conduct follow-up interviews with members of the community. These interviews would also shed light on the reasons for peaks in activity levels.

It could also be useful to re-run the community analysis using smaller time frames as 6 months may be a too big window for a community of this size. In addition, the community evolution could be analyzed using time frames covering the period before the formal adoption of QA in order to track down potential migration from one layer to the other.

A single case study cannot provide a recipe for success that can be applied to all FLOSS projects, but can be used to create hypotheses to be validated in future studies. Based on the findings of this paper the following hypotheses were formulated:

H1: The majority of the QA team members perform non-QA tasks as well.

H2: Approximately 80% of QA tasks are performed by a small percentage of the community.

H3: Increase in activity levels is not linked to time progression.

H4: Members performing QA are not an isolated layer in the community.

References

1. Barham, A.: The emergence of quality assurance in open source software development. In: Proceedings of the OSS 2011 Doctoral Consortium (2011)
2. Halloran, T. J., Scherlis, W. L.: High quality and open source software practices. In: 2nd Workshop on Open Source Software Engineering (2002)
3. Hedberg, H., Iivari, N., Rajanen, M., & Harjumaa, L.: Assuring Quality and Usability in Open Source Software Development. In: First International Workshop on Emerging Trends in FLOSS Research and Development, FLOSS'07, pp. 2-2 (2007)
4. Michlmayr, M., Hunt, F., Probert, D.: Quality practices and problems in free software projects. In: Proceedings of the First International Conference on Open Source Systems, pp. 24–28 (2005)
5. Schmidt, D. C., Porter, A.: Leveraging open-source communities to improve the quality & performance of open-source software. In: Proceedings of the 1st Workshop on Open Source Software Engineering (2001)
6. Chengalur-Smith I., Sidorova A., Daniel S.: Sustainability of Free/Libre Open Source Projects: A Longitudinal Study. In: JAIS 11 (2001)

7. Spinellis D., Gousios G., Karakoidas V., Louridas P., Adams P. J, Samoladas I., Stamelos I.: Evaluating the Quality of Open Source Software. In: Electronic Notes in Theoretical Computer Science, Volume 233, Proceedings of the International Workshop on Software Quality and Maintainability (2009)
8. Aberdour M.: Achieving Quality in Open Source Software. In: IEEE Software, pp. 58-64 (2007)
9. Zhao, L., Elbaum, S.: Quality assurance under the open source development model. In: Journal of Systems and Software - JSS, vol. 66, no. 1, pp. 65-75 (2003)
10. Crowston K., Howison, J.: The social structure of Free and Open Source software. In: First Monday, Vol. 10, No. 2 (2004)
11. Oezbek C., Prechelt, L., Thiel F.: The Onion has Cancer: Some Social Network Analysis Visualizations of Open Source Project Communication. In: Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development (FLOSS '10), pp. 5-10 (2010)
12. Masmoudi H., den Besten M. L., De Loupy C. and Dalle J. M.: 'Peeling the Onion': The Words and Actions that Distinguish Core from Periphery in Bug Reports and How Core and Periphery Interact Together. In: Fifth International Conference on Open Source Systems (2009)
13. Jensen, C., Scacchi, W.: Role Migration and Advancement Processes in OSSD Projects: A Comparative Case Study. In: Proceedings of the 29th international conference on Software Engineering, pp. 364-374 (2007)
14. http://plone.org/
15. http://plone.org/community/teams/qa-team
16. Aral, S., Brynjolfsson, E., Van Alstyne, M.: Productivity Effects of Information Diffusion in E-mail Networks. In: Proceedings of ICIS 2007 (2007)
17. Watts, D. J., A.: Twenty-first century science. In: Nature, Vol. 445, No. 7127, pp. 489-489 (2007)
18. https://github.com/MetricsGrimoire/MailingListStats
19. https://www.ohloh.net/
20. Howison, J., Wiggins, A., Crowston, K.: Validity Issues in the Use of Social Network Analysis for the Study of Online Communities. In: Journal of the Association for Information Systems (2012)
21. Christley S., Madey G.: Global and Temporal Analysis of Social Positions at SourceForge.net. In: The Third International Conference on Open Source Systems, IFIP WG 2.13, Limerick, Ireland (2007)