Quality Improvement in Volunteer Free Software Projects: Exploring the Impact of Release Management

Martin Michlmayr
Centre for Technology Management
University of Cambridge
Cambridge, CB2 1RX, UK
martin@michlmayr.org

***Abstract*— Even though free software has achieved great popularity and success in recent years, there are a number of product quality challenges facing the open source development model. There is significant room for further quality improvement and one area that deserves special attention is release management. This research will identify problems with current release practices, verify possible advantages of an increasingly popular release model, and develop interventions to improve release management in free software projects. The research also aims to answer the fundamental question as to how volunteer projects can deliver predictable and high quality software.**

***Index Terms*— release management, process improvement, quality assurance, volunteer projects, free software, open source.**

I. Background and Motivation

This doctoral research focuses on quality management in free software and open source projects. While some open source projects show a high level of quality, often on par or even surpassing that of closed source and proprietary software [4], [9], there are a number of unique challenges facing open source projects [7], [6]. One central question is how free software and open source projects can ensure a consistently high level of quality when many of the participants are volunteers whose involvement in a project continually changes in an unpredictable fashion [6].

This research aims to identify existing quality related problems in free software projects and to use the outcomes as a starting point for the development of quality improvement strategies. In exploratory interviews with a number of free software and open source developers, release management has been found to be a problematic area; this research will therefore focus on this topic in particular as one aspect of quality management.

There are several reasons as to why release management in distributed, volunteer free software projects may often be associated with problems. First of all, many of those who maintain software projects are programmers, who do not necessarily have the coordination and management skills required for release management [10]. Secondly, extra commitment from project participants is necessary during a release so that deadlines are met, but volunteers may not be able to put in more work than usual. Finally, there is often a dichotomy between the requirements of users and developers. Since developers mainly use development releases, they might not see the need for well tested and stable releases aimed at less technical and corporate users.

Inadequate release management can lead to a number of problems, such as software which is out of date, breaks compatibility, or does not meet the quality standards or the requirements of users. This research aims to identify such problems and good practices in open source projects, and develop further processes and techniques to improve release management in free software and open source projects. There will be a close interaction with the free software community to ensure that the outcomes of this research will be used by projects to improve release management.

II. Proposed Research

In interviews with twenty experienced free software and open source developers from a variety of projects, release strategies and processes, along with a number of problems with current release cycles, have been identified. Some developers described the advantages of a new release strategy, time-based release, over the more traditional feature-based release strategy. In time-based releases, a clear schedule is followed and a release is made according to a strict time plan, while feature-based releases are oriented towards the completion of certain features.

It has been suggested that time-based releases are particularly suited for large and modular projects because they allow individual developers to independently follow the schedule which has been set. This decreases the amount of coordination necessary for a release. Furthermore, predictable and steady releases seem to lead to greater motivation and faster development since developers know when their features must be ready in order to make the next release. According to this hypothesis, time-based releases offer a number of advantages to users: releases are more predictable, the development leads to more features and better code, and the release schedule allows for more systematic testing of the software.

This research will test whether time-based releases are actually associated with such advantages. The circumstances under which time-based releases should be chosen over more conventional release strategies will be studied. Following this, the question of how a project can successfully move to a predictable time-based release will be addressed. A number of open source projects that follow good release practices will be observed; interventions that can improve

Proceedings of the First International Conference on Open Source Systems
Genova, 11th-15th July 2005
Marco Scotto and Giancarlo Succi (Eds.), 309-310

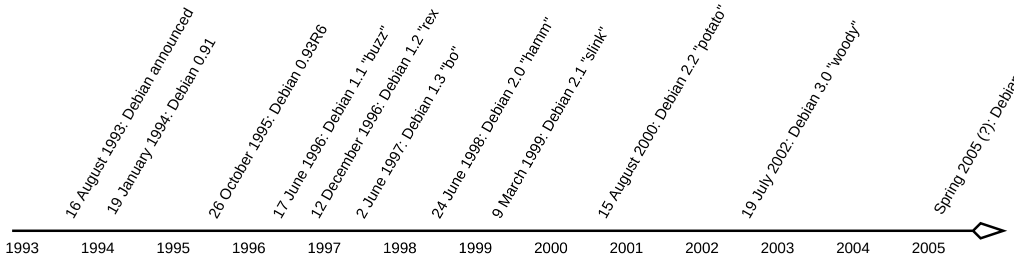

Fig. 1. Debian has experienced increasingly delayed and unpredictable releases in recent years

release management will then be proposed. These will subsequently be tested in live projects using action research.

III. Research Methods

Broadly speaking, the research can be subdivided into the following three phases which in turn employ particular research methods.

A. Identification of processes and problems

In this phase, current release processes and strategies will be investigated. Problems related to these processes will also be identified. This phase will mostly employ interviews, along with surveys, in order to obtain an in-depth input from a wide variety of sources, such as developers and end-users of free software and open source.

B. Investigation of time-based release strategy and testing of hypotheses

Based on the results from the first phase, hypotheses will be generated that will subsequently be tested. At the time of writing, there are a number of preliminary hypotheses that will be clarified and refined further before the research moves to phase two. This phase is characterized by a positivist approach in which hypotheses are tested in quasi-experiments. Comparable projects employing time and feature-based releases will be compared using empirical data. For these studies, a number of methods will be used to mine and analyze data. For example, a tool has been developed to reconstruct the status of bug reports on any given date, thereby allowing the investigation of a project's evolution over time. Furthermore, existing tools to analyze the development process and evolution based on data from version control systems, such as CVS, will be used [2], [8].

C. Development of interventions

In the third phase, case studies will be performed to study good practices. Based on these studies, interventions will be developed to improve release management. They will subsequently be tested in live projects. On the assumption that the results of phase two demonstrates that time-based releases are indeed a viable strategy offering certain advantages over other strategies, this phase will also consider the migration of a project to time-based releases. This work is based on action research involving live projects, such as Debian [3], which faces considerable problems with its releases and is searching for solutions (Fig. 1). There will also be quasi-experiments to test the effectiveness and the impact on the quality of different release practices.

IV. Conclusions

This research focuses on release management as one aspect of quality management and quality improvement in volunteer free software and open source projects. Release management is a problematic area in open source development in which significant improvements are possible. Research in this area that is carried out in close collaboration with the free software community has the potential to make a substantial contribution. In addition to identifying good release management practices, this research will investigate whether a group of volunteers can make predictable and high quality releases. This addresses fundamental questions regarding the open source development model and qualifies whether consistent levels of quality and predictable schedules are possible in distributed, volunteer projects.

V. Acknowledgements

This work is supported in part by Fotango, the NUUG Foundation and the EPSRC.

References

[1] J. R. Erenkrantz, "Release management within open source projects," in *3rd Workshop on Open Source Software Engineering*. ICSE, 2003, pp. 51–55.

[2] D. M. German, "Mining CVS repositories, the softChange experience," in *International Workshop on Mining Software Repositories*. ICSE, 2004, pp. 17–21.

[3] J. M. González-Barahona, G. Robles, M. Ortuño Pérez, L. Rodero-Merino, J. Centeno González, V. Matellan-Olivera, E. Castro-Barbero, and P. de-las Heras-Quirós, "Analyzing the anatomy of GNU/Linux distributions: methodology and case studies (Red Hat and Debian)," in *Free/Open Source Software Development*, S. Koch, Ed. Hershey, PA, USA: Idea Group Publishing, 2004, pp. 27–58.

[4] T. J. Halloran and W. L. Scherlis, "High quality and open source software practices," in *Proceedings of the 2nd Workshop on Open Source Software Engineering*. Orlando, FL, USA: ICSE, 2002.

[5] N. Jørgensen, "Putting it all in the trunk: Incremental software engineering in the FreeBSD open source project," *Information Systems Journal*, vol. 11, no. 4, pp. 321–336, 2001.

[6] M. Michlmayr, "Managing volunteer activity in free software projects," in *Proceedings of the 2004 USENIX Annual Technical Conference, FREENIX Track*, Boston, USA, 2004, pp. 93–102.

[7] M. Michlmayr and B. M. Hill, "Quality and the reliance on individuals in free software projects," in *Proceedings of the 3rd Workshop on Open Source Software Engineering*. Portland, OR, USA: ICSE, 2003, pp. 105–109.

[8] G. Robles, J. M. González-Barahona, and R. A. Ghosh, "GlueTheos: Automating the retrieval and analysis of data from publicly available software repositories," in *International Workshop on Mining Software Repositories*. ICSE, 2004, pp. 28–31.

[9] D. C. Schmidt and A. Porter, "Leveraging open-source communities to improve the quality & performance of open-source software," in *Proceedings of the 1st Workshop on Open Source Software Engineering*. Toronto, Canda: ICSE, 2001.

[10] A. Senyard and M. Michlmayr, "How to have a successful free software project," in *Proceedings of the 11th Asia-Pacific Software Engineering Conference*. Busan, Korea: IEEE Computer Society, 2004, pp. 84–91.