

Open Source Software Development (OSSD) Based On Software Engineering

Dengya Zhu , Vidyasagar Potdar, and Elizabeth Chang
School of Information Systems, Curtin University of Technology.
GPO Box U1987, Perth WA 6845, Australia
{dengya.zhu, vidyasagar.potdar, elizabeth.chang}@cbs.curtin.edu.au

Abstract. With the advent of Open Source Software (OSS) at the end of last century, many proponents believe that OSS is a new software development process and some even advocate OSS as a revolution for software engineering. The Cathedral and the Bazaar is a typical metaphor of the software development methodologies for the Closed Source Software (CSS) and the OSS. By comparing the phased (namely, requirement analysis, document design and system design, coding, testing and maintenance) software development methodology proposed by Software Engineering (SE), and by studying the management tools provided by SourceForge.net, we believe OSS development method not only follows the phased software development process, but also in return enriches the theory of SE.

“Software engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1)” [5]. Software product engineering includes software requirements, design, coding, testing, and software operation and maintenance [2]. Contemporary software development process is also iterative and agile [5]. Frederick and Brooks claim *No Silver Bullet* [1]; Raymond, however, argues that OSS development process is a breakthrough of SE. By comparing the different development phases suggested by SE with the typical development procedure of OSS, the writers try to verify that the OSS development process is not only based on SE, but also in return enriches the SE theory in testing and maintenance phase.

The purpose of requirement analysis is to manifest the exact needs of software and document it unambiguously. It is true that seldom are there formal documents of requirements among OSS development. However, usually there is a mailing list or newsgroup to discuss the requirements [9]; some OSS programmers are themselves user [7]. They can also refer to the existing CSS to get the requirement [3].

Software design concerns with the transformation of requirements into a description of how these requirements are to be implemented. Although lack of formal design documentation, successful OSS project tend to be architected by developers of extraordinary skills and experience; the underlying architecture and implementation often begins as an inheritance from a traditional SE project [3]; mail archives and community chat sites are also sources of relevant documentation.

Construct software components that are identified and described in the design documents is the goal of coding. While coding is only a small percentage within the phased SE, for some OSS developers, software is nothing but coding. Spend a lot

Please use the following format when citing this chapter:

Zhu, D., Potdar, V., and Chang, E., 2006, in IFIP International Federation for Information Processing, Volume 203, Open Source Systems, eds. Damiani, E., Fitzgerald, B., Scacchi, W., Scotto, M., Succi, G., (Boston: Springer), pp. 345-346

time on coding can compensate the lack of sufficiency design document, by “release early, release often, and listen to your customers” [6], this OSS development strategy has been proven results in faster, feasible and economic coding [4].

“Given enough eyeballs, all bugs are shallow” [6]. “By sharing hypotheses and results with a community of peers, the scientist enable many eyes to see what one pair of eyes might miss” [9]. The success of many OSS products has proven that software test productivity scale up as the number of developers helping to debug the software increases [8].

SourceForge.net is the world’s largest OSS development web site which provides free hosting and management to OSS development projects (<http://sourceforge.net>). SourceForge.net provides a wide range of services, such as web tools for community and project management, file release system, compile farm, version control system, communication tools, publicity, and project web service. These services facilitate the OSS developers to follow the phased development process suggested by SE.

References

1. Frederick, P. and Brooks, Jr. No Silver Bullet: Essence and Accidents of Software Engineering, IEEE Computer 20 (1987), 10-19.
2. Hilburn, T. B. Hirmanpour, I. Khajenoori, S Turner, R. and Qasem, A. A Software Engineering Body of Knowledge, Carnegie Mellon Software Engineering Institute,
3. Massey, B. Where Do Open Source Requirements Come From. In Proceedings of the 2nd Workshop on Open-Source Software Engineering, (Orlando, Florida, May 19-25, 2002).
4. Potdar, V. and Chang, E. 2004, Open Source and Close Source Software Development Methodology. In The 4th Workshop on Open Source Software Engineering, (Edinburgh, Scotland, May 25, 2004). ACM Press, New York, 2004, 105-110.
5. Rressman, R.P. Software Engineering: A Practitioner’s Approach, 6th ed. McGraw-Hill, New York, 2005.
6. Raymond, E. S. The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolution, O’Reilly, Sebastopol, 1999.
7. Scacchi, W. Software Development Practices in Open Software Development Communities. In Proceedings of the 23rd International Conference on Software Engineering. (Toronto, Canada, May 15, 2001). ACM Press, New York, 2001, 48-51.
8. Schmidt, D. C. and Porter, A. Leveraging Open-Source Communities to Improve the Quality & Performance of Open-Source Software. In Proceedings of the 23rd International Conference on Software Engineering. (Toronto, Canada, May 15, 2001). ACM Press, New York, 2001, 52-56.
9. Vixire, P. Software Engineering. In C. DiBona, S. Ockman & M. Stone(eds), Open Sources: Voices from the Open Source Revolution, O’Reilly Sebastopol, 1999.