Open Source Software Development Process

Yongsen MA

Shanghai Jiao Tong University

E-mail: mayongsen@gmail.com

I. STATEMENT OF PURPOSE

I want a detailed description of the process. For example, how are crash reports handled? How are bug reports handled? How are bugs classified and confirmed? How are the assignments to individual developers made? How to merge code changes in Git? How are code inconsistency handled? In each step of the process, have you identified any software engineering issues which have rooms for improvements?

Aaron Koblin: Artfully visualizing our humanity

Why free software has poor usability, and how to improve it

II. INTRODUCTION

A. motivating, joining, participating and contributing

   1) acquire: knowledge, experience, opportunities; backup, platform

   2) participate: happiness, communication

   3) contribute: freedom, trustworthy

   1) developer

   2) user(evaluation)

   1) public

   2) private

B. modeling, examination, investigation

   1) individuals

   2) groups

3) organizations

1) operate systems

2) web

3) application

4) network

1) contribute:

2) process: stable, scalable

3) acquire: software, individuals, groups

1) graph theory

2) multiproject

3) interdependent

III. MAINTAINING

How are the assignments to individual developers made? How to merge code changes in Git? How are code inconsistency handled? In each step of the process, have you identified any software engineering issues which have rooms for improvements?

A. Components

1) Home Page
2) Code Repository
3) Mailing List
4) Bug Tracking System
5) Wiki

B. Participate

1) Starting
2) Discussion

- Subscribe Mailing List
- Take part in News Group
- Participate in Conference

   3) Programming

- Consummate documents
- Running test codes
- Report Bugs
- Submit patch

   4) Improving

C. Git and ath9k

IV. DEBUGGING

For example, how are crash reports handled? How are bug reports handled? How are bugs classified and confirmed?

A. Basic Debugging

B. Functional Debugging

V. IMPROVEMENT

REFERENCES

[1] M. Bazilian, A. Rice, J. Rotich, M. Howells, J. DeCarolis, S. Macmillan, C. Brooks, F. Bauer, and M. Liebreich. Open source software and crowdsourcing for energy analysis. *Energy Policy*, 49(0):149 – 153, 2012.

[2] A. Bonaccorsi and C. Rossi. Why open source software can succeed. *Research Policy*, 32(7):1243 – 1258, 2003.

[3] K. Crowston and H. Annabi. Information systems success in free and open source software development: Theory and measures. In *Software Process: Improvement and Practice*, pages 123–148, 2006.

[4] M. Glassman and M. J. Kang. Intelligence in the internet age: The emergence and evolution of open source intelligence (osint). *Computers in Human Behavior*, 28(2):673 – 682, 2012.

[5] G. Hertel, S. Niedner, and S. Herrmann. Motivation of software developers in open source projects: an internet-based survey of contributors to the linux kernel. *Research Policy*, 32(7):1159 – 1177, 2003.

[6] B. Kogut and A. Metiu. Opensource software development and distributed innovation. *Oxford Review of Economic Policy*, 17(2):248–264, 2001.

[7] S. Prehn. Open source software development process. *Term Paper July*, 29:2007, 2007.

[8] W. Scacchi, J. Feller, B. Fitzgerald, S. Hissam, and K. Lakhani. Understanding free/open source software development processes. *Software Process: Improvement and Practice*, 11(2):95–105, 2006.

[9] G. von Krogh and E. von Hippel. Special issue on open source software development. *Research Policy*, 32(7):1149 – 1157, 2003.

[10] J. Wang. Survival factors for free open source software projects: A multi-stage perspective. *European Management Journal*, 30(4):352 – 371, 2012.

[11] C. Yilmaz, A. M. Memon, A. Porter, A. S. Krishna, D. C. Schmidt, and A. Gokhale. Techniques and processes for improving the quality and performance of open-source software. In *Software Process: Improvement and Practice*. John Wiley & Sons, 2006.

[12] M. Zschoch. The success of open source. *Canadian Journal of Political Science/Revue canadienne de science politique*, 40(01):250–252, 2007.