Chapter 51

Open Source Software Development Process Model: A Grounded Theory Approach

Keng Siau
Missouri University of Science and Technology, USA

Yuhong Tian
Express Scripts Holding Company, USA

ABSTRACT

The global open source movement has provided software users with more choices, lower software acquisition cost, more flexible software customization, and possibly higher quality software product. Although the development of open source software is dynamic and it encourages innovations, the process can be chaotic and involve members around the globe. An Open Source Software Development (OSSD) process model to enhance the survivability of OSSD projects is needed. This research uses the grounded theory approach to derive a Phase-Role-Skill-Responsibility (PRSR) OSSD process model. The three OSSD process phases -- Launch Stage, Before the First Release, and Between Releases -- address the characteristics of the OSSD process as well as factors that influence the OSSD process. In the PRSR model, different roles/actors are required to have different skills and responsibilities corresponding to each of the three OSSD process phases. This qualitative research contributes to the software development literature as well as open source practice.

INTRODUCTION

With the increasing popularity of open source software such as Mozilla Firefox, Linux, Apache, and mySQL in the business world, open source software is attracting more and more attentions in the academic community as well (Xu *et al*., 2011; Aksulu and Wade, 2010; Crowston and Scozzi, 2008). Advocates of open source software development argued that it could improve software product quality, encourage software evolution and innovation, and enhance business values (Mehra and Mookerjee, 2012; Allen, 2012; Shaikh and Cornford, 2012; Chengalur-Smith, Nevo and Demertzoglou, 2010; Amrit, 2009; Brydon and Vining, 2008; Long and Siau 2005,

DOI: 10.4018/978-1-4666-7230-7.ch051

2006, 2007, 2008; Long *et al*., 2007; Raymond, 2001; O'Reilly, 1999;). Among others, one key advantage of open source software is its assumed high quality due partly to the intensive peer review or "many eyeballs." Members of the open source communities give direct, specific and immediate feedback to the software code written by others. All members have access to the source code and can submit code patches. This software development model is called "Bazaar" model by Raymond (2001). Traditional cathedral models (Raymond, 2001) of information system development are in short of second party review. OSSD, or parallel development through Internet (Krogh, 2003), has practically been proven to be a very successful counterstrategy to this problem.

Although this "Bazaar" model of OSSD (Open Source Software Development) naturally encourages innovation and software evolution, it also causes some problems. Requirements analysis and conceptual modeling of the problem domains are issues (e.g., see Chua *et al*., 2012; Bucher and Diner, 2012; Chan *et al*. 2012; Poels 2011; Edwards and Sridhar, 2005). The virtual environment (e.g., Nath *et al*., 2008) and motivation of team members (e.g., Liu *et al.* 2010) need to be addressed too. Also, lacking of conformity and unstructured project management tends to decrease the survivability of the OSSD project. In OSSD, there are no formally documented norms or customs except for the implied customs and taboos learned through experience, not to mention any comprehensive development process guidelines. Cusumano (2004) believed that many OSSD projects are "semi-organized chaos." The state of affairs of OSSD urgently calls for an increased understanding of the unique OSSD process and the development of OSSD process models to help guide OSSD project management.

The nature of OSSD projects is dynamic. Collaborations in OSSD are extraordinary, which makes a systematic software development process very imperative. The unique features of OSSD imply that many existing standardized or quasi-standardized software development processes such as CMM (Capability Maturity Model) and UP (Unified Process), cannot be readily applied to OSSD.

Although managing an OSSD project has some similar characteristics as that of traditional software development. The differences are very apparent. OSSD has its own unique characteristics. To name just a few, first, the virtual development team of an open source project can become very large and change frequently, which requires appropriately breaking up the project into distinct components and documenting the specifications clearly so that programming teams can work on them without much communication. Second, the voluntary, peer-to-peer decentralized virtual team (Kwok and Gao, 2004) demands different project leadership (Pauleen, 2003) than that of the usual software development team. Third, the lifecycle of OSS is supposed to be longer because the publicly available source codes are usually continuously being updated (Lerner and Tirole, 2002). As Berglund and Priestley (2001, p134) pointed out: "The nature of open-source development still remains somewhat uncharted territory… but is typically (among other characteristics) robust, public, just-in-time, user-driven, global, community oriented, critical-mass dependent, non-directional in its growth, developed from the bottom up, and change-prone." These situations call for the study of OSSD process and the development of OSSD process model. This research aims to identify key factors and variables that affect the OSSD process, and to develop a process model to guide OSSD practice.

LITERATURE REVIEW

A software process refers to the activities performed during the development of a software product or system (Siau *et al.*, 2010; Curtis *et al.*, 1992). Two of the well-known software processes are CMM and UP. Both CMM and UP focus on