Chapter 43
Managing Knowledge in Open Source Software Test Process

Tamer Abdou
Concordia University, Canada

Peter Grogono
Concordia University, Canada

Pankaj Kamthan
Concordia University, Canada

ABSTRACT

The increasing adoption and use of Open Source Software (OSS) motivates study of its development. This chapter explores the state-of-the art in OSS development processes, in general, and OSS testing processes, in particular. A conceptual model for software Testing Knowledge Management (TKM) that aims to provide an understanding of the testing domain is introduced. The TKM model is informed by earlier studies and guided by international testing standards. Moreover, the TKM model is equipped with different forms of knowledge, reusable across software projects. Using the TKM model as an integrative conceptual model enables understanding of how knowledge life cycle stages are mapped onto the test process of OSS, what type of knowledge is created at each stage, and how knowledge is converted from one stage to another. The chapter is supported by representative examples of OSS that are mature and currently in widespread use.

INTRODUCTION

For the past decade or so, Open Source Software (OSS) has been playing an increasingly significant role in society. OSS has opened new vistas in communication, education, and entertainment. However, an examination of major, publicly-available, source code foundries reveals that the successes of OSS have come with their share of abandonments. From a software engineering perspective, this brings attention to the quality of OSS, and commitments towards assuring it.

The test process aims to improve the quality of software, and is one of the most significant processes in software development (Kaner, Falk, & Nguyen, 1999). This process in the context of OSS can be considered a knowledge-intensive process involving (1) several tasks on different

DOI: 10.4018/978-1-4666-7230-7.ch043

platforms, and (2) several decisions that need to be made to satisfy software requirements. The knowledge required to make proper decisions and to assess them is diverse and broad (such as voting in Apache server and review in Mozilla Firefox browser). This chapter explores Knowledge Management (KM)-related activities in the software test process identified in OSS (Abdou, Grogono, & Kamthan, 2012). Moreover, the chapter reviews various approaches, techniques, and tools that can be used to capture and manage the knowledge that is required or can be produced during the test process.

The rest of the chapter is organized as follows. First, OSS development process and test process in OSS are introduced. Next, an overview of KM is given, including different modes of knowledge conversion from the perspective of software testing. This is followed by a review of different activities of the test process in OSS to illustrate the relation between test process and KM. The chapter ends with directions for future research and concluding remarks.

BACKGROUND

In this section, OSS software development process is discussed, different activities related to the test process in OSS are reviewed and compared with those in conventional software, and basics of knowledge as per OSS are given.

Open Source Software Development Process

The earlier research on OSS focused on the aspects of development, project, virtual community, and the roles of contributors (Raymond, 1999; Zhao & Elbaum, 2003; Lonchamp, 2005). In several initial studies, it was observed that there are stark differences between conventional software engineering and OSS development. In particular, it has been found that, in contrast to conventional software processes, the OSS development process is unstructured (Mockus, Fielding, & Herbsleb, 2002; Zhao & Elbaum, 2003) and lacks formal documentation, including that for testing (Lonchamp, 2005; Wang, Guo, & Shi, 2007).

The OSS development process follows a layered approach with an onion shaped structure (Crowston & Howison, 2005; Showole, Sahibuddin, & Ibrahim, 2011; Crowston & Howison, 2006), as shown in Figure 1. At the center of this structure are the core developers who direct the design, contribute most of the code, and steer the evolution of the software. In the innermost ring are the co-developers who are responsible for submitting patches, which are reviewed and checked-in by core developers. In the intermediate ring are the active users who do not contribute code, but provide use cases and bug reports based on their own test suites. Finally, in the outermost ring are the passive users and observers who are not interested in contributing code, but like to stay informed of the development.

It is only recently that a generic model of OSS development process based on 'mature' OSS projects has been proposed (Lonchamp, 2005). (An OSS project is considered mature if it involves more than ten developers.) The model comprises a set of elements, where each element could represent a process role, tool, activity, or document. In this model, developers work concurrently on either coding or testing. Moreover, this model is consistent with previous OSS development process models (Gillian, 2001; Scacchi, 2003), and enables systematic analysis, reuse, and comparison.

Test Process in Open Source Software Development

There are currently a number of standards for test processes, including ISO/IEC TR 19759, BS 7925-2, and ISO/IEC WD 29119-2 (Reid, 2000; ISO/IEC, 2010). The ISO/IEC 29119 Test Process Standard (or, ISO/IEC Test Process, for short), currently under development, serves as a