Chapter 14
Predicting OSS Development Success: A Data Mining Approach

Uzma Raja
University of Alabama, USA

Marietta J. Tretter
Texas A&M University, USA

ABSTRACT

Open Source Software (OSS) has reached new levels of sophistication and acceptance by users and commercial software vendors. This research creates tests and validates a model for predicting successful development of OSS projects. Widely available archival data was used for OSS projects from Sourceforge.net. The data is analyzed with multiple Data Mining techniques. Initially three competing models are created using Logistic Regression, Decision Trees and Neural Networks. These models are compared for precision and are refined in several phases. Text Mining is used to create new variables that improve the predictive power of the models. The final model is chosen based on best fit to separate training and validation data sets and the ability to explain the relationship among variables. Model robustness is determined by testing it on a new dataset extracted from the SF repository. The results indicate that end-user involvement, project age, functionality, usage, project management techniques, project type and team communication methods have a significant impact on the development of OSS projects.

INTRODUCTION

Software development is an expensive endeavor and the risk of failure is high. In this scenario, users are looking for reliable but cheap alternatives to expensive software systems. Open source software (OSS) that is developed online by volunteer participants, offers one such alternative (Roberts, Hann, & Slaughter, 2006). These projects are distributed publicly with free and open access to their source code. Many organizations are accepting OSS projects as an inexpensive and reliable alternative to commercial Closed Source Software (CSS) projects. Some commercial soft-
ware development companies adopt existing OSS projects, while others launch their own projects in OSS communities (Cearley, Fenn, & Plummer, 2005). A few OSS projects are headed towards the stock market offering a new venue of high tech investment (Ricadela, 2007). However, the success rate of OSS projects is not any higher than CSS projects (Chengalur-Smith & Sidorova, 2003; Krishnamurthy, 2002). Therefore, as the use of OSS projects increases, it is critical to have performance evaluation models for these projects.

To understand why some OSS projects are more successful than others, it is critical to understand the development process and the factors that impact successful project development. The process model underlying software system success, as proposed by DeLone and McLean (1992), has three steps to it: system creation, system use and consequences. This research focuses on the very first step of system creation and analyzes factors that promote successful system creation or completion. The goal of this study is to address two vital questions:

- Can we predict successful OSS development?
- Can we identify the factors that lead to successful OSS development?

Data archives of OSS projects are publicly available. These archives contain rich project lifecycle information. There has been a move towards automation of software development metric collection and analysis (German, 2004). Mokus, Fielding, and Herbsleb (2002) studied two large scale OSS developments i.e. Apache and Mozilla, using the development archives. The availability of data archives for OSS projects invites the use of mining techniques. Williams and Hollingsworth (2005) demonstrated how data mined from source code repositories can improve static analysis tools. Jensen and Scacchi (2005) used data mining techniques for studying OSS development communities by combining Text Mining and link analysis to discover update patterns. Dinh-Trong and Bieman (2005) analyzed the maintenance process by mining repositories of a long-lived open source project, FreeBSD. Zimmerman et al. (2005) developed an approach to use association rule mining on CVS data to recommend source code that is potentially relevant to a given code fragment. Ying, Murphy, Ng, and Chu-Carroll (2004) developed a mining approach on the change history of source code to identify relevant code for change task.

OSS projects evolve in cyberspace and their growth models do not follow the traditional CSS developed through organized resources. Godfrey and Tu (2001) conducted one of the first analyses of Linux code evolution and compared it to the existing laws of software evolution. Robles et al. (2006) built on their work to test the model on a newer dataset and found evidence of OSS evolution being different than CSS evolution. Paulson, Succi, and Eberlein (2004) compared the evolution of OSS and CSS projects using linear approximation techniques. Koch (2007) found that the nature of OSS evolution largely depends on the size of the project. Software evolution studies have also facilitated study of software maintenance, since they provide a framework for analyzing OSS projects. While the studies in OSS evolution have focused on growth patterns of projects, there are also studies that have examined the overall success of OSS projects.

Crowsten, Annabi, and Howison (2003) presented a framework for OSS success. Several studies have since focused on various aspects of OSS success. Stewart, Ammeter, and Maruping (2006) explore effects of license choice on OSS development; defining OSS success in terms of number of downloads. Mokus, Fielding, and Herbsleb (2002) have studied OSS success from the point of view of downloads as well. Other studies have investigated success form the point of view of the social network that is supported by the project (Grewal, Lilien, & Mallapragada, 2006; Singh, Tan, & Mookerjee, 2007). The number
Related Content

A Comparative Study of the EUREQA Tool for End-User Development
www.igi-global.com/article/comparative-study-eureqa-tool-end/67581?camid=4v1a

Enhancing ERP System with RFID: Logistic Process Integration and Exception Handling
www.igi-global.com/article/enhancing-erp-system-rfid/58513?camid=4v1a

Conceptual Modeling Method for Separation of Concerns and Integration of Structure and Behavior
www.igi-global.com/article/conceptual-modeling-method-separation-concerns/61395?camid=4v1a

Integrating DSLs into a Software Engineering Process: Application to Collaborative Construction of Telecom Services
www.igi-global.com/chapter/integrating-dsls-into-software-engineering/71828?camid=4v1a