Chapter 2.16
Dynamical Simulation Models of the Open Source Development Process

I.P. Antoniades
Aristotle University of Thessaloniki, Greece

I. Samoladas
Aristotle University of Thessaloniki, Greece

I. Stamelos
Aristotle University of Thessaloniki, Greece

L. Angelis
Aristotle University of Thessaloniki, Greece

G.L. Bleris
Aristotle University of Thessaloniki, Greece

ABSTRACT

This chapter will discuss attempts to produce formal mathematical models for dynamical simulation of the development process of Free/Open Source Software (F/OSS) projects. First, a brief overview for simulation methods of closed source software development is given. Then, based on empirical facts reported in F/OSS case studies, we describe a general framework for F/OSS dynamical simulation models and discuss its similarities and differences to closed source software simulation. A specific F/OSS simulation model is introduced. The model is applied to the Apache project and to the gtk+ module of the GNOME project, and simulation outputs are compared to real data. The potential of formal F/OSS simulation models to turn into practical tools used by F/OSS coordinators to predict key project factors is demonstrated. Finally, issues for further research and efforts for improvement of this first-attempt model are discussed.

INTRODUCTION

There have been a few studies attempting to define the Open Source Software (OSS) development process in general terms (Bollinger, Nelson, Self & Turnbull, 1999; Feller & Fitzgerald, 2000; McConnell, 1999; O'Reilly, 1999; Raymond, 1998; Wilson, 1999), and there have also been a few case studies of OSS projects: Linux (Godfrey & Tu, 2000), Apache WWW-server (Mockus, Fielding, & Herbsleb, 2000), FreeBSD (Jorgensen, 2001), GNOME (Capiluppi, Lago, & Morisio, 2003; Koch & Schneider, 2000). The latter studies presented some interesting qualitative data for the F/OSS development process, managerial issues, and programmer attitudes, as well as *quantitative* data regarding the total Lines of Code (LOC) added as a function of time, the defect density of the code produced, number of programmers and contributions per project module/task, average work-effort/time to submit a contribution (code change, defect correction, code testing), and other statistical measures. Despite the fact that these studies have produced interesting results validating or disproving certain hypotheses regarding F/OSS development on a per case basis, there is not sufficient global understanding nor a precise definition of the open source development process—the results show both similarities and clear differences in processes and outputs among different projects, but there is no adequate explanation of presented facts based on more general principles. In many cases, the authors offer descriptive explanations based on plausible assumptions but, as there is no general model to quantify their claims together with their possibly complicated interactions, the validity of such explanations cannot be directly demonstrated.

Therefore, there is a need to move from descriptive models based on special cases to a more general quantitative mathematical model that would hopefully be used as a demonstrating tool of real case results. Most importantly, this model could serve as a *predicting* tool of key F/OSS project factors, such as project failure/success, dynamical evolution of source code, defect density/architectural quality, expected number of programmers involved, and distribution of work effort to distinct project modules and tasks.

Previous studies have shown that the dynamical evolution of the above key factors is quite sensitive to (a) the type of software developed, and (b) the specific technical management framework of an F/OSS project. Therefore, the model should be general enough so that, by a straightforward adjustment of model parameters, it is able to simulate various types of F/OSS projects under alternative managerial scenarios.

The "predictive power" of such a model could be viewed as follows: by first calibrating the model parameters against available historical data from a certain time period within the development phase of an F/OSS project, the model should be able to approximately reproduce the *future evolution* of the same F/OSS project.

This chapter will discuss attempts to produce formal mathematical models for dynamical simulation of the development process of F/OSS projects. Whereas several such models and corresponding computer simulation studies exist for the traditional (closed source) software development process, it is only very recently that such attempts have started to appear in international literature regarding the F/OSS development process. We will briefly describe closed source simulation models and their practical applications in software engineering. We will then introduce a general framework for generic F/OSS dynamic simulation models, compare and contrast it to traditional closed source simulation models, and discuss its possible use in describing the F/OSS development process and its possible practical applications for the F/OSS community. Then, we will introduce a specific simulation model and demonstrate it against the Apache case study (Mockus et al., 2000) and evolution of the gtk+ module of the GNOME project (Koch & Schneider, 2000; http://bulunga.dat.escet.urjc.es/gnome-cvs/index.