OPEN UNIVERSITY UK

MASTER DISSERTATION

Evaluating phylogenetic methods for quantifying risks and opportunities presented by forks in open source software

Author:
Alvaro ORTIZ TRONCOSO

Supervisor:
Dr. Doug LEITH

A dissertation submitted in partial fulfilment of the requirements for the degree of Master of Science in Computing (Software Engineering)

8. September 2017

OPEN UNIVERSITY UK

Abstract

Master of Science in Computing (Software Engineering)

Evaluating phylogenetic methods for quantifying risks and opportunities presented by forks in open source software

by Alvaro ORTIZ TRONCOSO

Software needs to evolve in order to deliver value to its stakeholders throughout its lifecycle. The most important drivers behind software evolution are changing user expectations, and a dynamic and innovative ecosystem. Open source software development is a software development paradigm that embraces change by blurring the distinction between users and developers. The cornerstone of open source development is a licensing scheme that grants anybody the right to examine, to copy and to modify the source code. Open source licenses have proven in many cases a viable alternative to strong intellectual property protection regimes. However, open source licenses expose software projects to a new kind of risk: forking. Forking happens when part of the team takes off in a new direction. Literature on the governance of open source projects disagrees on whether forking is a risk or an opportunity: the traditional view is that forking is the result of a failure of the project to keep its resources together however, as successful software products have lately emerged from forks, the traditional view is challenged. Notwithstanding, methods for quantifying the evolution of forks are currently scarce: the present research attempts to port methods from phylogenetics, a branch of evolutionary biology that attempts to unravel the mechanisms behind the evolution of living organisms, to the study of the evolution of forks and postulates that the progress of a fork can be modelled using these methods. Methods and concepts from evolutionary biology were validated by applying them to three cases of software forks. A statistical analysis shows that the history of a forked project can be reconstructed using phylogenetic trees, and finds evidence that the eventuality of a fork could be predicted. However, no evidence was found that the outcome of a fork can be foretold using these methods. The present research concludes by porting basic concepts from evolutionary biology into a software development context and elaborates how phylogenetic methods and concepts can be used by practitioners to increase their understanding of forking processes.

Acknowledgements

I am extremely grateful to the Open University staff, who have provided me with such professional and well informed support. Special mention must be made of my tutor, Dr. Doug Leith, whose guidance has been invaluable throughout. I also would like to thank my colleagues at the Technical University Berlin, in particular Erhard Zorn and Dr. Stefan Born, for balancing a busy workload. I am especially thankful for the encouragement shown by my family and in particular for the patience afforded me by my dear friend Ina Kemter.

Contents

Glossary

Branch
A thread of development within a project or team; branches are common in open source development (Robles and González-Barahona, 2012).

Evolution
Defined in paragraph 1.3.

Fork
Defined in paragraph 1.3.

Merge
A rejoining of separate development strands that had branched or forked previously, either by integrating source code or by dismissing parts of either project (Robles and González-Barahona, 2012).

Open source software
Software development paradigm that blurs the difference between users and developers (Hippel and Krogh, 2003). Open source software licenses grant users the right to fork a project (Robles and González-Barahona, 2012).

Phylogenetic tree
A pictorial representation of the degree of relationship between entities sharing a common ancestry (Baum and Offner, 2008).

Release
A stage in the software lifecycle corresponding to a new generation of the system (Lehmann, 1980).

Chapter 1

Introduction

1.1 Background to the problem

A software system is constantly subject to change pressure from its environment, therefore it needs to evolve to deliver value to its stakeholders throughout its lifetime: it has been regarded at least since the 1980's that software evolution is the most expensive part of the software lifecycle (Lehman, 1980). Therefore, an understanding of the capacity of a system to adapt to changes in its environment can impact on the software production process (Yu and Ramaswamy, 2006).

Open source development challenges traditional best practices in software development by blurring the difference between users and developers (Hippel and Krogh, 2003). The open source development model originated in the academic community and has transcended into private enterprise, where in many cases it has proven an efficient alternative to strong intellectual property protection (Kogut and Metiu, 2001).

Common tasks in software evolution include cloning, branching and merging of the code base; in addition to these processes, the open source license grants the freedom to fork a project: open source projects can evolve in parallel, splitting in two or more different projects, steered by different development teams.

The gain in popularity of the open source development model has brought forward the importance of understanding forking processes. Kogut and Metiu (2001) argue that forking results in competing versions of the original project and that forking is therefore a major failure risk for open source projects. Nyman and Lindman (2013) argue that forking is a remedy against ailments of proprietary software (planned obsolescence, vendor lock-in, hostile takeovers, etc.) and that forking facilitates experimentation. Robles and González-Barahona (2012) suggest that a purposeful fork can solve technical-, license- and team-related problems by restoring the balance between the stakeholders of a project. Therefore, a controversy exists within the software engineering community, whether forking is a risk or an opportunity.

1.2 Justification for the research

Concepts in software evolution and biological evolution are often described using a common vocabulary (Yu and Ramaswamy, 2006), so it seems natural to look at evolutionary biology for potential methods to solve this controversy.

Lehman (1980) postulates that there are recurring patterns which govern software evolution, independently of the decisions taken by individual managers and programmers. Therefore, software evolution might, as biological evolution, be understood as a process which is not designed, but resulting from characteristics inherent to a population: A population need not be composed of biological entities, characteristics need not be encoded in DNA and the environment can be artificial,

thus the term "evolution" can be used to describe processes in different domains, as long as the population considered ensures its own perpetuation (Nehaniv et al., 2006). If the term "software system" is taken to encompass a system's infrastructure, code base and community of developers (Yu and Ramaswamy, 2006), then a software system is capable of sustaining itself, and thus the change processes affecting a population of software releases can be described as evolution (figure 1.1).

1.3 Definitions

Based on the work by Robles and González-Barahona (2012) a working definition of a fork can be formulated as follows:

Fork

A fork is a bifurcation from an existing project, resulting in an autonomous development strand, with its own name, infrastructure, code base and community of developers.

Based on the work by Nehaniv et al. (2006) and Yu and Ramaswamy (2006), a working definition of the evolution of software can be formulated as follows:

Software evolution:

The evolution of software is a process which affects a population of software releases. Software releases are characterized by code and organizational resources. Software evolution is distinct from the governance of a software project, as software evolution is independent of the decisions taken by individual managers and programmers.

1.4 Scope of the research

Forking is a process explicitly enabled by open source software licenses; forking is contrary to strong intellectual property protection; therefore the scope of the research is limited to open source software. Software forks are known to have occurred in the areas of networking, web applications, development environments, multimedia, games, operating- and desktop- systems, utilities, graphics software, databases, enterprise resource planning, security and package management (Robles and González-Barahona, 2012), thus affecting a large portion of the software development domain.

Some projects that have gained a large user base originated from forks, for example the MariaDB database engine, the Android operating system and the LibreOffice suite of office applications, forked for different reasons and with different outcomes. Rather than trying to gain a comprehensive overview of the impact of forking on software development, a task that was undertaken by Robles and González-Barahona (2012), the present research examined three forks: MySQL/MariaDB, Linux/Android and OpenOffice/LibreOffice. Data was collected from the projects' online repositories, without interacting directly with the developers ("third degree data"). This approach entails that data cannot be controlled nor its quality be assessed through other means, therefore, the availability and completeness of the data archived in the repositories was a factor that played an important role in the choice of projects to examine.

Using real-world case studies for describing a software development situation has been practiced in computer science, and it is possible to empirically test hypotheses using this approach (Runeson and Höst, 2009). Hypotheses were formulated as research questions, presented in paragraph 2.1.

1.5 Aim

Any organization aiming to adopt open source software development might face a fork situation during the software's lifecycle, or might decide to fork existing software as a means to solve technical-, license- and team-related problems or to facilitate experimentation, as suggested by Robles and González-Barahona (2012). The aim of this research is to gain empirical evidence of whether phylogenetic methods from evolutionary biology can quantify the risks and opportunities associated with software forking processes.

To this end, the present research reviewed the current state of literature on forking, selected suitable open source software repositories to collect data and implemented selected phylogenetic methods using appropriate libraries. An attempt was made to advance the understanding of forking processes by answering the research questions detailed in chapter 2.

1.6 Outline of the dissertation

The rest of the dissertation is organized as follows: chapter 2 defines the research questions, reviews existing literature and examines the objectives in detail. Chapter 3 examines the data acquisition process, justifies the choice of phylogenetic techniques and relates the chosen statistical techniques to the research questions. Chapter 4 details how the data was acquired, processed using phylogenetic techniques and analysed using statistical techniques. Chapter 5 concludes, delineates possible further research and reflects on the research process as it was carried out.