Chapter II

Analyzing the Anatomy of GNU/Linux Distributions: Methodology and Case Studies (Red Hat and Debian)

Jesús M. González-Barahona, Universidad Rey Juan Carlos, Spain
Gregorio Robles, Universidad Rey Juan Carlos, Spain
Miguel Ortuño-Pérez, Universidad Rey Juan Carlos, Spain
Luis Rodero-Merino, Universidad Rey Juan Carlos, Spain
José Centeno-González, Universidad Rey Juan Carlos, Spain
Vicente Matellán-Olivera, Universidad Rey Juan Carlos, Spain
Eva Castro-Barbero, Universidad Rey Juan Carlos, Spain
Pedro de-las-Heras-Quirós, Universidad Rey Juan Carlos, Spain

ABSTRACT

GNU/Linux distributions are probably the largest coordinated pieces of software ever put together. Each one is in some sense a snapshot of a large fraction of the libre software development landscape at the time of the release and, therefore, its study is important to understand the appearance of that landscape. They are also the working proof of the possibility of releasing reliable software systems in the range...
of 50-100 millions of lines of code, even when the components of such systems are built by hundreds of independent groups of developers, with no formal connection to the group releasing the whole system. In this chapter, we provide some quantitative information about the software included in two such distributions: Red Hat and Debian. Differences in policy and organization of both distributions will show up in the results, but some common patterns will also arise. For instance, both are doubling their size every two years, and both present similar patterns in programming language usage and package size distributions. All in all, this study pretends to show how GNU/Linux distributions are with respect to their source code, and how they evolve over time. A methodology of how to make comparable and automated studies on this kind of distributions is also presented.

INTRODUCTION

Libre software\textsuperscript{1} provides software engineering with a unique opportunity to make detailed characterizations of software projects that can be complete, detailed, and reproducible, since the source code is available for anyone to read. This makes it possible to build complete models based on public and repeatable studies. Based on this idea, it seems reasonable to collect data from libre software projects, to start building up a castle of numbers that can later be used to sustain theories about how libre software is developed.

In this respect, we have found GNU/Linux\textsuperscript{2} distributions to be a perfect example of what to study. During the second half of the 1990s, GNU/Linux distributions evolved and grew, to the point that at the beginning of the 2000s they include the most comprehensive, coordinated compilations of libre software. Therefore, when we study the most representative distributions, we are in fact analyzing a very important, and representative, subset of the mature libre software available at the time of the release of such distributions. Answering questions like which languages are more usual in these distributions, or how is the mean package size evolving, tells us about how the libre software community is working, and may help us in making predictions for the future (for instance, “when, if ever, will C++ surpass C as the most popular language in libre software distributions?”).

What is more important in terms of libre software engineering is the huge size of these distributions that makes them the state-of-the-art in terms of management of software aggregates (libre or not). It is really difficult to find coordinated collections of software of the size usual in GNU/Linux distributions, with complex interdependencies, composed by the results of hundreds of libre software projects (sometimes coordinated by volunteers, sometimes by companies, or, in many cases, by a mixture of both), which, when delivered, satisfy the requirements of literally millions of users worldwide. Understanding with some detail how these distributions are and how they evolve may help us to understand how this delivery of 50-100 millions
Creating Open-Source Interactive Articles for the Wider Publics
[www.igi-global.com/chapter/creating-open-source-interactive-articles/70127?camid=4v1a](www.igi-global.com/chapter/creating-open-source-interactive-articles/70127?camid=4v1a)

Impact of Programming Language Fragmentation on Developer Productivity: A Sourceforge Empirical Study
[www.igi-global.com/article/impact-programming-language-fragmentation-developer/44971?camid=4v1a](www.igi-global.com/article/impact-programming-language-fragmentation-developer/44971?camid=4v1a)