

## Chapter 7.23

# Exploring the Effects of Process Characteristics on Product Quality in Open Source Software Development

**Stefan Koch**

*Vienna University of Economics and Business Administration, Austria*

**Christian Neumann**

*Vienna University of Economics and Business Administration, Austria*

### ABSTRACT

There has been considerable discussion on the possible impacts of open source software development practices, especially in regard to the quality of the resulting software product. Recent studies have shown that analyzing data from source code repositories is an efficient way to gather information about project characteristics and programmers, showing that OSS projects are very heterogeneous in their team structures and software processes. However, one problem is that the resulting process metrics measuring attributes of the development process and of the development environment do not give any hints about the quality, complexity,

or structure of the resulting software. Therefore, we expanded the analysis by calculating several product metrics, most of them specifically tailored to object-oriented software. We then analyzed the relationship between these product metrics and process metrics derived from a CVS repository. The aim was to establish whether different variants of open source development processes have a significant impact on the resulting software products. In particular we analyzed the impact on quality and design associated with the numbers of contributors and the amount of their work, using the GINI coefficient as a measure of inequality within the developer group.

## INTRODUCTION

In recent years, free and open source software (OSS) has drawn increasing interest, both from the business and academic worlds. Projects in different application domains, like most notably the operating system Linux, together with the suite of GNU utilities, the office suites GNOME and KDE, Apache, sendmail, bind, and several programming languages, have achieved huge successes in their respective markets. Undeniably, they constitute software systems of high quality. This has led to discussions and analyses of the underlying development process, as OSS is unique not only in its licenses and legal implications.

The main ideas of this development model are described in the seminal work of Raymond (1999), *The Cathedral and the Bazaar*, first published in 1997. Raymond contrasts the traditional model of software development, which he likens to a few people planning a cathedral in splendid isolation, with the new 'collaborative bazaar' form of open source software development. In the latter model, a large number of developer-turned-users come together without monetary compensation to cooperate under a model of rigorous peer review and take advantage of parallel debugging, which altogether leads to innovation and rapid advancement in developing and evolving software products. In order to enable this while minimizing duplicated work, the source code of the software needs to be accessible, which necessitates suitable licenses, and new versions need to be released often. Most often, the license a software is under is used to define whether it is open source software, applying for example the open source definition (Perens, 1999) or the approach of free software as embodied in the GNU GPL (Stallman, 2002). Nevertheless, usually a certain development style and culture are also implicitly assumed, although no formal definition or description of an open source development process exists, and there is considerable variance in the practices actually employed by open source projects. Also the re-

lationship to and insights regarding practices of agile software development (Erickson, Lyytinen, & Siau, 2005; Turk, France, & Rumpe, 2005; Merisalo-Rantanen, Tuunanen, & Rossi, 2005) have been discussed (Koch, 2004a).

Possible advantages and disadvantages to the development of software of this new development model have been hotly debated (Vixie, 1999; McConnell, 1999; Bollinger, Nelson, Self, & Turnbull, 1999; Cusumano, 2004; Feller, Fitzgerald, Hissam, & Lakhani, 2005). For example the question of whether open source development positively or negatively impacts quality and security has been a topic of several analyses (Witten, Landwehr, & Caloyannides, 2001; Hansen, Köhntopp, & Pfitzmann, 2002; Payne, 2002; Stamelos, Angelos, Oikonomou, & Bleris, 2002; Koru & Tian, 2004; Feller et al., 2005). Different viewpoints have also developed regarding whether or not the open source development approach increases efficiency of software production (Feller et al., 2005). Critics argue that the largely missing requirements engineering and design phases, together with the trend to search for bugs in the source code late in the lifecycle, lead to unnecessarily high effort hidden by the relative ease of spreading it throughout the world (McConnell, 1999; Vixie, 1999). Proponents of the OSS development model counter with arguments of very high modularity, fast release cycles, and efficient communication and coordination using the Internet (Bollinger et al., 1999; Raymond, 1999).

Currently, much empirical research is proceeding on OSS processes. Often, the research relies on data available through mining the communication and coordination tools and their repositories (Cook, Votta, & Wolf, 1998; Dutoit & Bruegge, 1998; Atkins, Ball, Graves, & Mockus, 1999; Kemerer & Slaughter, 1999) in place in OSS projects in order to describe and characterize the development team and processes. Most notably, the source code control systems used have been found to be a source of information, together with mailing lists and bug tracking systems. These analyses

27 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/exploring-effects-process-characteristics-products/29548](http://www.igi-global.com/chapter/exploring-effects-process-characteristics-products/29548)

## Related Content

---

### A Method to Support Fault Tolerance Design in Service Oriented Computing Systems

Domenico Cotroneo, Antonio Pecchia, Roberto Pietrantuono and Stefano Russo (2012). *Theoretical and Analytical Service-Focused Systems Design and Development* (pp. 362-376).

[www.irma-international.org/chapter/method-support-fault-tolerance-design/66808](http://www.irma-international.org/chapter/method-support-fault-tolerance-design/66808)

### Metastructuring for Standards: How Organizations Respond to the Multiplicity of Standards

Ronny Gey and Andrea Fried (2022). *Research Anthology on Agile Software, Software Development, and Testing* (pp. 1272-1295).

[www.irma-international.org/chapter/metastructuring-for-standards/294519](http://www.irma-international.org/chapter/metastructuring-for-standards/294519)

### Finding Optimal Transport Route and Retail Outlet Location Using Mobile Phone Location Data

Giridhar Maji and Soumya Sen (2022). *International Journal of Software Innovation* (pp. 1-20).

[www.irma-international.org/article/finding-optimal-transport-route-and-retail-outlet-location-using-mobile-phone-location-data/301226](http://www.irma-international.org/article/finding-optimal-transport-route-and-retail-outlet-location-using-mobile-phone-location-data/301226)

### The Impact of Regulatory Compliance on Agile Software Processes with a Focus on the FDA Guidelines for Medical Device Software

Hossein Mehrfard and Abdelwahab Hamou-Lhadj (2011). *International Journal of Information System Modeling and Design* (pp. 67-81).

[www.irma-international.org/article/impact-regulatory-compliance-agile-software/53206](http://www.irma-international.org/article/impact-regulatory-compliance-agile-software/53206)

### Reverse Engineering from an XML Document into an Extended DTD Graph

Herbert Shiu and Joseph Fong (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 1313-1333).

[www.irma-international.org/chapter/reverse-engineering-xml-document-into/29447](http://www.irma-international.org/chapter/reverse-engineering-xml-document-into/29447)