Effects of the Number of Developers on Code Quality in Open Source Software: A Case Study

Brandon Norick, Justin Krohn, Eben Howard, Ben Welna, Clemente Izurieta

{ brandon.norick, justin.krohn, eben.howard, ben.welna }@msu.montana.edu, clemente.izurieta@montana.edu

MONTANA STATE UNIVERSITY | College of ENGINEERING

Department of Computer Science

Abstract / Overview

Eleven mature open source software (OSS) projects written in C/C++ were analyzed to determine if the number of committing developers impacts code quality. We used cyclomatic complexity, lines of code per function, comment density and maximum nesting as surrogate measures of code quality.

Previous Work

Poor code quality was found to be common in previous research projects on open source software. Though extensive studies have been conducted to examine the effects that the number of developers have on code quality in commercial systems, no significant studies have been done on open source systems.

Examined OSS Projects

We selected SourceForge projects deemed mature (as measured by the total number of downloads) and in active development. Selected projects were: Filezilla, WinSCP, Miktex, Notepad++, UltraVNC, Audacity, Exult, SMPlayer, TCL, Wireshark and TortoiseSVN. The major components of all selected projects were written in C/C++.

Tools and Methods

We utilized *Understand* from SciTools to gather all metrics listed below. Only the C/C++ sections of the code were analyzed. No external libraries were analyzed. We used the applications StatCVS and StatSVN to determine the number of developers who committed to the project within 90 days of the version's release. We define an 'active developer' as one who committed within the 90 days prior to the version's release.

Metrics Utilized

McCabe's Cyclomatic Complexity

Despite reservations from some researchers, cyclomatic complexity is often viewed as a good indicator of the effort required to produce and maintain code. In this study, it was found that procedural code with a score of 10 or lower indicates acceptable levels of complexity.

Comment Density

Research indicates that the percentage of comments to code should be approximately 30%. This metric suggests code is well documented, and therefore relatively easy to maintain and debug. Higher percentages are indicative of overly complex code.

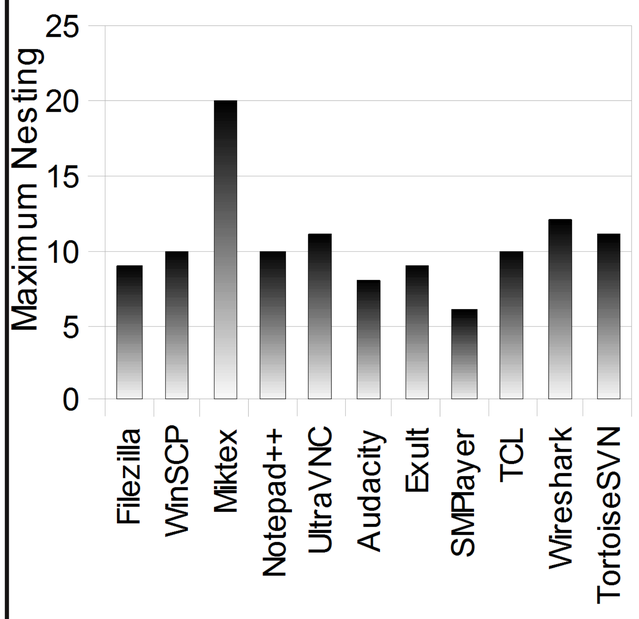
Lines of Code per Function

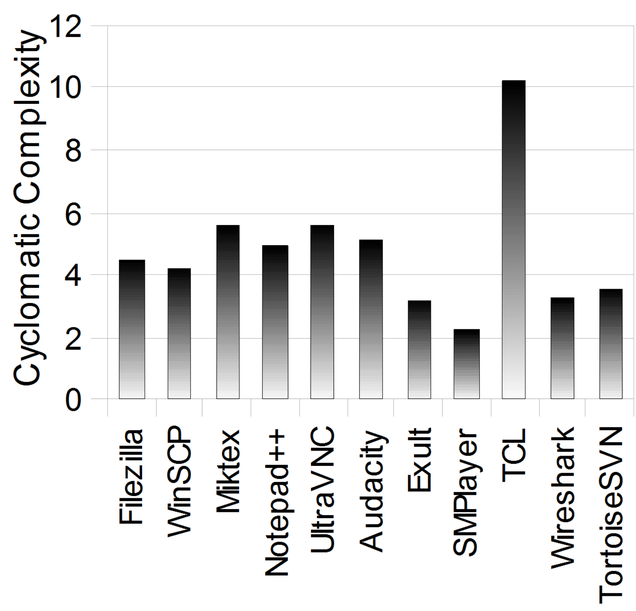
Research suggests that approximately fifty lines per function is considered ideal. There also exists research that demonstrates a correlation between lines of code and cyclomatic complexity. A discrepancy between these metrics should be analyzed carefully.

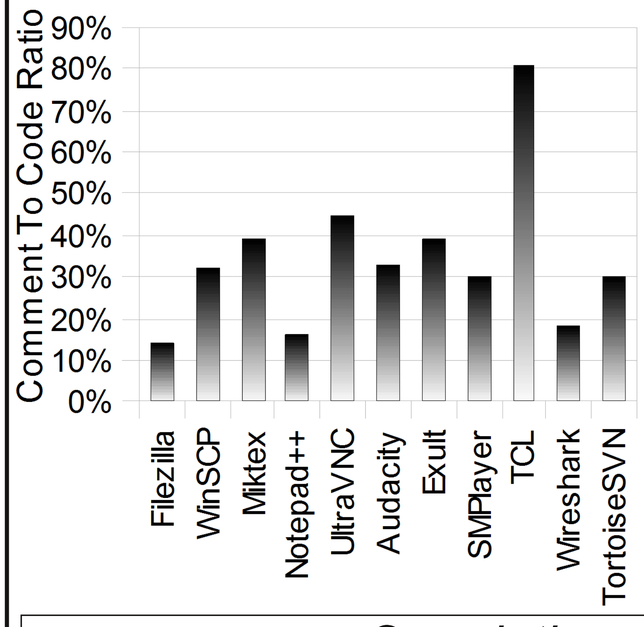
Maximum Nesting

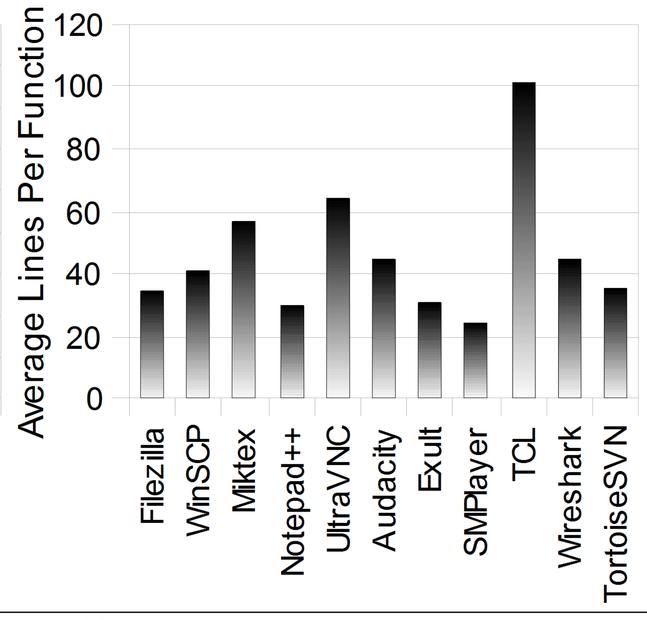
Maximum nesting (measured over the total lines of source code) is an indicator of the maximum complexity of that code. As this number increases, the code becomes more difficult to understand and the effort required to debug and modify the code also increases.

Results









Each graph was sorted in ascending order by the number of committing developers. Correlation indices suggest that there is no evidence to support a relationship between the number of developers and code quality. Ten out of eleven projects showed signs of good code quality regardless of the number of developers involved. One project (TCL) was sufficiently different in code style and purpose, that when compared with the other projects, its apparent poor quality metrics can be attributed to its intended use.

Correlation coefficients between quality metrics and the total number of committing developers

Quality Metric	Pearson's r^2	Spearman's r^2
Comment to code ratio	0.000225	0.003249
Average lines per function	0.003844	0.000081
Maximum nesting	0.001296	0.001681
Cyclomatic complexity	0.022801	0.084681