**The Effect of Class Size on Software Maintainability in Cyber Software**

**Objective:**

The purpose of this report is to analyze the effect of class size on software maintainability for the purpose of optimizing software solutions in the context of a business or enterprise environment using C&K metrics. Programs that are at least 5 years old and that are primarily used for Cybersecurity applications will be evaluated. This software criteria helps distinguish the software as remaining relevant over time, having gone through years of maintenance in a very rapidly changing field.

The main question being evaluated is, when measuring a program written in Java, what effect does class size have on the maintenance of the program?

To appropriately evaluate this question, certain C&K metrics that emphasize maintainability will be measured. Specifically, the Weighted Methods Per Class(WMC) and Lack of Cohesion of Methods(LCOM) metrics will be analyzed.

**Subject Program:**

The program being evaluated in this report is one developed by the National Security Agency(NSA) called Ghidra. Ghidra is a popular reverse engineering tool that is typically used to study Malware and other software. According to the GitHub documentation,"This framework includes a suite of full-featured, high-end software analysis tools that enable users to analyze compiled code on a variety of platforms including Windows, macOS, and Linux. Capabilities include disassembly, assembly, decompilation, graphing, and scripting, along with hundreds of other features"(National Security Agency, n.d.).

| **Attribute** | **Description** |
| --- | --- |
| **Name** | Ghidra |
| **Developer** | National Security Agency (NSA) |
| **Initial Release** | March 5, 2019 |
| **Latest Version** | 11.0.3 |
| **Programming Language** | Java, with some components written in C++ |
| **Platform** | Cross-platform (Windows, macOS, Linux) |
| **Main Features** | - Disassembler and decompiler |
|  | - Graphical user interface (GUI) |
|  | - Support for various processor instruction sets |
|  | - Scripting capabilities (Python and Java) |
|  | - Extensible architecture |

**Tools:**

For the collection of metrics and evaluation of C&K metrics, the Java Code Metrics Calculator (CK) tool created by Maurício Aniche was used(Aniche,15). CK evaluates source code and provides a report with the C&K metrics along with other metrics to evaluate software.

**Results:**

The utilization of the CK tool to evaluate Ghidra produced some surprising results. As mentioned earlier, the primary C&K metrics being used is WMC and LCOM. The result set produces several other metrics but WMC and LCOM are focused on for the purpose of evaluating maintainability over a period of time. The summary statistics of the WMC, LCOM and Lines of Coe are shown in the table below:

A table with numbers and symbols

Description automatically generated

Some of the general observations made are that we are evaluating 520 classes that make up Ghidra. An acceptable WMC and LCOM would be under 20 and under .5 respectively for each class. For WMC, 75% of the classes have a value of under 21. Also, 75% fall below .68 signifying generally acceptable values for both WMC and LCOM. It is important to note that on a large scale software product such as Ghidra, C&K metrics are used as a guideline and best practice to produce robust software but achieving 100% compliance rate is unrealistic.

When exploring the WMC metrics a bit more, we can see that there are some outliers when observing the standard deviation and when observing the boxplot below.

A graph with numbers and lines

Description automatically generated

It is clear that there are approximately four classes that have a WMC greater than 300 which is well outside of the target range. Specifically, the following classes were identified as potential candidates for refactoring based on these results.

A screenshot of a computer code

Description automatically generated

When removing these outliers, a clearer picture of the WMC values of a majority of classes within Ghidra can be observed. A graph with numbers and circles

Description automatically generated

**Conclusion:**

The key question addressed in this report is the effect of class size on the maintainability of Java programs, measured using the Weighted Methods Per Class (WMC) and Lack of Cohesion of Methods (LCOM) metrics. The subject of this analysis is Ghidra, a sophisticated reverse engineering tool developed by the National Security Agency (NSA).

To collect and evaluate the relevant C&K metrics, the Java Code Metrics Calculator (CK) tool was used. The analysis of Ghidra using this tool brought several interesting insights. Specifically, 520 classes were evaluated and some of the primary observations were that 75% of the classes have a WMC below 21 and an LCOM below 0.68, indicating generally acceptable maintainability metrics. However, outliers were identified, with approximately four classes exhibiting WMC values exceeding 300, far above the acceptable range. These outlier classes suggest potential candidates for refactoring. As mentioned earlier, it is important to remember that these metrics are guidelines and serve to provide a framework to ensure that best practices are being followed to help with maintainability. Despite the presence of some outliers in both WMC and LCOM, the overall maintainability of Ghidra remains robust as demonstrated by the years of remaining a relevant and popular tool in a fast-paced industry. Ghidra’s success illustrates the utility of C&K metrics in guiding software development and maintenance practices in large-scale projects.

**References:**

Aniche, M. (2015). CK. GitHub. Retrieved May 27, 2024 from https://github.com/mauricioaniche/ck

National Security Agency. (2019). Ghidra. GitHub. Retrieved May 27, 2024 from https://github.com/NationalSecurityAgency/ghidra