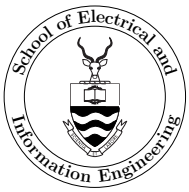

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG
School of Electrical and Information Engineering
ELEN4010 - Software Development III

Individual Project — 2018

Code Metrics

1 Introduction

Code metrics are a subset of *software metrics*. Software metrics are defined as follows [1]:

> A software metric is a measure of software characteristics which are quantifiable or countable. Software metrics are important for many reasons, including measuring software performance, planning work items, measuring productivity, and many other uses.

Code metrics are focused specifically on determining source code characteristics and are generally extracted using static analysis tools. There are numerous code metrics that have been developed [2, 3], the most basic of which is Lines of Code (LOC).

2 Requirements

For this project you are required to investigate code metrics in some detail. In particular, you must

1. Identify five code metrics in addition to LOC and explain what each of these metrics are measuring and any caveats that must be taken into account when using these metrics.
2. Choose a tool (or tools) that can be used to calculate the metrics that you have identified above and run this tool against a small test code base for which you can manually calculate the metrics. You can then verify if the tool is working according to your understanding of the measurements.
3. Calculate your chosen metrics (using the tool) for three, or more, *major* releases of an open-source code base and critically analyse and interpret your findings. It will be helpful to understand the open-source project itself, and what the differences are between the releases, as this will inform your understanding of the derived metrics. The code base that you choose may be written in any language other than C++.

3 Deliverables

This is an individual assignment — each student is required to work on their own.

3.1 Project Report

Submit via Sakai a technical report of no more than 6 pages in length, excluding title page, abstract, references and appendices. The report must address the requirements as per Section 2.

You must include the following appendices:

- a listing of the source code of the test code base, and
- instructions on how to set up and run the analysis tool which was used to derive the metrics. This should include any custom scripts/code which was developed.

4 Deadline and Submissions

The submission date for all deliverables is specified in the "4th Year Courses — Project Hand-In Dates 2018" document.

All submissions must be in strict accordance with the guidelines contained in the *School's Blue Book* and the rules contained in the *School's Red Book*.

5 Assessment

Your report will be assessed based on the following factors:

- Your understanding of the code metrics that you have chosen and their usage.
- The effort and rigour that you have applied in verifying the behaviour of the analysis tool that you have chosen.
- The production, presentation, and critical analysis of the results that are generated for the code base that you have analysed.
- The number and quality of sources investigated and referenced in the report.
- The quality of the report as a professional and technical document.

The project assessment form is available on the course home page.

5.1 Plagiarism

Refer to the *School's Blue Book* for an explanation of what plagiarism is and how to avoid it.

All instances of plagiarism from either the internet or within the class will be severely dealt with. No two students may have identical or overly similar reports.

References

[1] Stackify. What are software metrics and how can you track them? `https://stackify.com/track-software-metrics/`, Last accessed: 15/02/2018.

[2] Metrics overview. `http://staff.unak.is/andy/StaticAnalysis0809/metrics/overview.html`, Last accessed: 15/02/2018.

[3] Wikipedia. Software metric. https://en.wikipedia.org/wiki/Software_metric, Last accessed: 15/02/2018.