Design and Software Quality Metrics

Exploring the Relationship between Design Metrics and Software Diagnosability using Machine Learning (2018)

Thomas Dornberger Sofie Kemper

2018-05-31

1 Metrics

1.1 Bug Characteristics

The following metrics are related to a known bug and aim to quantify features of the bug such as its location and size. The following statistics are taken from the Defects4J Dissection (see 2.1).

In addition, for all static metrics analysed via TeamScale on the project level, we define buggy-file versions (prefix BF-) which measure the corresponding statistic only on the files that are responsible for the bug, i.e., those files that are modified in the bug's corresponding patch. The aggregation of these metrics for multiple files is achieved in the same way as the project-level aggregation, i.e., some findings are summed up whereas others are normalised by the respective source lines of code or the maximum values are used (e.g., maximum cyclomatic complexity number).

On top of that, the dynamic metrics we defined on the project level are also extended to the bug level (prefix B-) by looking at bug-specific properties of the call- and data-dependency graphs, e.g., the in-degree of the vertices which correspond to the faulty methods. In cases where there are multiple faulty methods, the "most extreme" value is used, e.g., the maximum in-degree or the minimum eccentricity.

Other ideas such as analysing whether the bug is part of a design antipattern or calculating object-oriented metrics on the bug method were rejected due to their high effort and low presumed meaningfulness for the software diafnosability.

1.1.1 Number of Files (B-NF)

This metric quantifies the number of files modified in the patch that fixes the bug.

1.1.2 Number of Classes (B-NC)

The number of classes changed in the patch is given by this metric.

1.1.3 Number of Methods (B-NM)

This metric indicates the number of methods modified in the patch.

1.1.4 Number of Lines (B-NL)

The metric counts the total number of lines modified in the patch.

1.1.5 Number of Lines Added/Removed/Modified (B-NAL/B-NRL/B-NML)

These metrics indicate the number of lines (of code) that were added, removed, or modified, respectively, in the patch. The sum of these three metrics corresponds to the number of lines (B-NL).

1.1.6 Number of Chunks (B-NCH)

This metric indicates the number of "chunks" modified in the patch. A chunk is defined as a section in the code containing sequential line changes.

1.1.7 Number of Repair Patterns (B-NRP)

This metric quantifies the number of different "repair patterns" employed in the patch for the corresponding bug. A repair pattern is a general pattern of fixing a bug, i.e., a characteristic of the bug patch. Examples include "Single Line", indicating that only a single line was changed, or "Wrong Variable Reference".

1.1.8 Number of Repair Actions (B-NRA)

The number of repair actions counts the number of different actions employed in the process of fixing the bug, i.e., visible in the patch diff. Examples of repair actions include "Variable replacement by another variable" or "Conditional (if) branch addition".

2 Tools

We have tried a multitude of different tools (SourceMeter, LOCC, SonarQube, CCCC, USC CodeCount, CLOC, etc.) and chosen the following for the interesting metrics they provide as well as the possibility to automate their usage, i.e., the possibility to execute them on the command line and output formats that can be easily parsed and used for our purposes.

2.1 Defects4J Dissection

The website `http://program-repair.org/defects4j-dissection/#!/` provides data that describe the defects4j bug dataset. For all buggy versions in defects4j it provides summary statistics of the bug. Additionally, the corresponding patch can be accessed.