Assignment 02

Software Quality Metrics

Understand is a commercial tool that calculates various metrics for your code to help you pinpoint complex and potentially problematic areas in the source code. This allows you to stay continuously aware of the health of your codebase. You may also export the metrics to HTML, CSV, or XML format for further analysis. Metrics are useful indicators of unhealthy code than as indicators of healthy code. A codebase with many range violation warnings is probably an indication that the code needs to be refactored, but no range violation warnings do not necessarily mean that the code is good.

Students can download *Understand* for **free**. Create an account using your university email address: https://www.scitools.com/student

Tasks

1. Utilize an **open-source Java or C#** project that has a long history of development with at least 6 releases/versions. You can use https://seart-ghs.si.usi.ch/ to search for a project; use the search parameters in the Appendix
2. Choose between 6 to 10 versions/releases of the project that is spread across months (or years) so that there is a noticeable difference in the metrics.
    a. Optional, but highly recommended, you can perform a T-Test to see if there is a statistical difference between the LOC of the two releases. In the report, you can specify the result of each t-test calculation
3. Use the tool *Understand* to calculate the metrics for each of the releases. The results should be saved in an Excel file. Use the values of these metrics to plot the variation of metrics in a line chart throughout the evolution of the software. Refer to the Appendix for metric definitions. The metrics to extract are:
    a. **Complexity Metrics**:
        i. Cyclomatic
        ii. RatioCommentToCode
        iii. MaxInheritanceTree
    b. **Object Oriented Metrics**:
        i. CountDeclInstanceMethod
        ii. CountDeclInstanceVariable
        iii. CountDeclMethod
        iv. CountClassCoupled
        v. PercentLackOfCohesion
    c. **Count Metrics**: *Pick a few metrics of your choice*

4. Use these charts to evaluate the evolution of the project from a quality perspective:
    a. Comment on each of these charts individually to describe the evolution of each metric.
    b. Using all these charts combined, try to locate any patterns that can be seen as indicators of good or bad quality and use them to give insights about any possible recommendations for developers in the future.
5. The next task is to determine which of these metrics has been drastically evolving in the versions that you have considered for this assignment. For each metric, compare its values in the first version with its values in the last version. The comparison is performed using the Mann-Whitney U Test test to verify which metric's difference between the two compared versions is found to be statistically significant.
    - For example, let us consider the first version (e.g., v1.0) and the last version (e.g., v8.5). v1.0 contains 5 classes, and v8.5 contains 9 classes. If we are comparing the LOC between these two versions, then we will end up with a column of 5 values representing v1.0, and another column of 9 values representing v8.5. You can hypothesize that v8.5 values are higher than v1.0 and verify of your hypothesis holds and whether the p-value is statistically significant.

Submission Artifacts

You need to submit it two artifiacts:

1. A spreadsheet containing the raw data for Task #3 and Task#5
2. A pdf of your writeup

You do not need to submit the source code of the releases. The pdf should contain hyperlinks to the project repository.

Deadline:

Refer to the course schedule document

Grading

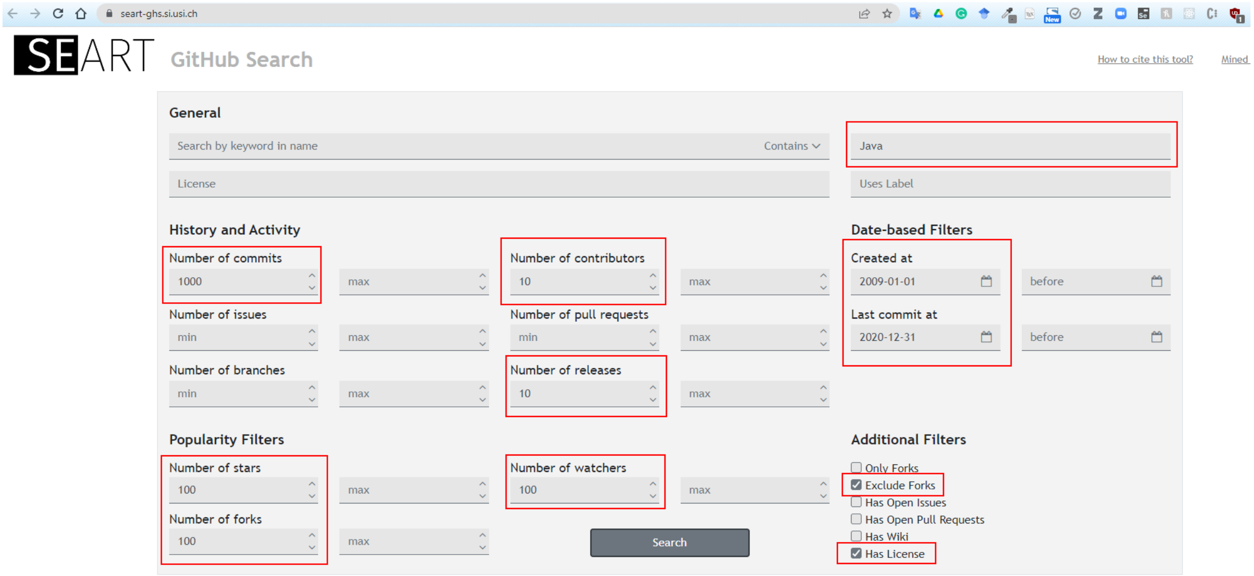
Task #3: 50%
Task #4: 20%
Task #5: 30%

Sample

A sample writeup is provided in the Assignment 02 directory in Laulima. Do keep in mind that this is just a sample to show how you would need to construct the writeup. Not all the metrics/charts in the sample are the same as what is required in Task #3.

Appendix

Searching for a project



General

Search by keyword in name: Contains
Java
License
Uses Label

History and Activity

Number of commits: 1000 – max
Number of contributors: 10 – max
Number of issues: min – max
Number of pull requests: min – max
Number of branches: min – max
Number of releases: 10 – max

Date-based Filters

Created at: 2009-01-01 – before
Last commit at: 2020-12-31 – before

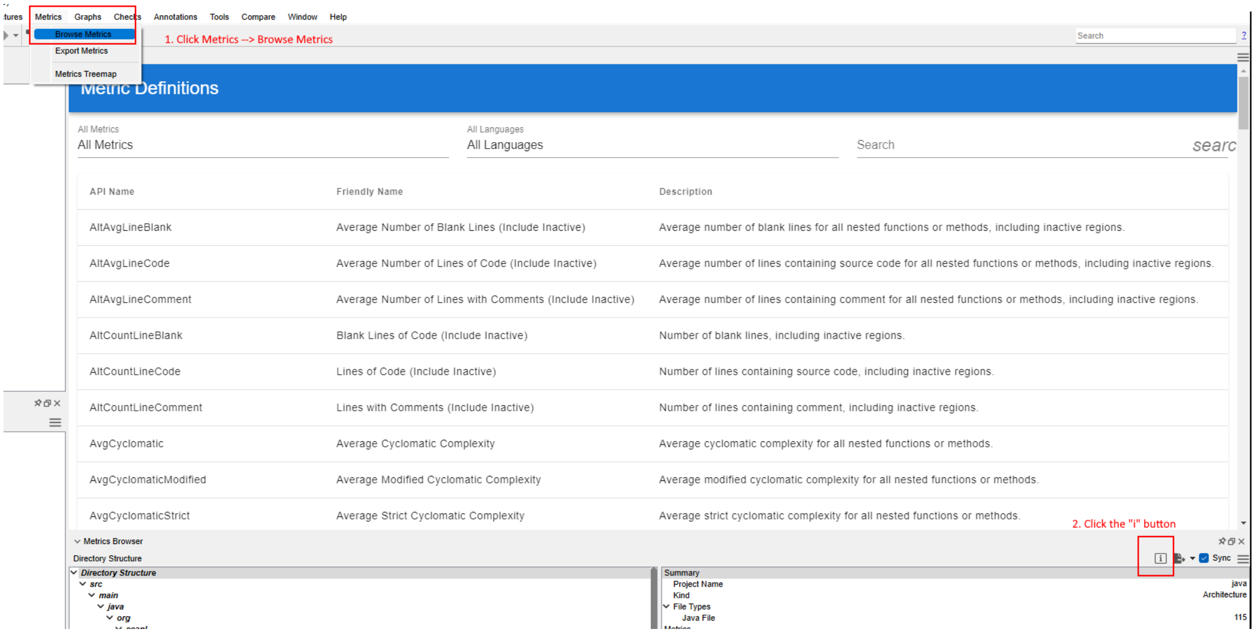
Popularity Filters

Number of stars: 100 – max
Number of watchers: 100 – max
Number of forks: 100 – max

Additional Filters

- ☐ Only Forks
- ☑ Exclude Forks
- ☐ Has Open Issues
- ☐ Has Open Pull Requests
- ☐ Has Wiki
- ☑ Has License

Metric Definitions



1. Click Metrics --> Browse Metrics

API Name	Friendly Name	Description
AltAvgLineBlank	Average Number of Blank Lines (Include Inactive)	Average number of blank lines for all nested functions or methods, including inactive regions.
AltAvgLineCode	Average Number of Lines of Code (Include Inactive)	Average number of lines containing source code for all nested functions or methods, including inactive regions.
AltAvgLineComment	Average Number of Lines with Comments (Include Inactive)	Average number of lines containing comment for all nested functions or methods, including inactive regions.
AltCountLineBlank	Blank Lines of Code (Include Inactive)	Number of blank lines, including inactive regions.
AltCountLineCode	Lines of Code (Include Inactive)	Number of lines containing source code, including inactive regions.
AltCountLineComment	Lines with Comments (Include Inactive)	Number of lines containing comment, including inactive regions.
AvgCyclomatic	Average Cyclomatic Complexity	Average cyclomatic complexity for all nested functions or methods.
AvgCyclomaticModified	Average Modified Cyclomatic Complexity	Average modified cyclomatic complexity for all nested functions or methods.
AvgCyclomaticStrict	Average Strict Cyclomatic Complexity	Average strict cyclomatic complexity for all nested functions or methods.

2. Click the "i" button

```
Directory Structure
 src
  main
   java
    org
```

Summary	
Project Name	java
Kind	Architecture
File Types	
Java File	115