selectel.meetup

Code metrics: why do we need them?

Mehti Musayev
Software Developer at Selectel, Ltd.

About me

selectel.meetup

- Education:
  - Saint Petersburg Presidential Physics and Mathematics Lyceum №239
  - Saint Petersburg State University, Mathematics and Mechanics Faculty
- Work experience:
  - 2011 - 2011, CoFiTe, Software Developer Intern
  - 2015 - 2016, Digital Security, Junior Research Software Engineer at Laboratory
  - 2016 - present, Selectel, Junior Software Developer
- GitHub contributions:
  - https://github.com/google/yapf, a formatter for Python files, owned by Google Inc.
  - https://github.com/pycqa/redbaron, Bottom-up approach to refactoring in python, Python Code Quality Authority
  - https://github.com/PyCQA/baron, writing refactoring code, Python Code Quality Authority

Software metric

selectel.meetup

- a software metric is a standard of measure of a degree to which a software system or process possesses some property

- metrics are functions, while measurements are the numbers obtained by the application of metrics (often metrics and measurements are used as synonymous)

Main usages

selectel.meetup

- scheduling
- software sizing
- programming complexity
- software development effort estimation
- software quality

Main metrics

selectel.meetup

- code coverage
- cohesion
- cyclomatic complexity (McCabe's complexity)
- Halstead complexity
- the maintainability index
- raw metrics (SLOC, comment lines, blank lines, &c.)

Code coverage

selectel.meetup

- a measure used to describe the degree to which the source code of a program is executed when a particular test suite runs

- higher coverage => more of program's source code executed during testing => a lower chance of containing undetected software bugs

Popular code coverage systems

selectel.meetup

- Coveralls - https://coveralls.io
- Coco (C/C++/C#/Tcl) - https://www.froglogic.com/coco/
- SonarQube (most famous languages) - http://www.sonarqube.org/

Cohesion

selectel.meetup

- degree to which the elements of a module belong together
- i.e., cohesion measures the strength of relationship between pieces of functionality within a given module
- in highly cohesive systems functionality is strongly related

Cyclomatic complexity (McCabe's complexity)

selectel.meetup

- developed by Thomas J. McCabe, Sr. in 1976
- Measurement which used to indicate the complexity of a program
- a quantitative measure of the number of linearly independent paths through a program's source code

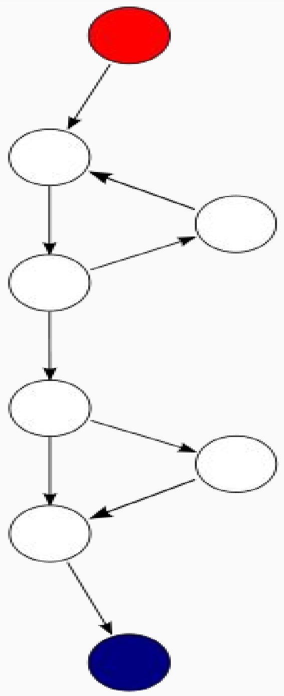
Cyclomatic complexity (McCabe's complexity)

- computed using the control flow graph of the program
- may be applied to individual functions, modules, methods or classes within a program

Cyclomatic complexity (McCabe's complexity)

selectel.meetup

- M = E - N + 2P
  - E = the number of edges of the graph.
  - N = the number of nodes of the graph.
  - P = the number of connected components.



Halstead complexity

selectel.meetup

- introduced by Maurice Howard Halstead in 1977
- depends on number of operators and operands in program
- the goal was to identify measurable properties of software, and the relations between them

Halstead complexity (calculation)

selectel.meetup

- $n = n_1 + n_2$ - program vocabulary
- $N = N_1 + N_2$ - program length
- $N_{\wedge} = n_1 * \log2(n_1) + n_2 * \log2(n_2)$ - calculated program length
- $V = N * \log2(n)$ - program volume
- $D = (n_1/2) * (N_2/n_2)$ - program difficulty
- $E = D * V$ - program effort

Where:

- n_1 - the number of distinct operators
- n_2 - the number of distinct operands
- N_1 - the total number of operators
- N_2 - the total number of operands

Maintainability index

selectel.meetup

- measures how maintainable the source code is
- calculated from lines-of-code measures, cyclomatic complexity measures and Halstead complexity measure
- used in several automated software metric tools such as MS Visual Studio

Maintainability index (the original formula)

selectel.meetup

$$MI = 171 - 5.2 \ln V - 0.23G - 16.2 \ln L$$

Where:

- V is the Halstead Volume
- G is the total Cyclomatic Complexity
- L is the number of Source Lines of Code (SLOC)

Maintainability index (the derivative used by SEI)

selectel.meetup

$$MI = 171 - 5.2\log_2 V - 0.23G - 16.2\log_2 L + 50\sin(\sqrt{2.4C})$$

Where:

- V is the Halstead Volume
- G is the total Cyclomatic Complexity
- L is the number of Source Lines of Code (SLOC)
- C is the percent of comment lines (important: converted to radians).

Maintainability index
(the derivative used by Visual Studio)

selectel.meetup

$$MI = \max\left[0, 100\,\frac{171 - 5.2\ln V - 0.23G - 16.2\ln L}{171}\right].$$

Where:

- V is the Halstead Volume
- G is the total Cyclomatic Complexity
- L is the number of Source Lines of Code (SLOC)
- C is the percent of comment lines (important: converted to radians).

Raw metrics

selectel.meetup

- LOC: The total number of lines of code. It does not necessarily correspond to the number of lines in the file.
- LLOC: The number of logical lines of code. Every logical line of code contains exactly one statement.
- SLOC: The number of source lines of code - not necessarily corresponding to the LLOC.

Raw metrics

selectel.meetup

- Comments: The number of comment lines. Multi-line strings are not counted as comment since, to the Python interpreter, they are just strings.
- Multi: The number of lines which represent multi-line strings.
- Blanks: The number of blank lines (or whitespace-only ones).

What to use to calculate measures?

selectel.meetup

- Radon (for Python code) - https://github.com/rubik/radon
- SonarQube (>20 languages) - https://blog.sonarsource.com/
- SourceMonitor (C++, C, C#, VB.NET, Java, Delphi, Visual Basic (VB6) or HTML) - http://www.campwoodsw.com/sourcemonitor.html
- CAST - http://www.castsoftware.com/Product/Application-Intelligence-Platform.aspx
- Klocwork Insight - http://www.klocwork.com/products-services/klocwork

Why?

selectel.meetup

- have numerous valuable applications in schedule
- planning budget
- cost estimation
- quality assurance testing
- software debugging
- software performance optimization
- optimal personnel task assignments

Demo

selectel.meetup

- Using Radon as example
- Radon is a Python tool which computes various code metrics

Remark for measuring in Radon

selectel.meetup

- Radon measures index with another formula
- It gets derivation from SEI and MS VS:

$$MI = \max\left[0, 100\frac{171 - 5.2\ln V - 0.23G - 16.2\ln L + 50\sin(\sqrt{2.4C})}{171}\right]$$

References

selectel.meetup

- Joan C. Miller, Clifford J. Maloney (February 1963). "Systematic mistake analysis of digital computer programs". Communications of the ACM. New York, NY, USA
- McCabe (December 1976). "A Complexity Measure". IEEE Transactions on Software Engineering: 308–320.
- Halstead, Maurice H. (1977). Elements of Software Science. Amsterdam: Elsevier North-Holland, Inc.
- Paul Omand and Jack Hagemeister. “Metrics for assessing a software system’s maintainability”. Proceedings International Conference on Software Mainatenance (ICSM), 1992. (doi)
- Don M. Coleman, Dan Ash, Bruce Lowther, Paul W. Oman. Using Metrics to Evaluate Software System Maintainability. IEEE Computer 27(8), 1994. (doi, postprint)
- Maintainability Index Range and Meaning. Code Analysis Team Blog, blogs.msdn, 20 November 2007.
- Arie van Deursen, Think Twice Before Using the “Maintainability Index”.

Thank you!

selectel.meetup

- Contact information:
  - Email: musayev[at]selectel(dot)ru
  - Telegram: https://telegram.me/me_mu, @me_mu

- GitHub: https://github.com/b5y
- Code examples: https://github.com/b5y/selectel_meetup_2016