

## SWM 43 – Research in Computing - Assignment 3

### Topic

---

Object-Oriented Product Metrics

### Bibliography

---

Ordered (1) primarily by star rating (starting with most relevant), (2) secondarily by author's last name (alphabetically).

\*\*\*

[1] Abreu, Fernando B., *Talk on "Design Metrics for Object-Oriented Software Systems"*. 7th ERCIM Database Research Group Workshop on Object Oriented Databases, 1995. [URL: <http://www.ercim.org/publication/ws-proceedings/7th-EDRG/18.pdf>]

Very concise (3page) condensed overview by a competent researcher in the field.

[2] Fenton, Norman E. and Martin Neil, *Software Metrics: Roadmap*. The Future of Software Engineering. ACM Press, New York, 2000.

Introduces the whole field of software metrics, including history of software metrics, various types, kinds, branches...

[3] Harrison, R., Counsell, S.J., Nithi, R.V., *An evaluation of the MOOD Set of Object-Oriented Software Metrics*. IEEE Transactions on Software Engineering. Volume 24, Issue 6, Jun 1998. pp. 491-496

A theoretical evaluation of the MOOD metrics originally proposed by Abreu. Some flaws pointed out, but generally the MOOD set is accepted. A brief, concise, well structured paper.

[4] Kuipers, Tobias and Joost Visser, *Maintainability Index Revisited - position paper*. System Quality and Maintainability (SQM 2007), satellite of CSMR 2007. [URL: <http://www.di.uminho.pt/~joostvisser/publications/MaintainabilityIndexRevisited.pdf>]

Short, to the point, really absorbing, (in my opinion valid) criticism of cyclomatic complexity, claiming CC is not exactly well suited for modern OO designs and/or languages such as Java

[5] Manns, Tom and Michael Coleman, *Software Quality Assurance*. New York: Macmillan, 1996.

Chapter 5 (pp. 95-126) titled Metrics for the Quality Manager covers the most useful time-tested metrics used in software development. Especially section 5.1 (Basic Concepts) seems like an interesting read, as it explains in a concise manner how software metrics should be interpreted, what's the motivation to use metrics on a (software) project, problems with capturing the data, and validation of metrics.

[6] Pressman, Roger, *Software Engineering: A practitioner's approach*, 6th edition. New York: McGraw-Hill, 2005.

Chapter 15 (pp. 461-497) titled Product Metrics provides an excellent summary of the subject area. Highly comprehensive, concise, good read.

[7] Rosenberg, Linda H., *Applying and Interpreting Object Oriented Metrics*. [online] Available from: [http://satc.gsfc.nasa.gov/support/STC\\_APR98/apply\\_oo/apply\\_oo.html](http://satc.gsfc.nasa.gov/support/STC_APR98/apply_oo/apply_oo.html) [Accessed on: 01-Dec-2009]

"The metrics are first defined, then using a very simplistic object oriented example, the metrics are applied. Interpretation guidelines are then discussed and data from NASA projects are used to demonstrate the application of the metrics. [...] The purpose of this paper is to help project managers choose a comprehensive set of metrics, not by default, but by using a set of metrics based on attributes and features of object oriented technology."

\*\*

[8] Aggarwal K.K., Singh Y., Kaur A. and Malhotra R., *Software Design Metrics for Object-Oriented Software*. Journal of Object Technology. Volume 6, Issue 1, Jan-Feb 2006. pp. 121-138 [URL: [http://www.jot.fm/issues/issue\\_2007\\_01/article4](http://www.jot.fm/issues/issue_2007_01/article4)]

The title of the paper does not really describe its content, as the paper is very much concentrating mainly on exceptions, program robustness, Java exception handling mechanism... Authors propose two metrics which are "... analytically evaluated against Weyuker's proposed set of nine axioms. These set of metrics are calculated and analyzed for standard projects and accordingly ways in which project managers can utilize these metrics are suggested." This evaluation of axioms in particular could be interesting to have a look at.

[9] Chidamber, Shyam R. and Chris F. Kemerer, *A Metrics Suite For Object Oriented Design*. IEEE Transactions on Software Engineering. Volume 20, Issue 6, Jun 1994. pp. 476-493

The metrics suite as proposed by Chidamber and Kemerer, known as the CK metrics suite. The original paper published in IEEE Transactions on Software Engineering in 1994.

[10] Gorman, Jason, *OO Design Principles & Metrics*. 2006 [online] Available from: <http://www.parlezuml.com/metrics/OO%20Design%20Principles%20&%20Metrics.pdf> [Accessed on: 03-Dec-2009]

A nice slideshow presentation showing some bad designs, how to refactor them and what METRICS to apply to avoid bad design.

[11] Henderson-Sellers, Brian. *Object-oriented metrics: Measures of Complexity*. New Jersey: Prentice Hall PTR, 1996.

This is a book often referred to by people in the field. Chapter 6 (pp. 109-164) titled Object Oriented Product Metrics covers the area I'm most interested in in good detail.

[12] McGarry J., Card D., Jones C., Layman B., Clark E., Dean J., Hall F., *Practical Software Measurement: Objective Information for Decision Makers*. Boston: Addison-Wesley, 2001.

Last chapter of this book, Chapter 8 (pp. 155-158) titled Measure for Success includes an analysis of how to successfully apply software measurement and why do some projects fail to measure effectively. In just three pages a lot of interesting conclusions are reached.

[13] Mäkelä, Sami and Ville Leppänen, *Client-based Cohesion Metrics for Java Programs*. Science of Computer Programming 74 (2009) pp. 355-378

nice to follow the methodology

[14] Nagappan N., Williams L., Vouk M., Osborne J., *Early Estimation of Software Quality Using In-process Testing Metrics*. In Third Software Quality Workshop, co-located with the International Conference on Software Engineering (ICSE 2005), pages 46–52, May 2005.

STREW metrics suite developed by a group of researchers, test-oriented metrics

[15] Sarker, Muktamye, *An Overview of Object Oriented Design Metrics*. [online] Available from: <http://www.cs.umu.se/education/examina/Rapporter/MuktamyeSarker.pdf> [Accessed on: 03-Dec-2009]

this paper is interesting for its rather wide take on the subject. Many metrics are mentioned, explained, and in an appendix there is a list of many various metrics. Could be possibly good to give it a read.

[16] Subramanian, Girish and William Corbin, *An Empirical Study of Certain OO Software Metrics*. The Journal of Systems and Software 59 (2001) pp. 57-63

nice study of various metrics. Quantitative research, numbers, tables, stats...

[17] Watson, Arthur H. and Thomas J. McCabe, *Structured testing: A Testing Methodology Using The Cyclomatic Complexity Metric*. Technical Report 500-235, NIST Computer Systems Laboratory, 1996.

Very well written by very competent researchers (renowned McCabe), the case studies in the appendix "[provide] empirical evidence by presenting several case studies from the software engineering literature."

\*

[18] Briand L., Wüst J., Daly J., Porter V., *A Comprehensive Empirical Validation of Product Measures for Object-Oriented Systems*. Journal of Systems and Software 51, 245-273, 2000.

Section 2.4.3. (Data Collected) lists the numerous measures used in this study and provides a definition and source of these. The whole document is dedicated to studying these measures, enormous tables containing various correlations and results of Univariate Analysis.

[19] Briand L., Wüst J., *Empirical Studies of Quality Models in Object-Oriented Systems*. Advances in Computers Vol. 59, 97-166, 2002.

Another which is very data-heavy and extremely hard to follow, however the tables in the appendix "describe the coupling, cohesion, inheritance, and size measures mentioned in this [paper]. [It lists the] acronym used for each measure, informal definitions of the measures, and literature references where the measures originally have been proposed. The informal natural language definitions of the measures should give the reader a quick insight into the measures." I am after the 'quick insight into the measures', so this appendix could prove to be quite useful.

[20] El-Emam K., Benlarbi S., Goel N., Melo W., Lounis H., and Shesh N. Rai, *The Optimal Class Size for OO Software*. IEEE Transactions on Software Engineering. Volume 28, Issue 5, May 2002. pp. 494-509

Appendix explains Box and Whiskers plots, which I might use. Also the paper as a whole could be interesting, but is rather too lengthy. Systems investigated are C++, C++ and Java, which is fine

[21] Gencel, Cigdem and Onur Demirors, *Functional Size Measurement Revisited*. ACM Transactions on Software Engineering Methodology, Volume 17, Issue 3, Article 15 (June 2008), 36 pages.

Absorbing abstract and intro, could be good to skim to perhaps get more ideas for my research.

[22] Kan, Stephen. *Metrics and Models in Software Quality Engineering*. Boston: Addison-Wesley, 1995.

Chapter 10 (pp.253-272) of this book covers Complexity Metrics and Models, and the book as a whole is dedicated to metrics for software quality engineering, so a great variety of metrics, not only product metrics, are introduced.

[23] McCabe, Thomas J., *A Complexity Measure*. IEEE Transactions on Software Engineering. Volume SE-2, Dec 1976. pp. 308-320

The classic. The rally BIG software metric, old paper, but very sound methodology based in graph theory, heavily mathematical in nature.

[24] Nagpurkar P. and C. Krintz, *Phase-based Visualization and Analysis of Java Programs*. Science of Computer Programming 59 (2006) pp. 64–81

The visualisation is interesting, would be nice if I could somehow try this.

[25] Nasser E., Counsell S., Shepperd M., *Class Movement and Re-location: An Empirical Study of Java Inheritance Evolution*. The Journal of Systems and Software xxx (2009) [Article in Press]

this could possibly be interesting, as the researchers are actually examining four concrete Java products and the report contains nice tables, graphs, etc...

[26] SDMetrics User Manual, Appendix B.1: *Class Metrics* [online] Available from: [http://www.sdmetrics.com/manual/Metrics\\_class.html](http://www.sdmetrics.com/manual/Metrics_class.html) [Accessed on: 03-Dec-2009]

This website provides a long list of various class metrics and their brief descriptions, which could be good to have a look at to gain an idea what's already been developed in the area.