

Universidade Federal da Bahia
Instituto de Matemática

Programa de Pós-graduação em Ciência da Computação

**CARACTERIZAÇÃO DA COMPLEXIDADE
ESTRUTURAL EM FERRAMENTAS DE
ANÁLISE ESTÁTICA DE CÓDIGO-FONTE**

Joenio Marques da Costa
joenio@joenio.me

QUALIFICAÇÃO DE MESTRADO

Salvador
08 de Julho de 2016

Universidade Federal da Bahia
Instituto de Matemática

Joenio Marques da Costa
joenio@joenio.me

CARACTERIZAÇÃO DA COMPLEXIDADE ESTRUTURAL EM FERRAMENTAS DE ANÁLISE ESTÁTICA DE CÓDIGO-FONTE

*Trabalho apresentado ao Programa de Pós-graduação em
Ciência da Computação do Instituto de Matemática da
Universidade Federal da Bahia como requisito parcial para
obtenção do grau de Mestre em Ciência da Computação.*

Orientadora: Profa. Dra. Christina von Flach G. Chavez
Co-orientador: Prof. Dr. Paulo Roberto Miranda Meirelles

Salvador
08 de Julho de 2016

AGRADECIMENTOS

(pendente)

RESUMO

(pendente)

Palavras-chave:

(pendente)

ABSTRACT

(pendente)

Keywords:

(pendente)

SUMÁRIO

Capítulo 1—Introdução	1
1.1 Motivação	1
1.2 Objetivos	1
Capítulo 2—Análise estática de código-fonte	3
2.1 Qualidade de software	3
2.2 Ferramentas de análise estática	3
2.3 Software científico	3
2.4 Contribuições esperadas	3
2.5 Trabalhos relacionados	3
Capítulo 3—Métricas de código-fonte	5
3.1 Complexidade estrutural	5
3.2 Contribuições esperadas	5
3.3 Trabalhos relacionados	5
Capítulo 4—Metodologia	7
4.1 Hipóteses	7
4.2 Coleta de dados	7
4.2.1 Revisão estruturada	7
4.3 Análise de dados	7
4.3.1 Distribuição dos valores das métricas	7
Capítulo 5—Caracterização das ferramentas	9
5.1 Resultados	9
Capítulo 6—Exemplo de uso	11
6.1 Analizo	11
Capítulo 7—Conclusão	13
7.1 Contribuições	13
7.2 Trabalhos futuros	13

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO 1

INTRODUÇÃO

1.1 MOTIVAÇÃO

(pendente)

1.2 OBJETIVOS

(pendente)

CAPÍTULO 2

ANÁLISE ESTÁTICA DE CÓDIGO-FONTE

(pendente)

2.1 QUALIDADE DE SOFTWARE

(pendente)

2.2 FERRAMENTAS DE ANÁLISE ESTÁTICA

(pendente)

2.3 SOFTWARE CIENTÍFICO

(pendente)

2.4 CONTRIBUIÇÕES ESPERADAS

(pendente)

2.5 TRABALHOS RELACIONADOS

(pendente)

CAPÍTULO 3

MÉTRICAS DE CÓDIGO-FONTE

(pendente)

3.1 COMPLEXIDADE ESTRUTURAL

(pendente)

3.2 CONTRIBUIÇÕES ESPERADAS

(pendente)

3.3 TRABALHOS RELACIONADOS

(pendente)

CAPÍTULO 4

METODOLOGIA

4.1 HIPÓTESES

(pendente)

4.2 COLETA DE DADOS

(pendente)

4.2.1 Revisão estruturada

(pendente)

4.3 ANÁLISE DE DADOS

(pendente)

4.3.1 Distribuição dos valores das métricas

(pendente)

CAPÍTULO 5

CARACTERIZAÇÃO DAS FERRAMENTAS

(pendente)

5.1 RESULTADOS

(pendente)

CAPÍTULO 6

EXEMPLO DE USO

(pendente)

6.1 ANALIZO

(pendente)

CAPÍTULO 7

CONCLUSÃO

7.1 CONTRIBUIÇÕES

(pendente)

7.2 TRABALHOS FUTUROS

(pendente)

REFERÊNCIAS BIBLIOGRÁFICAS

- ALMEIDA, L. T.; MIRANDA, J. M. de. Código limpo e seu mapeamento para métricas de código fonte. *Monografia de Graduação em Ciência da Computação*, p. 74, 2010. Disponível em: <http://www.ime.usp.br/~cef/mac499-10/monografias/lucianna-joao/arquivos/monografia.pdf>.
- BECK, K. *Implementation Patterns*. [S.l.]: Pearson Education, 2007.
- BINKLEY, D. Source code analysis: A road map. In: IEEE COMPUTER SOCIETY. *2007 Future of Software Engineering*. [S.l.], 2007. p. 104–119.
- CERT. *Secure Coding Tools*. 2016. [Online; acessado 23 Abril de 2016]. Disponível em: <http://www.cert.org/secure-coding/tools/index.cfm>.
- CHIDAMBER, S. R.; KEMERER, C. F. A metrics suite for object oriented design. *Software Engineering, IEEE Transactions on*, IEEE, v. 20, n. 6, p. 476–493, 1994.
- DARCY, D. P. et al. The structural complexity of software an experimental test. *Software Engineering, IEEE Transactions on*, IEEE, v. 31, n. 11, p. 982–995, 2005.
- HARRISON, R.; COUNSELL, S. J.; NITHI, R. V. An evaluation of the mood set of object-oriented software metrics. *Software Engineering, IEEE Transactions on*, IEEE, v. 24, n. 6, p. 491–496, 1998.
- HITZ, M.; MONTAZERI, B. *Measuring Coupling and Cohesion In Object-Oriented Systems*. [s.n.], 1995. Disponível em: <http://www.isys.uni-klu.ac.at/PDF/1995-0043-MHBM.pdf>.
- ISO, I. Iec25010: 2011 systems and software engineering—systems and software quality requirements and evaluation (square)—system and software quality models. *International Organization for Standardization*, p. 34, 2011.
- JÚNIOR, M. R. P. Estudo de métricas de código fonte no sistema android e seus aplicativos. p. 82, 2015. Disponível em: <https://fga.unb.br/tcc/software/tcc-2015.1-engenharia-de-software/marcos-ronaldo-pereira-junior/v3-tcc.pdf>.
- KIRKOV, R.; AGRE, G. Source code analysis - an overview. *Cybernetics and Information Technologies*, v. 10, n. 2, p. 60–77, 2010.
- KON, F. et al. Free and open source software development and research: Opportunities for software engineering. In: *SBES*. [s.n.], 2011. p. 82–91. Disponível em: <http://dblp.org/db/conf/sbes/sbes2011.html\\#KonMLTCM11>.

KRISHNAMURTHI, S.; VITEK, J. The real software crisis: Repeatability as a core value. *Communications of the ACM*, ACM, v. 58, n. 3, p. 34–36, 2015.

MAFRA, S. N.; TRAVASSOS, G. H. Estudos primários e secundários apoiando a busca por evidência em engenharia de software. 2006.

MARTIN, R. C.; HAN, L. *Clean Code*. [S.l.]: Publishing House of Electronics Industry, 2012.

MCCABE, T. J. A complexity measure. *Software Engineering, IEEE Transactions on*, IEEE, n. 4, p. 308–320, 1976.

MCCONNELL, S. *Code Complete*. 2nd. ed. [S.l.]: Microsoft Press, 2004.

MEIRELLES, P. R. M. *Monitoramento de métricas de código-fonte em projetos de software livre*. Tese (Doutorado) — Universidade de São Paulo, São Paulo, Brazil, 2013.

NIST. *SAMATE - Source Code Security Analyzers*. 2016. [Online; acessado 20 Abril de 2016]. Disponível em: http://samate.nist.gov/index.php/Source_Code_Security_Analyzers.html.

PORTILLO-RODRÍGUEZ, J. et al. Tools used in global software engineering: A systematic mapping review. p. 663–685, 2012. Disponível em: <http://dblp.org/db/journals/infsof/infsof54.html\#Portillo-RodriguezVPB12>.

RIBEIRO, A. C. Análise estática de código-fonte com foco em segurança: Metodologia para avaliação de ferramentas. 2015.

ROSENBERG, L. H.; HYATT, L. E. Software quality metrics for object-oriented environments. *Crosstalk journal*, v. 10, n. 4, 1997.

SHARBLE, R. C.; COHEN, S. S. The object-oriented brewery: a comparison of two object-oriented development methods. *ACM SIGSOFT Software Engineering Notes*, ACM, v. 18, n. 2, p. 60–73, 1993.

SHIH, T. K. et al. Decomposition of inheritance hierarchy dags for object-oriented software metrics. In: *Engineering of Computer-Based Systems, 1997. Proceedings., International Conference and Workshop on*. [S.l.: s.n.], 1997. p. 238–245.

SOCIETY, I. C. *Guide to the Software Engineering Body of Knowledge*. Version 3.0. [S.l.], 2014.

SPIN. *Static Source Code Analysis Tools for C*. 2016. [Online; acessado 23 Abril de 2016]. Disponível em: <http://www.spinroot.com/static>.

STOL, K.-J.; FITZGERALD, B. A holistic overview of software engineering research strategies. In: *3rd International Workshop on Conducting Empirical Studies in Industry*. [S.l.: s.n.], 2015. p. 8.

TERCEIRO, A. et al. Analizo: an extensible multi-language source code analysis and visualization toolkit. p. 6, 2010.

VITEK, J.; KALIBERA, T. Repeatability, reproducibility, and rigor in systems research. In: ACM. *Proceedings of the ninth ACM international conference on Embedded software*. [S.l.], 2011. p. 33–38.

WHEELER, D. A. *Static analysis tools for security*. 2015. [Online; acessado 23 de Abril de 2016]. Disponível em: <http://www.dwheeler.com/essays/static-analysis-tools.html>.

WIKIPEDIA. *List of tools for static code analysis*. 2016. [Online; acessado 23 Abril de 2016]. Disponível em: https://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis.