Caracterização da qualidade interna de ferrramentas de análise estática de código fonte

Joenio Marques da Costa
Universidade Federal da Bahia (UFBA)
joenio@colivre.coop.br

28 de abril de 2016

1 Introdução

(à fazer)

1.1 Contribuições esperadas

(à fazer)

2 Fundamentação teórica

Qualidade de software é assunto presente em diversos estudos em engenharia de software, ela diz respeito à quão bem um software é projetado e quão bem o software está em conformidade com este design, embora existam várias definições (KITCHENHAM; PFLEEGER, 1996) (MCCONNELL, 2004) (ISO, 2012) (WIKIBOOKS, 2013) (STAINES, 2015) podemos concluir que existem duas dimensões básicas para medir qualidade de software, estas dimensões estão relacionadas a características de qualidade interna e externa de um software.

2.1 Qualidade interna

Segundo McConnell (2004), qualidade interna são aquelas características que preocupam o desenvolvedor, como por exemplo: manutenabilidade, flexibilidade, portabilidade, reusabilidade, readability, testabilidade, compreensão. São questões relacionadas ao código-fonte e em como o software foi construído, como design, boas práticas, etc.

2.2 Qualidade externa

Qualidade externa são aquelas características que afetam o usuário, como por exemplo: correctiness, usability, eficiência, reliability, integridade, adaptabilidade, acurácia, robustez. Estas características impactam extritamente no uso do software e não em como o software foi construído.

Estes características, sejam internas ou externas, segundo a ISO 25010 (???) podem ser divididas em subcaracterísticas, usabilidade por exemplo é dividida em: understandability, learnability, operability, attractiveness, compliance.

Esta característica está relacionada a facilidade com que usuários aprendem e utilizam um sistema, ela passa por questões como facilidade de instalação, aprendizado, e uso. Wikibooks (2013) enumera algumas questões à respeito da usabilidade úteis para mensurar tal característica:

1. A interface de usuário é intuitica (auto-explicativa/auto-documentada)?
2. É fácil de executar uma operação simples?
3. É possível executar operações complexas?
4. O software dá mensagens de erro compreensíveis?
5. Os elementos se comportam como esperado?
6. O software é bem documentado?
7. A interface do usuário é responsiva ou muito lenta?

2.3 Qualidade interna x Qualidade externa

Sabe-se que, em algum nível, as caractarísticas de qualidade interna afetam as características de qualidade externa (MCCONNELL, 2004), softwares que não possuem boa manutenabilidade por exemplo afetam a habilidade de correção de defeitos, que por sua vez afetam as características de exatidão (correctness) e confiabilidade (reliability).

2.4 Métricas de código-fonte

Uma métrica, segundo definição da ISO/IEC 25010 (2001), é a composição de procedimentos para a definição de escalas e métodos para medidas (MEIRELLES, 2013).

Métricas são classificadas quanto aos critérios utilizados para determiná-las, quanto ao método de obtenção (MEIRELLES, 2013). Entre as classificações possíveis temos aquelas consideradas objetivas que tratam de características do código-fonte, muitos trabalhos tentam correlacionar métricas de software com qualidade de software. (Subramanyam e Krishnan, 2003) Briand et al. (2000) (Zuse, 1990).

Em suma, há uma variedade de métricas baseadas análise estática do código-fonte que permitem a avaliação de produtos de software (Gousios et al., 2007).

3 Metodologia

Neste capítulo será apresentada a metodologia utilizada no estudo como meio de validar as seguintes hipóteses:

H1: *Existem publicações sobre ferramentas de análise estática com disponibilidade de código-fonte*

H2: *Existem ferramentas de análise estática disponíveis livremente na indústria com disponibilidade de código-fonte*

H3: *Existem valores de referência para métricas de código-fonte para ferramentas de análise estática*

H4: *Ferramentas da indústria possuem melhores valores de métricas de código-fonte*

H5: *É possível relacionar a característica de qualidade externa usabilidade à valores de métricas de qualidade interna*

As seções à seguir descrevem as atividades de cada etapa da metodologia.

3.1 Planejamento do estudo

3.1.1 Seleção das métricas

Com base no trabalho realizado por Meirelles (2013) onde associou-se qualidade de software à qualidade de código através de métricas de código-fonte como indicador para o sucesso de projetos de software livre, foram utilizadas as métricas abaixo neste estudo, e utilizaremos aqui as mesmas:

Métricas de tamanho:

LOC (Lines of Code), AMLOC (Average Method LOC), Total Number of Modules or Classes

Indicadores estruturais:

NOA (Number of Attributes), NOM (Number of Methods), NPA (Number of Public Attributes), NPM (Number of Public Methods), ANPM (Average Number of Parameters per Method), DIT (Depth of Inheritance Tree), NOC (Number of Children), RFC (Response For a Class), ACCM (Average Cyclomatic Complexity per Method)

Métricas de acoplamento:

ACC (Afferent Connections per Class), CBO (Coupling Between Objects), COF (Coupling Factor)

Métricas de coesão:

LCOM (Lack of Cohesion in Methods), SC (Structural Complexity)

3.1.2 Seleção das fontes de ferramentas de análise estática

Para ser possível validar as hipóteses aqui levantadas é necessário realizar uma busca por ferramentas de análise estática desenvolvidas no contexto da academia e da indústria, para isso, será feito um planejamento detalhado para realizar a seleção de ferramentas em cada um destes contextos.

Academia No contexto acadêmica a busca por ferramentas será feita através de artigos publicados em conferências que tenham histórico de publicação sobre ferramentas de análise estática de código fonte. Estes artigos serão analisados e aqueles com publicação de ferramenta de análise estática serão selecionados.

Indústria Na indústria a busca por ferramentas será feita a partir de referências encontradas na internet, algumas organizações mantém listas de ferramentas para análise de código-fonte, a Wikipedia também mantém uma lista de ferramentas, estas referências serão utilizadas como ponto de partida e cada ferramenta será analisada a fim de validar se são da indústria ou surgiram em contexto acadêmico.

Uma vez que as ferramentas tenham sido selecionadas inicia-se a extração de seus atributos de qualidade interna.

3.1.3 Seleção da ferramenta de análise estática de código-fonte

Para realizar a caracterização das ferramentas através dos seus atributos de qualidade interna é necessário uma ferramenta capaz de analisar estaticamente o código-fonte destas ferramentas e extrair atributos relacionados à sua qualidade interna. Para isto utilizaremos o Analizo(KON, 2010). Falta Justificar! Quais vantagens? Referencias?

3.2 Coleta de dados

A partir das fontes selecionadas na etapa anterior serão realizadas duas atividades para identificar e mapear as ferramentas de análise estática com código-fonte disponível, uma atividade relacionada ao levantamento de ferramentas da academia, outra atividade relacionada ao levantamento de ferramentas da indústria.

3.2.1 Ferramentas da academia

A seleção de ferramentas será relizada através de uma revisão estruturada dos artigos selecionados a partir das seguintes conferências:

- ASE - Automated Software Engineering[1]
- CSMR[2] - Conference on Software Maintenance and Reengineering[3]
- SCAM - Source Code Analysis and Manipulation Working Conference[4]

Chamamos de revisão estruturada um processo disciplinado para seleção de artigos a partir de critérios bem definidos de forma que seja possível a reprodução do estudo por parte de pesquisadores interessados. Alguns resultados preliminares podem ser consultados na Tabela 1 da Seção 4.1.

3.2.2 Ferramentas da indústria

A seleção de ferramentas da indústria será feita a partir de uma busca manual em fontes encontradas na Internet sobre ferramentas de análise estática.

O projeto SAMATE[5] - *Software Assurance Metrics and Tool Evaluation* disponível em NIST (2016) mantém uma lista de ferramentas de análise estática mantida, mais sobre o projeto SAMATE pode ser encontrado em Ribeiro (2015).

O software Spin mantém em seu site uma lista de ferramentas comerciais e de pesquisa para análise estática de código-fonte para C em Spin (2016).

O Instituto de Engenharia de Software do CERT mantém uma lista de ferramentas de análise estática em CERT (2016).

O software Flawfinder oferece em seu site um link com referências para inúmeras ferramentas livres, proprietárias, gratuitas mas não-livres de ferramentas de análise estática e outros tipos de análise em Wheeler (2015).

Uma outra fonte contendo uma relação expressiva de ferramentas é mantida na Wikipedia em Wikipedia (2016).

[1] http://ase-conferences.org

[2] A conferência CSMR tornou-se SANER - Software Analysis, Evolution, and Reengineering a partir da edição 2015.

[3] http://ansymore.uantwerpen.be/csmr-wcre

[4] http://www.ieee-scam.org

[5] http://samate.nist.gov

Tabela 1: Total de artigos analisados por edições do SCAM

Edição	Total de artigos	Artigos com ferramenta
SCAM 2001	23	-
SCAM 2002	18	-
SCAM 2003	21	-
SCAM 2004	17	-
SCAM 2005	19	-
SCAM 2006	22	2
SCAM 2007	23	1
SCAM 2008	29	-
SCAM 2009	20	-
SCAM 2010	21	1
SCAM 2011	21	1
SCAM 2012	22	4
SCAM 2013	24	-
SCAM 2014	35	1
SCAM 2015	?? (pendente)	?
Total	315	10

Estas fontes serão pesquisadas manualmente em busca de ferramentas de análise estática que tenham sido desenvolvidas no contexto da indústria, algums resultados preliminares podem ser encontrados nas Tabelas 3 e 2.

3.3 Caracterização dos artigos

Caracterização dos papers analisados na revisão estruturada e caracterização teórica do ecosistema das ferramentas da academia.

3.4 Caracterização das ferramentas

Será realizada uma caracterização prática das ferramentas, tanto acadêmica quando da indústria, através da análise e extração de métricas de código-fonte das mesmas.

3.5 Exemplo de uso

Por fim, os valores de métricas de referência encontradas serão utilizadas como guia para refatorar a ferramenta Analizo.

(à fazer)

4 Conclusão

4.1 Resultados preliminares

A Tabela 1 apresenta um resumo do número de artigos em cada edição do SCAM e quantos artigos trazem publicação de ferramenta de análise estática com código fonte disponível.

As Tabelas 3 e 2 apresentam ferramentas do NIST após avaliação inicial sobre disponibilidade do código-fonte. Das 54 ferramentas apenas 19 tinham código fonte disponível.

Assim, temos um total de 19 ferramentas da indústria com código-fonte disponível e ??? da academia com código fonte disponível, é preciso avaliar em qual linguagem de programação foi escrita cada ferramentas pois só iremos analisar aquelas em C, C++ o Java que são suportadas pelo Analizo.

Ferramenta utilizada para isto foi a sloccount, uma ferramenta livre para contagem de linhas de código fonte, dá estatística de em qual linguagem é escrita em porcentagem. Abaixo destaco a linguagem de programação que tem maior porção em porcentagem.

Após analise ficamos com um total de 25 ferramentas, 15 da indústria e 10 da academia.

Segue a lista de todos os projetos e qual a fonte (industria ou academia):

4.2 Cronograma

(à fazer)

Referências

CERT. *Secure Coding Tools*. 2016. [Online; acessado 23 Abril de 2016]. Disponível em: ⟨http://www.cert.org/secure-coding/tools/index.cfm⟩.

ISO, I. Iso/iec 25022:2012. *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of quality in use*, 2012.

KITCHENHAM, B.; PFLEEGER, S. L. Software quality: The elusive target. p. 12–21, 1996. Disponível em: ⟨http://dblp.org/db/journals/software/software13.html\#KitchenhamP96⟩.

KON, A. T. J. C. J. M. P. M. L. R. R. L. A. C. C. F. Analizo: an extensible multi-language source code analysis and visualization toolkit. p. 6, 2010.

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⟩.

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

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

STAINES, A. S. Some aspects of software quality assurance for small projects. *International Journal of Scientific and Engineering Research*, v. 6, n. 5, p. 441–445, 2015.

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⟩.

WIKIBOOKS. *Introduction to Software Engineering*. [S.l.]: Wikibooks, 2013. v. 12.

Tabela 2: Lista de ferramentas do SAMATE - NIST com código fonte não disponível

Ferramenta	Avaliacao
ABASH	código não disponível
ApexSec Security Console	código não disponível
Astrée	código não disponível
bugScout	código não disponível
C/C++test®	código não disponível
dotTEST™	código não disponível
Jtest®	código não disponível
HP Code Advisor (cadvise)	código não disponível
Checkmarx CxSAST	código não disponível
CodeCenter	código não disponível
CodePeer	código não disponível
CodeSecure	site offline
CodeSonar	código não disponível
Coverity SAVE™	código não disponível
Csur	código não disponível
DoubleCheck	código não disponível
Fluid	código não disponível
Goanna Studio and Goanna Central	código não disponível
HP QAInspect	código não disponível
Insight	código não disponível
ObjectCenter	código não disponível
Parfait	código não disponível
PLSQLScanner 2008	código não disponível
PHP-Sat	link para código offline
PolySpace	código não disponível
PREfix and PREfast	código não disponivel
QA-C, QA-C++, QA-J	código não disponível
Qualitychecker	código não disponível
Rational AppScan Source Edition	código não disponível
Resource Standard Metrics (RSM)	código não disponível
SCA	código não disponível
SPARK tool set	código não disponível
TBmisra®, TBsecure®	código não disponível
PVS-Studio	código não disponível
xg++	código não disponível

Tabela 3: Lista de ferramentas do SAMATE - NIST com código fonte disponível

Ferramenta	Avaliacao
BOON	código disponível
Clang Static Analyzer	código disponível
Closure Compiler	código disponível
Cppcheck	código disponível
CQual	código disponível
FindBugs	código disponível
FindSecurityBugs	código disponível
Flawfinder	código disponível
Jlint	código disponível
LAPSE	código disponível
Pixy	código disponível
PMD	código disponível
pylint	codigo disponivel
RATS (Rough Auditing Tool for Security)	código disponível
Smatch	código disponível
Splint	código disponível
UNO	código disponível
Yasca	código disponível
WAP	código disponível

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⟩.

Tabela 4: Lista com total de ferramentas a ser analisadas

Ferramenta	Linguagem	Fonte
BOON	ansic	industria
CQual	ansic	industria
RATS	ansic	industria
Smatch	ansic	industria
Splint	ansic	industria
UNO	ansic	industria
Clang Static Analyzer	cpp	industria
Cppcheck	cpp	industria
Jlint	cpp	industria
WAP	java	industria
Closure Compiler	java	industria
FindBugs	java	industria
FindSecurityBugs	java	industria
Pixy	java	industria
PMD	java	industria
Indus	java	academia
TACLE	java	academia
JastAdd	java	academia
WALA	java	academia
error-prone	java	academia
AccessAnalysis	java	academia
Bakar Alir	java/ada/python	academia
InputTracer	ansic	academia
srcML	cpp/cs (cs = C# ?)	academia
Source Meter	java	academia