

Caracterização de Softwares Científicos no Contexto de Engenharia de Software Experimental

Joenio Marques da Costa

Universidade Federal da Bahia (UFBA)

joenio@colivre.coop.br

15 de outubro de 2015

1 Resumo do projeto

Ciência Aberta é um movimento que tem por objetivo tornar a pesquisa científica, seus dados e sua disseminação acessíveis à todos os interessados, dentre as várias iniciativas deste movimento, destaca-se a preocupação com a reprodutibilidade dos resultados de pesquisas de forma independente e aberta, visto que a maioria dos componentes necessários para a reprodução dos resultados de uma pesquisa – por exemplo, códigos fonte e dados – não são publicados.

Em diversas linhas de pesquisa da Computação e, em especial, em Engenharia de Software, é bastante comum que novos sistemas de software sejam desenvolvidos, tais sistemas costumam ser utilizados como meio para atingir os resultados da pesquisa ou, em alguns casos, são o próprio fim do estudo realizado. Neste trabalho, tais sistemas ou ferramentas de software são nosso objeto de pesquisa e serão chamados de *software científico*.

Seja como meio ou como fim, tais *softwares científicos* são produtos de software e, tal como propõe a Engenharia de Software Experimental, precisam ser avaliadas com uso de métodos científicos adequados. Neste contexto, é de fundamental importância que os softwares utilizados ou desenvolvidos durante estudos estejam disponíveis e em funcionamento.

Neste trabalho de mestrado, será realizada uma revisão sistemática de literatura, a partir de artigos da área de Engenharia de Software que tratam de publicação de "softwares científicos", visando coletar e caracterizar alguns atributos de tais softwares para prover recomendações iniciais para o seu desenvolvimento e adoção em ampla escala em estudos empíricos em Engenharia de Software.

2 Objetivos

O objetivo principal deste trabalho é melhorar a compreensão sobre as ferramentas de software para análise de código fonte desenvolvidos durante pesquisas em Engenharia de Software – aqui denominados "software científico" – por meio de sua caracterização no contexto de estudos empíricos em Engenharia de Software.

São objetivos específicos deste trabalho:

- Realizar uma revisão sistemática, com base em artigos publicados nas principais conferências e periódicos da área de Engenharia de Software, sobre "software científico";

- Caracterizar os "softwares científicos" coletados em termos de atributos relacionados ao seu desenvolvimento;
- Caracterizar os "software científicos" coletados em termos de atributos relacionados à sua distribuição;
- Caracterizar os "software científicos" coletados em termos de atributos relacionados ao seu uso em estudos empíricos, sejam primários ou replicações;
- Sintetizar informações sobre "softwares científicos" coletados;
- Prover um conjunto inicial de recomendações para facilitar a adoção de "softwares científicos" em estudos empíricos em Engenharia de Software.

3 Fundamentação teórica

3.1 Ciência Aberta

Ciência Aberta é um movimento que tem por objetivo tornar a pesquisa científica, seus dados e sua disseminação acessíveis à todos os interessados, sejam amadores ou profissionais (WIKIPEDIA, 2015). Sua principal motivação está em possibilitar a reprodução dos resultados de pesquisas e em garantir transparência das metodologias utilizadas, isto aumenta o impacto social das pesquisas e gera economia de tempo e dinheiro para os pesquisadores e para as instituições (RIN/NESTA, 2010).

Este movimento é guiado por princípios básicos de transparência, acessibilidade e reusabilidade universais, disseminadas via ferramentas online, ele é dividido em quatro grandes áreas: (1) Open Access, (2) Open Data, (3) Open Source e (4) Open Reproducible Research. Dentre elas destaca-se a Open Reproducible Research por preocupar-se com a reprodutibilidade dos resultados de pesquisas de forma independente (STODDEN, 2009) e aberta, no entanto, esta área tem recebido ainda pouca atenção da comunidade de pesquisa (PONTIKA et al., 2015) (GRAND et al., 2010b) apesar do aumento geral do interesse pelas práticas da Ciência Aberta (GRAND et al., 2010a).

Enquanto pesquisadores publicam artigos descrevendo e divulgando seus resultados, é raro que façam o mesmo com toda a produção gerada durante a pesquisa. A maioria dos componentes necessários para a reprodução dos resultados de uma pesquisa – por exemplo, códigos fonte e dados – usualmente permanecem não publicados. Este é um problema sério já que um dos fundamentos da ciência é que novas descobertas sejam reproduzidas antes de serem consideradas parte da base de conhecimento (STODDEN, 2009).

Neste sentido, Prlić e Procter (2012) dão dicas para o desenvolvimento aberto de software científico e citam que disponibilizar o código criado durante pesquisas não apenas aumenta o impacto como também se torna essencial para outros reproduzirem os resultados encontrados. Eles citam ainda que manutenabilidade e disponibilidade do software após a publicação é o maior problema enfrentado pelos pesquisadores que desenvolvem tais softwares, e é aí que a participação no desenvolvimento aberto desde o início pode trazer maior benefício.

Dentro deste contexto, e considerando que pesquisas em engenharia de software produzem bastante softwares científicos, surge a preocupação de avaliar tais softwares em termos de sua manutenabilidade e disponibilidade a partir de métodos científicos adequados.

3.2 Engenharia de Software

Sistemas de software são utilizados em praticamente todas as áreas do conhecimento humano e têm exercido um papel essencial em nossa sociedade (MAFRA; TRAVASSOS, 2006). A dependência crescente de serviços

oferecidos por tais sistemas evidencia a necessidade de produzir software de qualidade, contornando os desafios relacionados a funcionalidades incompletas ou incorretas, custos acima do esperado ou prazos não cumpridos.

Diante destes desafios, surge a Engenharia de Software, uma disciplina centrada no desenvolvimento de sistemas de software (WESSLÉN, 2012) através de uma abordagem sistemática, disciplinada, e quantificável para o desenvolvimento, operação e manutenção (SOCIETY, 2014).

Nas últimas décadas, o foco em estudos empíricos na área de Engenharia de Software tem crescido significativamente (STOL; FITZGERALD, 2015), resultando no uso crescente de métodos como surveys, estudos de caso, experimentos e revisões sistemáticas de literatura. Através destes estudos empíricos, pesquisadores transformam a Engenharia de Software em uma disciplina mais científica e controlável – a Engenharia de Software Experimental – provendo meios para avaliar e validar métodos, técnicas, linguagens e ferramentas.

O crescimento no número de pesquisas e publicações em Engenharia de Software Experimental desperta a atenção para a necessidade de verificar a validade dos estudos empíricos realizados – um ponto central em qualquer pesquisa científica. A validade de um estudo empírico deve ser averiguada com o intuito de aumentar o nível de confiança em seus resultados, replicação costuma ser citado como um importante meio para atingir tal objetivo (ALMQVIST, 2006).

Um dos primeiros artigos discutindo replicação de experimentos em Engenharia de Software foi publicado por Basili et al. (MÄNTYLÄ; LASSENIUS; VANHANEN, 2010) e sugere replicação não apenas como uma escolha, mas como um possível "próximo passo" a ser tomado após o experimento original ser concluído. Apesar do conceito replicação de estudos empíricos em Engenharia de Software estar usualmente associado à experimentação, argumenta-se que ele deve ser estendido para incluir ao menos estudos de caso e surveys (BASILI; SELBY; HUTCHENS, 1986).

Em diversas linhas de pesquisa da Computação e, em especial, em Engenharia de Software, é bastante comum que novos sistemas de software sejam desenvolvidos, tais sistemas costumam ser utilizados como meio para atingir os resultados da pesquisa ou, em alguns casos, são o próprio fim do estudo realizado. Neste trabalho, tais ferramentas de software são nosso objeto de pesquisa e serão chamados de "software científico" – Portillo-Rodríguez et al. (2012) utiliza o termo "research tool" para designar este mesmo tipo de software.

Softwares científicos são produtos de software e, em geral, precisam ser avaliados com uso de métodos científicos adequados, e, é de fundamental importância que estejam disponíveis e em funcionamento (KON et al., 2011).

3.2.1 Análise de Código Fonte

O rápido crescimento no número de softwares nas últimas décadas leva a uma crescente demanda por mecanismos e ferramentas de apoio à compreensão de, desenvolvedores necessitam entender em profundidade a implementação de um determinado software antes de realizar atividades de correção ou refatoração de forma eficiente (KIRKOV; AGRE, 2010).

Isto é evidenciado ao perceber que a complexidade dos softwares vem crescendo à cada dia (KIRKOV; AGRE, 2010), tornando assim, extremamente útil a existência de ferramentas de análise automática de código-fonte, tais ferramentas auxiliam os desenvolvedores e engenheiros à compreender a implementação de um determinado software de forma profunda e abrangente.

Essas ferramentas geram modelos de alto nível representando entidades, relacionamentos, métricas, características ou outra informação qualquer extraída diretamente do código-fonte, e permitem que sejam geradas visualizações em diversos níveis de abstração representando o código-fonte.

3.3 Revisão Sistemática e Meta-análise

Muitos pesquisadores argumentam que, para se obter progressos em uma determinada área do conhecimento, os resultados de vários experimentos e outros estudos empíricos – surveys e estudos de caso, por exemplo – devem ser combinados. Quando um conjunto de estudos empíricos é coletado sobre um tópico, a síntese ou agregação entra em cena.

Quando os procedimentos da meta-análise, ou seja, síntese baseada em métodos estatísticos (ALMQVIST, 2006), não são aplicáveis, a síntese descritiva deve ser utilizada. Esta inclui visualização, tabulação e estatística descritiva dos dados. Quanto mais ampla é a questão guiando a revisão de literatura, mais métodos qualitativos são necessários para sua síntese. Cruzes e Dybå (2011) apresentam uma visão geral de métodos qualitativos de síntese, entre eles a síntese temática.

Adicionalmente, com o aumento na adoção de estudos empíricos em Engenharia de Software, surge a necessidade de agregar evidências de múltiplos estudos relacionados, de modo a obter respostas a questões impossíveis de serem respondidas com os estudos individuais. A coleta e síntese de evidências empíricas podem ser realizadas com rigor científico, por meio de Revisão Sistemática da Literatura (KITCHENHAM; CHARTERS, 2007). A Revisão Sistemática da Literatura é um meio de avaliar e interpretar pesquisas relevantes – estudos primários – sobre uma dada questão em particular, tópico, área, ou fenômeno de interesse.

4 Metodologia

Primeiramente será feita uma revisão sobre estudos secundários – em especial, revisões sistemáticas – relacionados ao uso e desenvolvimento de ”softwares científicos”.

Em seguida, será realizada uma revisão sistemática de literatura, com base nas recomendações encontradas em (KITCHENHAM; CHARTERS, 2007), a partir de artigos da área de Engenharia de Software que tratam de publicação de ”softwares científicos” para análise de código fonte, visando caracterizar seus atributos a partir das seguintes perguntas:

- Quais são os ”softwares científicos” associados a pesquisas em engenharia de software?
- Como os ”softwares científicos” publicados nestes estudos são licenciados e distribuídos?
- Como os ”softwares científicos” são desenvolvidos, em termos de métodos e processos recomendados pela engenharia de software?
- Há colaboração entre pesquisadores de universidades distintas no desenvolvimento dos ”softwares científicos” publicados?
- Os ”softwares científicos” são avaliados internamente e externamente? Se sim, como são avaliados?

A partir das informações coletadas na revisão sistemática será feita agregação e síntese dos dados, utilizando métodos quantitativos e qualitativos, com objetivo de identificar temas recorrentes e problemas comuns, além de elaborar conclusão a respeito dos diversos estudos analisados na revisão sistemática.

Após a caracterização realizada, espera-se propor um conjunto preliminar de recomendações para desenvolvimento e adoção de ”softwares científicos” para a comunidade acadêmica de Engenharia de Software.

5 Resultados esperados

- Caracterização dos atributos das ferramentas de software desenvolvidas durante pesquisas em engenharia de software, chamadas aqui de *"softwares científicos"*;
- Síntese dos resultados e lições a respeito de temas recorrentes e problemas comuns no desenvolvimento e publicação de *"softwares científicos"*;
- Conjunto preliminar de recomendações para desenvolvimento e adoção de *"softwares científicos"* para a comunidade acadêmica de Engenharia de Software;
- Artigos científicos publicados;
- Dissertação de mestrado.

6 Atividades e metas

6.1 Meta 1: Realizar revisão sistemática

Atividades:

- Pesquisar estudos secundários sobre o tema;
- Identificar fontes de dados: bibliotecas digitais; anais em conferências sobre ferramentas;
- Levantar e selecionar dados/papers sobre ferramentas de software
- Iniciar e documentar análise dos papers selecionados
- Agregar e sintetizar informações encontradas

6.2 Meta 2: Divulgar resultados

Atividades:

- Apresentar qualificação de mestrado
- Elaborar artigo científico sobre caracterização de ferramentas de pesquisa publicadas no Brasil
- Elaborar artigo científico com lições a respeito de desenvolvimento e publicação de ferramentas de pesquisa
- Elaborar artigo científico com recomendações para desenvolvimento e adoção de ferramentas de pesquisa
- Elaborar dissertação de mestrado
- Apresentar dissertação de mestrado

Referências

- ALMQVIST, J. P. F. Replication of controlled experiments in empirical software engineering - a survey. *Department of Computer Science, Faculty of Science, Lund University*, 2006.
- BASIL, V. R.; SELBY, R. W.; HUTCHENS, D. H. Experimentation in software engineering. *Software Engineering, IEEE Transactions on*, v. 7, p. 733–743, 1986.
- CRUZES, D. S.; DYBÅ, T. Recommended steps for thematic synthesis in software engineering. In: *ESEM*. [s.n.], 2011. p. 275–284. Disponível em: <http://dblp.org/db/conf/esem/esem2011.html\\#CruzesD11>.
- GRAND, A. et al. Muddying the waters or clearing the stream? open science as a communication medium. 2010.
- GRAND, A. et al. On open science and public engagement with engineering. *European Association for Studies in Science and Technology*, p. 1–4, 2010.
- KIRKOV, R.; AGRE, G. Source code analysis - an overview. *Cybernetics and Information Technologies*, v. 10, n. 2, p. 60–77, 2010.
- KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. *EBSE Technical Report*, n. Version 2.3, p. 65, 2007.
- 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>.
- MAFRA, S. N.; TRAVASSOS, G. H. Estudos primários e secundários apoiando a busca por evidência em engenharia de software. 2006.
- MÄNTYLÄ, M. V.; LASSENIUS, C.; VANHANEN, J. Rethinking replication in software engineering: Can we see the forest for the trees? In: *ICSE workshop RESER*. [S.l.: s.n.], 2010.
- PONTIKA, N. et al. Fostering open science to research using a taxonomy and an elearning portal. 2015.
- 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>.
- PRLIĆ, A.; PROCTER, J. B. Ten simple rules for the open development of scientific software. *PLoS Computational Biology*, vol. 8, issue 12, p. e1002802, v. 8, p. 2802, dec 2012.
- RIN/NESTA. *Open to All? Case studies of openness in research*. [S.l.], 2010. Disponível em: <http://www.rin.ac.uk/our-work/data-management-and-curation/open-science-case-studies>.
- SOCIETY, I. C. *Guide to the Software Engineering Body of Knowledge*. Version 3.0. [S.l.], 2014.
- STODDEN, V. Enabling reproducible research: Open licensing for scientific innovation. *International Journal of Communications Law and Policy, Forthcoming*, v. 13, p. 1–25, 2009.
- 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.

WESSLÉN, C. W. P. R. M. H. M. C. O. B. R. A. *Experimentation in Software Engineering*. [S.l.]: Springer Science & Business Media, 2012.

WIKIPEDIA. *Open Science*. 2015. Open Science. [Online; acessado 13 Outubro de 2015]. Disponível em: http://en.wikipedia.org/wiki/Open_science.