



INVESTIGANDO O IMPACTO DE COMPETÊNCIAS DO DESENVOLVEDOR NA CONSTRUÇÃO DE SISTEMAS COM QUALIDADE ESTRUTURAL

Luiz F. M. Regis¹ (IC), Helber W. S. Moreira¹ (IC), *Roberto F. de Oliveira¹ (PQ),
Eduardo M. Fernandes² (PQ), Danyllo W. Albuquerque³ (PQ), Willian N. Oizum⁴ (PQ)

roberto.oliveira@ueg.br

¹Universidade Estadual de Goiás – Câmpus Posse – GO – Brasil

²Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio – Brasil

³Unipê – Câmpus João Pessoa- PB – Brasil

⁴Instituto Federal do Paraná – Câmpus Paranavaí – IFPR – Brasil

Resumo: A longevidade de um sistema de *software* depende da sua qualidade estrutural. Portanto, é desejado que um desenvolvedor de sistema tenha adquirido certas competências de forma a auxiliá-lo na construção de sistemas com qualidade estrutural aceitável. Dentre essas competências, está o domínio e o uso de abordagens e ferramentas de apoio à identificação de problemas de qualidade estrutural, tais como anomalias que dificultam mudanças e correções de defeitos em um sistema. Observamos várias abordagens e ferramentas propostas pela literatura, mas poucas são conhecidas e usadas na prática industrial. Nosso projeto de pesquisa visa elicitá-las principais competências e avaliar o seu impacto na construção de sistemas com qualidade estrutural aceitável. Nosso intuito é apoiar uma formação acadêmica sólida do desenvolvedor em relação à qualidade estrutural de sistemas. Nós exemplificamos tais competência através do uso de ferramentas de apoio à identificação de anomalias afetando sistemas de *software* durante a sua construção.

Palavras-chave: Desenvolvimento de Sistemas. Competências. Qualidade Estrutural. Anomalias.

Introdução

A sociedade contemporânea requer que os sistemas de *software* sejam capazes de acompanhar as constantes necessidades da população (LAVALLÉE e ROBILLARD, 2015). Isso requer que o desenvolvedor de sistemas tenha as competências necessárias para modificar os sistemas sem que isso leve à degradação da qualidade estrutural destes (OLIVEIRA *et al.*, 2016). De fato, a longevidade de um sistema depende da sua qualidade estrutural (MACIA *et al.*, 2012). Uma forma de garantir tal qualidade estrutural é detectar e remover anomalias que afetam a estrutura do sistema (FOWLER, 1999).

Há evidência de que desenvolvedores carentes de certas competências são propensos a introduzir anomalias ao construir sistemas assim que aprendem a

programar, especialmente em seus primeiros empregos (ESTÁCIO *et al.*, 2015). Competências podem ser: *teóricas*, ou seja, atreladas ao conhecimento sobre programação, suas linguagens e formas de abstração; ou *práticas*, isto é, atreladas ao domínio e ao uso de ferramentas e abordagens de apoio ao desenvolvimento. Logo, certas competências precisam ser estimuladas durante a formação acadêmica do desenvolvedor em cursos técnicos, de graduação e de pós-graduação.

Nós propusemos o projeto de pesquisa “Uma investigação sobre o impacto de competências na qualidade estrutural do *software*”, promovido pelo Departamento de Informática da Universidade Estadual de Goiás (UEG), *Câmpus Posse*. Objetivamos investigar e avaliar as competências necessárias ao desenvolvedor de sistemas para garantir a construção de sistemas com qualidade estrutural aceitável. Nosso intuito é apoiar a formação acadêmica de desenvolvedores em cursos de Sistemas de Informação e afins. Neste artigo, introduzimos o projeto e discutimos os resultados já obtidos a respeito de uma competência: a detecção de anomalias.

Material e Métodos

A Figura 1 ilustra nosso método de pesquisa para investigar as competências necessárias ao desenvolvedor para construir sistemas com qualidade estrutural adequada. O método possui as três seguintes etapas. *Revisão da Literatura*: visa sumarizar os últimos 10 anos de literatura sobre uma competência. *Exposições*: visam conscientizar a comunidade quanto ao estímulo de competências na formação acadêmica de desenvolvedores. *Workshop*: visa, junto a docentes e alunos de iniciação científica, discutir estratégias para estimular competências em cursos de Sistemas de Informação e afins na formação acadêmica de desenvolvedores.

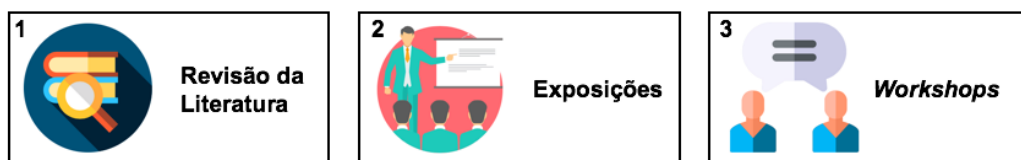


Figura 1 – Método de pesquisa para investigação de competências do desenvolvedor

Através do nosso método, investigamos esta competência: o conhecimento e o uso de abordagens e ferramentas de apoio à detecção de anomalias. Usamos os seguintes materiais. *Revisão da literatura*: utilizamos a base de artigos Google Scholar e planilhas eletrônicas. *Exposições*: apresentamos banners, em 2017, na I

Semana de Iniciação Científica da UEG, *Câmpus Posse* (GO), e na IV Semana de Tecnologia da Informação do Instituto Federal do Paraná (IFPR), *Câmpus Paranavaí* (PR). *Workshop*: fomentamos discussões acaloradas sobre o assunto no I Workshop do Núcleo de Pesquisas em Engenharia de Software da UEG, *Câmpus Posse*.

Salienta-se que, para alcançar o objetivo final deste projeto de pesquisa, será necessário realizar outras três etapas: (1) entrevistar coordenadores e docentes de disciplinas de Engenharia e Desenvolvimento de Software, a fim de identificar as suas percepções quanto à bibliografia e à metodologia de ensino adotada; (2) conduzir experimentos com discentes de Sistemas de Informação e afins, a fim de investigar o impacto do uso das ferramentas e abordagens da literatura na formação acadêmica do desenvolvedor; (3) orientar os docentes das disciplinas supracitadas na melhoria da formação de seus alunos e futuros desenvolvedores.

Resultados e Discussão

Nesta seção, apresentamos os resultados obtidos em âmbito teórico (Etapa 1 da Figura 1) e prático (Etapas 2 e 3). Através da Etapa 1, obtivemos vários resultados dos quais destaca-se o seguinte. A Figura 2 apresenta o percentual de abordagens adotadas para detectar anomalias. Nós identificamos seis categorias de abordagens: *ferramenta* (programa semi-automático que detecta anomalias); *técnica* (pseudo-código que pode ser implementado; *método* (passo a passo técnica); *metodologia* (processo realizado pelo desenvolvedor); *métrica* (valor numérico); e *outras abordagens*, tais como estudos envolvendo aprendizado de máquina (FONTANA et al., 2015) e linguagens de apoio à especificação de anomalias (MOHA et al., 2008). **As duas categorias mais abundantemente exploradas pela literatura são ferramentas (38% dos artigos coletados) e técnicas (15%).**

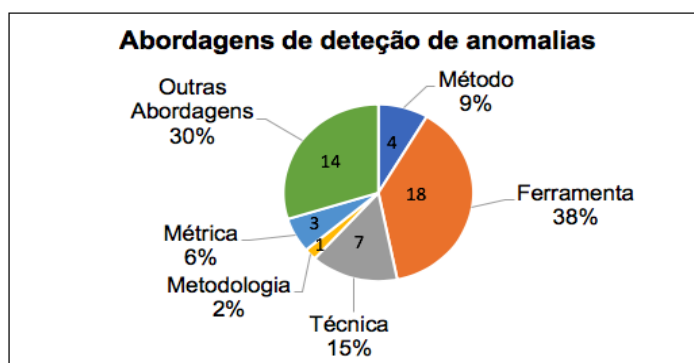


Figura 2 – Tipos de abordagens de detecção de anomalias de código



Nossa experiência prática na Etapa 2 reforçou a necessidade prática do nosso projeto de pesquisa. Em exposições, a comunidade de Sistemas de Informação e áreas afins **mostrou-se aberta ao ensino de detecção e eliminação de anomalias em disciplinas de programação**. A comunidade também **revelou interesse em estimular competências** quanto ao uso de abordagens e ferramentas para detectar anomalias, alinhando a formação do desenvolvedor às necessidades reais do mercado de trabalho quanto à qualidade estrutural de sistemas de *software*.

Por fim, nossa experiência prática através da Etapa 3 revelou observações interessantes a respeito da necessidade de se revisitar a formação acadêmica de desenvolvedores. De fato, por meio de *workshop*, nós **discutimos com os docentes de vários cursos sobre a necessidade de se revisar os currículos de base para o ensino de Engenharia e Desenvolvimento de Software**, a fim de concretizar o estímulo a certas competências de maior impacto na qualidade estrutural de programas. Assim, os docentes serão capazes de, a médio e longo prazo, foram profissionais cada vez mais capacitados a construir sistemas de *software* com qualidade estrutural adequada aos anseios da indústria.

Considerações Finais

Esse projeto de pesquisa vem promovendo uma reflexão sobre a necessidade de se incutir, no ensino da programação de instituições como a UEG/Câmpus Posse, o IFPR/Câmpus Paranavaí e a Unipê-João Pessoa, conteúdo que estimule competências necessárias à construção de sistemas com qualidade estrutural adequada. Dentre nossas contribuições, ressaltamos a provocação de docentes em confrontar o conteúdo de suas aulas com as necessidades de mercado de trabalho. Essa provocação tem conscientizado os docentes a educar para o uso de abordagens e ferramentas propostas pela literatura, as quais são muitas vezes desconhecidas ou pouco exploradas pela indústria.

Nossa hipótese de pesquisa é que, ao tornar o ensino mais completo e alinhado às necessidades do mercado, haja uma consequente elevação na qualidade estrutural dos sistemas de *software* desenvolvidos por eles após a sua formação acadêmica. Porém, ainda há muito trabalho a ser feito. Em especial, planejamos outras etapas metodológicas tais como: (1) entrevistas com docentes;



(2) experimentos controlados; e (3) orientações práticas para os docentes a fim de aperfeiçoar a formação do desenvolvedor.

Agradecimentos

Agradecemos a todos os que, direta ou indiretamente, contribuíram para que fosse possível a realização desse projeto de pesquisa. Também agradecemos à Universidade Estadual de Goiás, *Câmpus Posse*, que proporcionou a base necessária ao desenvolvimento desse projeto.

Referências

ESTÁCIO, B.; OLIVEIRA, R.; MARCZAK, S.; KALINOWSKI, M.; GARCIA, A.; PRIKLADNICK, R.; LUCENA, C. Evaluating collaborative practices in acquiring programming skills: findings of a controlled experiment. In: **29th Brazilian Symposium on Software Engineering**. Belo Horizonte, MG: IEEE, 2015. p. 150-159.

FONTANA, A. F.; FERME, V.; ZANONI, M. Filtering code smells detection results. In: **Proceedings of the 37th International Conference on Software Engineering - Volume 2 (ICSE '15)**, v. 2. Piscataway, NJ: IEEE Press, 2015. p. 803-804

FOWLER, M.; BECK, K. **Refactoring**: improving the design of existing code. Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1999.

LAVALLÉE, M.; ROBILLARD, P. Why good developers write bad code: an observational case study of the impacts of organizational factors on software quality. In: **37th Int. Conference on Software Engineering (ICSE)**. Florence, Italy: IEEE, 2015. p. 677-687.

MACIA, I.; ARCOVERDE, R.; GARCIA, A.; CHAVEZ, C.; STAA, A. On the relevance of code anomalies for identifying architecture degradation symptoms. In: **16th European Conference on Software Maintenance and Reengineering**. Szeged, Hungary: IEEE, 2012. p. 277-286.

MOHA, N.; GUÉHÉNEUC, YG.; LE MEUR, AF.; DUCHIEN, L. A domain analysis to specify design defects and generate detection algorithms. In: **Fundamental Approaches to Software Engineering**. Berlin, Heidelberg: Springer, 2008. p. 276-291

OLIVEIRA, R.; ESTÁCIO, B.; GARCIA, A.; MARCZACK, S.; PRIKLADNICKI, R.; KALINOWSKI, M.; LUCENA, C. Identifying code smells with collaborative practices: a controlled experiment. In: **X Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS)**. Maringá, PR: IEEE, 2016. p. 61-70.