

Elevando a Qualidade Estrutural do Software a Partir da Melhoria das Habilidades e Aptidões de Desenvolvedores de Sistemas de *Software*

Paulo Brandão¹, Roberto Oliveira¹, José Júnior¹
Danyllo Albuquerque² e Cíntia Souza¹

¹Universidade Estadual de Goiás – UEG – *Câmpus* Posse-GO

²Faculdade UniNassau - Campus Palmeira – Campina Grande –PB
paulovitorbrandao7@gmail.com, dwa@mauriciodenassau.edu.br
{roberto.olivera, jose.junior, cintia.souza}@ueg.br

Sistemas de *software* estão integrados a diversos aspectos da vida humana, desde atividades críticas como controle financeiro e de tráfego, até entretenimento digital. Rotineiramente, esses sistemas sofrem mudanças visando alterar funcionalidades existentes (manutenção) ou inserir novas funcionalidades (evolução). Tais mudanças são requeridas para que um sistema atenda às necessidades dos seus usuários, prolongando a sua vida útil e evitando a sua obsolescência. Porém, atividades ineficazes de manutenção e evolução de sistemas podem levar à degradação do *software* [Oliveira et al., 2016]. Estudos prévios indicam que a diminuição da qualidade estrutural do software está intimamente ligada à sua degradação [Macia et al., 2012] e, ademais, o tempo de vida do software depende diretamente das ações de combate a esta degradação [Gamma et al., 1994]. Neste sentido, torna-se imperativa a descoberta de meios para indicar precocemente problemas associados à qualidade estrutural do *software*.

A qualidade estrutural do *software* pode ser mantida por meio da detecção e da remoção de anomalias de código [Fowler, 1999; Macia et al., 2012], as quais são problemas estruturais que dificultam a manutenção e a evolução do sistema de *software* [Fowler, 1999]. Quando tais anomalias são identificadas precocemente, torna-se facilitada sua remoção e a manutenção da qualidade estrutural do *software* [Albuquerque et al., 2014]. Portanto, as anomalias devem ser identificadas e removidas para que a degradação do *software* não exija um esforço demasiado para sua manutenção [Macia et al., 2012; Oliveira et al., 2016].

Estudos indicam que desenvolvedores introduzem anomalias de código em dois momentos: assim que iniciam o aprendizado de programação; e quando participam dos primeiros projetos de desenvolvimento [Estácio et al., 2015; Hermans e Aivaloglou, 2016]. Tais trabalhos indicam que a formação acadêmica inadequada é um dos fatores que conduz os desenvolvedores a introduzirem tais anomalias. Assim, o presente trabalho investiga e avalia as principais habilidades a serem desenvolvidas durante a formação acadêmica dos desenvolvedores visando à construção de sistemas de *software* com qualidade estrutural.

Para alcançar esse objetivo, as seguintes etapas são fundamentais: (1) Levantamento bibliográfico visando identificar os desafios associados ao desenvolvimento de *software* com qualidade estrutural, (2) Coleta de dados junto a especialistas da área de programação; (3) Condução de experimentos propondo novos métodos e técnicas na formação dos desenvolvedores; (4) Avaliação experimental do impacto da mudança na formação dos desenvolvedores em relação à qualidade estrutural do *software*. Ressalta-se que as atividades (1) e (2) encontram-se em fase de

conclusão. Como resultados preliminares, foram encontrados 77 trabalhos sobre problemas de qualidade estrutural de *software*. A grande maioria dos trabalhos encontrados são estudos exploratórios (35%) ou estudos de caso (26%). Isso demonstra uma carência por estudos empíricos para avaliar aspectos relacionados à qualidade estrutural em sistemas de *software*.

Em suma, espera-se alcançar as seguintes contribuições: reflexão e revisão dos currículos de base para o ensino de programação; proposição de metodologias visando o desenvolvimento de *software* fácil de manter e evoluir; capacitação dos discentes e docentes dos cursos de programação das instituições de ensino superior participantes; e divulgação dos resultados obtidos por meio de artigos.

Referências

- Albuquerque, D.; Garcia, A.; Oliveira, R. and Oizumi, W. [2014]. “Detecção Interativa de Anomalias de Código: Um Estudo Experimental”. 11th Workshop on Software Modularity. Maceió, Brasil: [s.n.], p. 53-66.
- Estácio, B.; Oliveira, R.; Marczak, S.; Kalinowski, M.; Garcia, A.; Prikladnicki, R. and Lucena, C. [2015]. “Evaluating Collaborative Practices in Acquiring Programming Skills: Findings of a Controlled Experiment”. 29th Brazilian Symposium on Software Engineering. Belo Horizonte, Brasil: IEEE. p. 150-159.
- Fowler, M. [1999]. “Refactoring: Improving the Design of Existing Code”. Boston: Addison-Wesley Publishing.
- Gamma, E.; Helm, R. and Johnson, R. [1994]. “Design Patterns: Elements of Reusable Object-Oriented Software”. Mountain View: Addison-Wesley Professional, 1994. 416 p.
- Hermans, F.; Aivaloglou, E. [2016]. “Do code smells hamper novice programming? A controlled experiment on Scratch programs”. 24th International Conference on Program Comprehension (ICPC). Austin, USA: IEEE. p. 1-10.
- Macia, I.; Arcoverde, R.; Garcia, A.; Chavez, C. and von Staa, A. [2012]. “On the Relevance of Code Anomalies for Identifying Architecture Degradation Symptoms”. In: 16th European Conference on Software Maintenance and Reengineering. Szeged, Hungary : IEEE. p. 277-286.
- Oliveira, R.; Estácio, B.; Garcia, A.; Marczak, S.; Prikladnicki, R.; Kalinowski, M. and Lucena, C. [2016] “Identifying Code Smells with Collaborative Practices: A Controlled Experiment”. In: X Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS). Maringá, Brasil: IEEE. p. 61-70.