

Um Ambiente Gamificado para o Ensino de Teste de Mutação

Dário da Silva Darienço
Universidade Estadual do Norte do Paraná
Bandeirantes, Paraná, Brasil
d.darienco@gmail.com

Pedro Henrique Dias Valle
Universidade Federal de Juiz de Fora
Juiz de Fora, Minas Gerais, Brasil
pedrovalle@ice.ufjf.br

O teste de software é um tópico muito importante da Engenharia de Software, no qual visa verificar o quão “confiável” é o software no quesito de garantir que ele funcione como o esperado [8]. Portanto, devem ser conduzidas verificações no software para identificar erros e então corrigi-los, reduzindo gastos com manutenção e garantindo aumento na qualidade do produto [1]. Em resumo, o teste de software tem por objetivo executar programas ou modelos com entradas específicas com intuito de levá-los a falhar e posteriormente, por meio da depuração, eliminar os defeitos que originaram as falhas [2].

Quando a atividade de teste é realizada de forma criteriosa e embasada tecnicamente, o que se tem é certa “confiança” de que se comporta corretamente para grande parte do seu domínio de entrada, agregando qualidade ao produto. Nesse sentido, foram propostas técnicas e diversos critérios pertencentes a cada técnica para que se possa se testar o software como um todo.

Nessa perspectiva, é de grande importância que os profissionais que atuam na área de teste de software estejam qualificados para contribuir no processo de desenvolvimento de um software, garantindo sua respectiva qualidade [5]. Para isso, os profissionais podem utilizar diferentes técnicas (e.g. funcional, estrutural e baseados em defeitos) para derivar diferentes casos de teste com o objetivo de revelar a presença de defeitos no sistema em teste [2].

Em particular, o teste de mutação ou análise de mutantes é um critério de teste mais conhecido da técnica de Teste Baseado em Defeitos. Esse critério adiciona possíveis defeitos ao programa uniformemente. Para isso, utilizam-se defeitos que são frequentemente cometidos por desenvolvedores [11]. Na técnica de Teste Baseado em Defeitos, o programa testado é alterado várias vezes, criando um conjunto de programas alternativos (mutantes) [2].

Nesse contexto, percebe-se um crescimento da indústria de software por profissionais mais qualificados para realizar testes, visto que o teste de software proporciona diversos benefícios, além de garantir a qualidade de software. Porém, diferentes estudos [4, 7, 10] destacam que há um *deficit* de profissionais qualificados na área de teste de software. As evidências mostram que muitos estudantes sentem-se desmotivados em aprender os conteúdos relacionados ao teste de software, criando um *deficit* na formação de profissionais qualificados para atuar na área de teste de software [11].

De acordo com Songhao Jia, Cai Yang [6] e Valle [11], o motivo da desmotivação pode estar associada com: (i) a incoerência entre

os conteúdos práticos e teóricos, causando menor interesse entre os estudantes; (ii) os conteúdos ensinados em sala de aula não serem os mesmos exigidos na indústria; (iii) os estudantes possuírem dificuldades em entender o processo e as etapas de testes; (iv) a falta de experiência dos estudantes no desenvolvimento de software.

Além disso, a atividade de teste de software, muitas vezes, é vista como uma atividade destrutiva pois seu objetivo é revelar a presença de defeitos nos códigos dos programadores [4, 12]. Isso pode contribuir para a desvalorização dos profissionais em relação a outros cargos dentro de uma empresa de software. Além disso, ensinar teste de software não é algo trivial pois tal atividade está relacionada com a habilidade de investigar das pessoas em verificar defeitos nos códigos dos sistemas.

Diante desse cenário, percebe-se que há um problema na formação de profissionais qualificados para atuar na área de teste de software. Portanto, se faz necessário resolver/amenizar esse problema para que a indústria de software possa continuar produzindo softwares com qualidade e também para que os estudantes e profissionais consigam se qualificar para suprir a necessidade da indústria de software e também para buscar a melhoria da área de teste de software.

Uma alternativa para isso seria a utilização da gamificação como recurso de apoio ao processo de ensino e aprendizagem dos alunos em relação aos conteúdos de teste de software. A gamificação fornece diversos benefícios devido sua dinâmica de apresentação dos conteúdos [9], nos quais os elementos de jogos podem ser utilizados para motivar os estudantes a aprenderem os conteúdos propostos em sala de aula [3]. Além disso, a gamificação incentiva a competição e também a colaboração entre os alunos, o que pode promover seu interesse em aprender os tópicos propostos em sala de aula [9].

Portanto, este trabalho tem como objetivo **fornecer mecanismos para apoiar o ensino de teste de software nos cursos de graduação, especificamente o teste de mutação**. A gamificação será utilizada como recurso de apoio durante o processo de ensino e aprendizagem desses conteúdos. Em particular, pretende-se desenvolver um ambiente de ensino gamificado para apoiar o ensino desses conteúdos. Para isso, será necessário: (i) identificar os conteúdos de teste de mutação que mais carecem de material de apoio; (ii) identificar quais elementos de gamificação serão considerados para apoiar o ensino de teste de mutação; (iii) definir as tecnologias (e.g., linguagens de programação/*game engines*) que serão utilizadas para apoiar o desenvolvimento do ambiente gamificado; (iv) desenvolver um ambiente de ensino gamificado para apoiar o ensino e treinamento de teste de mutação; (v) avaliar a eficiência do ambiente proposto para ajudar no ensino de teste de mutação.

REFERÊNCIAS

- [1] Draylson Micael de Souza, José Carlos Maldonado, and Ellen Francine Barbosa. 2012. Aspectos de Desenvolvimento e Evolução de um Ambiente de Apoio ao

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'22, Abril 24–29, 2022, Feira de Santana, Bahia, Brasil (On-line)

© 2022 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

- Ensino de Programação e Teste de Software. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 23.
- [2] M.E. Delamaro, J.C. Maldonado, and M. Jino. 2016. *Introdução ao teste de software* (2 edição ed.). Elsevier, Rio de Janeiro.
- [3] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From game design elements to gamefulness: defining "gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. 9–15.
- [4] G. Fraser. 2017. Gamification of Software Testing. In *2017 IEEE/ACM 12th International Workshop on Automation of Software Testing (AST)*. IEEE, Buenos Aires, Argentina, 2–7.
- [5] Gordon Fraser and Andrea Arcuri. 2012. Whole Test Suite Generation. *IEEE Transactions on Software Engineering* 99 (01 2012). <https://doi.org/10.1109/TSE.2012.14>
- [6] Songhao Jia and Cai Yang. 2013. Teaching software testing based on cdio. *World Transactions on Engineering and Technology Education* 11, 4 (2013), 476–479.
- [7] J. M. Rojas and G. Fraser. 2016. Code Defenders: A Mutation Testing Game. In *2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, Chicago, 1–6.
- [8] Ian Sommerville. 2011. Engenharia de software, 9a. *São Palo, SP, Brasil* (2011), 63.
- [9] Armando M Toda, Ana CT Klock, Wilk Oliveira, Paula T Palomino, Luiz Rodrigues, Lei Shi, Ig Bittencourt, Isabela Gasparini, Seiji Isotani, and Alexandra I Cristea. 2019. Analysing gamification elements in educational environments using an existing Gamification taxonomy. *Smart Learning Environments* 6, 1 (2019), 1–14.
- [10] Pedro Valle, Ellen Francine Barbosa, and José Maldonado. 2015. Um mapeamento sistemático sobre ensino de teste de software. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 26. 71.
- [11] Pedro Henrique Dias Valle. 2016. *Jogos educacionais: uma contribuição para o ensino de teste de software*. Ph.D. Dissertation. Universidade de São Paulo.
- [12] Pedro Henrique Dias Valle, Ellen Francine Barbosa, and Jose Carlos Maldonado. 2015. CS curricula of the most relevant universities in Brazil and abroad: Perspective of software testing education. In *2015 International Symposium on Computers in Education (SIE)*. IEEE, 62–68.