

UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE SISTEMAS DE INFORMAÇÃO

CÁSSIO LINDEN ALBERT

PESQUISA SOBRE GAMIFICAÇÃO NOS TESTES DE SOFTWARE

São Leopoldo
2019

Cássio Linden Albert

PESQUISA SOBRE GAMIFICAÇÃO NOS TESTES DE SOFTWARE

Artigo apresentado como trabalho de grau
A da disciplina de Elaboração de Projetos
pelo Curso de Sistemas de Informação da
Universidade do Vale do Rio dos Sinos –
UNISINOS

Professor: Prof. Cristiano Bonato Both

São Leopoldo

2019

PESQUISA SOBRE GAMIFICAÇÃO NOS TESTES DE SOFTWARE

Cássio Linden Albert*

Cristiano Bonato Both**

Resumo:

Palavras-chave:

1 INTRODUÇÃO

2 FUNDAMENTAÇÃO TEÓRICA

Basicamente ensinar alguns termos que podem ser novos para o leitor. Isso serve para ele entender o que será abordado na análise;

* Aluno do curso de Sistemas de Informação. Email: cassiola@edu.unisinos.br

** Professor da Unisinos. Email: cboth@unisinos.br

2.1 Gamificação

2.2 Teste de software

2.2.1 Técnicas de teste

2.2.2 Níveis de teste

2.2.3 Teste de mutantes

2.3 Machine Learning

2.4 Metodologia ágil

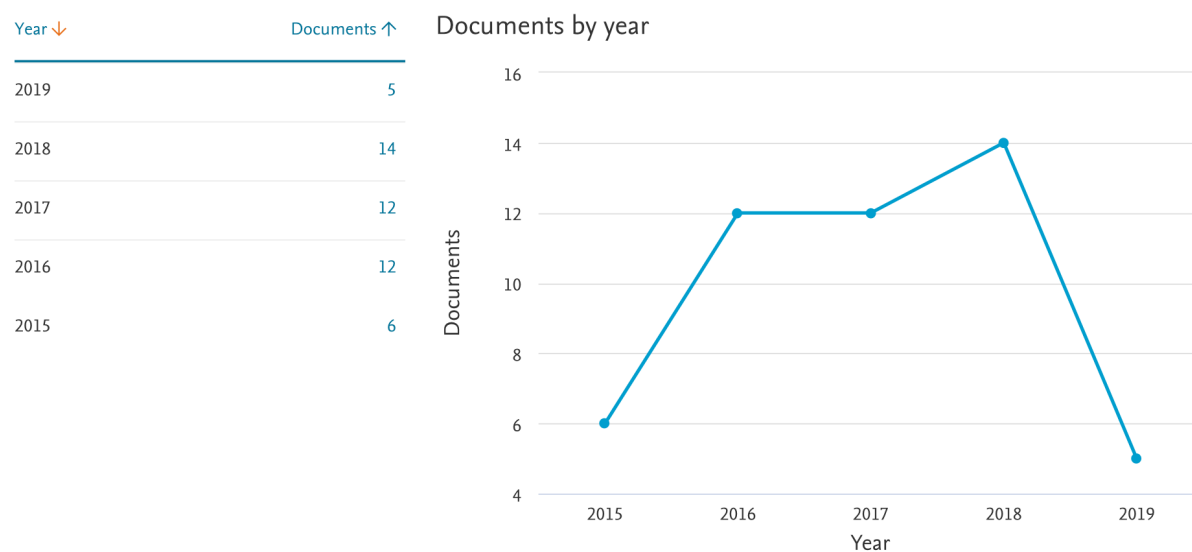
3 METODOLOGIA

Para começar esse trabalho, foi feita uma pesquisa sistemática, cujo objetivo era confirmar que a gamificação nos testes de software é um tema relevante. Para isso, utilizou-se a ferramenta Scopus, que é um grande repositório de citações e resumos revisados por especialistas das mais diversas áreas, oferecendo várias estatísticas e gráficos. Para os artigos utilizados nesse trabalho, os seguintes termos foram pesquisados:

(TITLE-ABS-KEY (gamification) AND TITLE-ABS-KEY (software AND testing)) AND (LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015)) AND (LIMIT-TO (SUBJAREA , "COMP"))

Essa *query* de pesquisa filtra artigos publicados entre 2015 e 2019 que envolvem termos estritamente relacionados a gamificação e testes de software, limitando-se à área da computação. Em 9 de abril de 2019, a *query* retornou 49 resultados. Abaixo há um gráfico mostrando o número de artigos publicados desde 2015. Considerando que a pesquisa foi feita no início de 2019, essa curva tem uma leve queda, mas a tendência de crescimento nesta área é grande.

Figura 1 – Curva de artigos publicados de 2015 a 2019



Para selecionar os artigos que serão abordados nessa pesquisa, foram verificados os títulos de cada artigo retornado pela *query* pesquisada no Scopus que tivesse relação direta com o tópico desejado, passando pela leitura do resumo de cada um e encerrando com a verificação da disponibilidade dos mesmos na base de dados da UNISINOS. Esta análise resultou na tabela abaixo:

Authors	Title	Year
Fraser G.	Gamification of software testing	2017
De Jesus G.M., Ferrari F.C., De Paula Porto D., Fabbri S.C.P.F.	Gamification in software testing: A characterization study	2018
Parizi R.M.	On the gamification of human-centric traceability tasks in software testing and coding	2016
Elgrably I.S., Oliveira S.R.B.	Gamification and evaluation of the use the agile tests in software quality subjects: The application of case studies	2018
Pedreira O., García F., Brisaboa N., Piattini M.	Gamification in software engineering - A systematic mapping	2015
García F., Pedreira O., Piattini M., Cerdeira-Pena A., Penabad M.	A framework for gamification in software engineering	2017
Houshmand M., Paydar S.	EMVille: A gamification-based approach to address the equivalent mutant problem	2017
Rojas J.M., Fraser G.	Code Defenders: A Mutation Testing Game	2016

Na seção de análise e discussão dos resultados, os artigos filtrados serão abordados e analisados mais a fundo.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

(FRASER, 2017) afirma que desenvolver bons testes de software é difícil e não é algo destacado no ensino da programação, além de não ser uma atividade que os desenvolvedores apreciam. O mesmo também destaca alguns aspectos importantes para bons testes, como a intuição humana, o entendimento de como o software deve funcionar e de qual o contexto do mesmo. Considerando os grandes problemas que podem ser causados por testes feitos de forma inadequada, o autor explica que seu artigo explora uma nova forma de resolver o problema dos testes: a gamificação. Com isso, quem já trabalha na área terá mais chances de se destacar, conhecendo novas ferramentas e técnicas de teste, além de facilitar o ensino nas escolas, onde o assunto é negligenciado quando se trata de programação. O autor também destaca o uso dessas ferramentas em crowdsourcing, onde várias pessoas trabalham numa só finalidade. Ele finaliza falando que a aplicação de gamificação tem potencial de melhorar a qualidade dos testes de software e, consequentemente, do software em geral.

O autor também cita que as pesquisas no meio ainda estão num estágio bastante inicial, porém, já mostrou resultados promissores. Talvez por não ter sido aplicado numa escala maior, ainda não há resultados concretos, porém, o autor poderia ter levantado alguns dados quantitativos relacionados ao experimento citado no quarto tópico do artigo.

(DE JESUS et al., 2018) apresentou um cenário geral de outros artigos que abordam gamificação nos testes de software, incluindo alguns dos artigos abordados aqui. O estudo foi dividido em seis categorias: contexto de aplicação, elementos de gamificação utilizados, objetivos da gamificação, técnicas de testes de software, níveis de teste e fases no processo de software. Considerando o contexto, os estudos eram focados em aplicações educacionais e industriais, sendo vários deles usados nos dois contextos; poucos eram exclusivos. Desses, os autores encontraram formas de aplicá-los em outros contextos. Por exemplo: um estudo que era focado no ambiente educacional sendo utilizado em ambientes industriais, e vice-versa. Sobre elementos de gamificação, os autores afirmam que a tríade PBL (*Points, Badges, Leaderboard*, podendo ser traduzido como *Pontos, Emblemas, Quadro de Líderes*) era a mais utilizada nos estudos, havendo apenas um estudo que não utilizou pontos e outro que não conseguiu aplicar sua metodologia no contexto correto. Entre os objetivos da gamificação, os autores apontam o maior desenvolvimento da criatividade, colaboração, engajamento, motivação e satisfação. Quanto às técnicas de teste, abordou-se a funcional, estrutural e baseados em falhas, sendo a última a mais utilizada nas aplicações de gamificação. Além disso, alguns estudos têm aplicabilidade nas três técnicas citadas. Nos níveis de teste, foram abordados os de unidade (dos quais a maioria são utilizados com outras ferramentas de automação), integração e de sistema. Aqui, os estudos tiveram foco nos testes de unidade. Os outros estudos não mencionaram em quais níveis eles poderiam ser aplicados, porém, os autores acreditam que a aplicabilidade é possível em todos os níveis citados. Quanto às fases no processo de software, os autores afirmam que praticamente todos os estudos podem ser aplicados em todo o processo, com exceção do planejamento e na configuração de dados e ambiente. Como sendo uma área nova na engenharia de software, os autores acreditam ser um tópico a ser monitorado quanto ao seu crescimento, especialmente na área de testes.

(ROJAS; FRASER, 2016) trouxe o desenvolvimento de um jogo chamado *Code*

Defenders, cujo objetivo é desenvolver e ensinar testes de mutantes. Ele funciona da seguinte forma: dois jogadores, sendo um atacante e um defensor, através da técnica de testes de mutantes, devem melhorar tanto seus casos de teste quanto o próprio código-fonte. No jogo, o atacante deve, propositalmente, introduzir uma falha numa classe Java (um mutante), e o defensor deve escrever um caso de teste no JUnit capaz de detectar esta falha (matar o mutante). O jogo funciona por rodadas de ataque e defesa, onde o primeiro a jogar é o atacante. Quando o atacante produz um mutante com sucesso, o defensor terá a chance de produzir uma defesa contra esse mutante. Caso o defensor consiga matar o mutante, ele leva um ponto; caso contrário, o atacante fica com um ponto. Se o mutante for equivalente, o *gameplay* muda: se o defensor suspeita que este mutante é equivalente, o atacante é desafiado a aceitar essa suspeita, perdendo pontos e passando-os para o defensor, ou criar um caso de teste que mata este mutante; neste caso, cada jogador fica com um ponto. Alguns desafios citados pelos autores e que pode diminuir o alcance é que ele deve ter, obrigatoriamente, dois jogadores humanos, e que devem ter experiência com código em Java. É bem interessante a ideia do jogo, e abstrair a parte do código, fazendo metáforas com outras situações, é bastante promissor, e pode trazer ainda mais pessoas para a área, tanto de desenvolvimento quanto para testes.

(HOUSHMAND; PAYDAR, 2017) traz um *framework* para combater os mutantes equivalentes de forma eficiente. Os autores citam que testes de mutantes são muito custosos, e que quando os mesmos são equivalentes, trazem dificuldades de serem detectados pelos atacantes, fazendo o esforço não valer a pena. Alguns outros trabalhos foram citados pelos autores para atacar esse problema, como limitar algumas características do código desenvolvido para facilitar a detecção de mutantes, verificar padrões no código mutante, ou comparar os códigos originais com os mutantes no momento da compilação do código (também chamado de TCE). Concordando que mutantes não podem ser detectados sem interferência humana, os autores propõem um modelo de três camadas, em que primeira usa a técnica de TCE, a segunda usa a gamificação e a terceira pega os resultados das duas camadas e usa a técnica de *machine-learning* para deixar a prática mais automatizada. A camada de gamificação se chamará *EMVille*, em que especialistas entrarão em jornadas para resolver ou mitigar problemas de mutantes equivalentes. Porém, o método de pontuação não ficou

claro, e não foi possível compreender de que formas o usuário irá pontuar, não pontuar ou, até mesmo, perder seus pontos. Além disso, na metáfora proposta para jogadores não muito familiarizados com o assunto, ficou bastante confuso, não parecendo ter relação com o problema que se está disposto a resolver. Porém, os autores afirmam que a proposta, testada num ambiente real, ajudou os profissionais a resolverem problemas de mutantes equivalentes.

(ELGRABLY; OLIVEIRA, 2018) mostra um estudo feito com estudantes de graduação e pós-graduação relacionado ao ensino de testes ágeis nas disciplinas de Qualidade de Software ensinadas nessas modalidades de ensino. Porém, o mesmo não tem fundamentação teórica, o que dificulta o entendimento de quem não faz parte daquela área de estudo.

(PARIZI, 2016) cita que...

(PEDREIRA et al., 2015) traz um panorama geral de gamificação no desenvolvimento de software. Aqui, é afirmado que, junto com as áreas de desenvolvimento e requisitos, a área de testes é bastante adequada para gamificação, visto que elas compartilham características como o nível de dificuldade em relação a outras tarefas do desenvolvimento de software, a necessidade de colaboração e, até mesmo, um alto índice de tédio. Nessas atividades, os autores afirmam que a gamificação ajuda bastante no rendimento, pois impulsiona o trabalho em equipe. Porém, mesmo considerando uma boa atividade para aplicação da gamificação, os autores não encontraram estudos muito completos na área de testes. Sobre elementos de gamificação, assim como foi apontado por (DE JESUS et al., 2018), os mais utilizados foram os pontos, emblemas e quadro de líderes, além de também citar jornadas, como no trabalho de (HOUSHMAND; PAYDAR, 2017). Um aspecto importante apontado aqui é que a gamificação deve ser aplicada com cuidado na engenharia de software: muitos dos trabalhos estudados pelos autores falharam ao serem aplicados, justamente pela maioria deles usarem apenas o sistema de pontos, e ignorar muitos dos mecanismos de gamificação possíveis, levando muito rapidamente ao desuso devido à repetitividade. Um paralelo que pode ser feito é com o cenário cada vez maior de jogos on-line, em que as desenvolvedoras que não incrementam seus jogos e não criam eventos para incentivar seus jogadores, por exemplo, acabam por perder público.

(GARCÍA et al., 2017) mostra...

5 CONCLUSÃO

Abstract:

Keywords:

REFERÊNCIAS

DE JESUS, G. M. et al. Gamification in software testing: a characterization study. In: ACM INTERNATIONAL CONFERENCE PROCEEDING SERIES, 2018. **Anais...** [S.l.: s.n.], 2018. p. 39–48.

ELGRABLY, I. S.; OLIVEIRA, S. R. B. Gamification and evaluation of the use the agile tests in software quality subjects: the application of case studies. In: ENASE 2018 - PROCEEDINGS OF THE 13TH INTERNATIONAL CONFERENCE ON EVALUATION OF NOVEL APPROACHES TO SOFTWARE ENGINEERING, 2018. **Anais...** [S.l.: s.n.], 2018. v. 2018-March, p. 416–423.

FRASER, G. Gamification of software testing. In: IEEE/ACM 12TH INTERNATIONAL WORKSHOP ON AUTOMATION OF SOFTWARE TESTING, AST 2017, 2017., 2017. **Proceedings...** [S.l.: s.n.], 2017. p. 2–7. Cited By :7.

GARCÍA, F. et al. A framework for gamification in software engineering. **Journal of Systems and Software**, [S.l.], v. 132, p. 21–40, 2017. Cited By :19.

HOUSHMAND, M.; PAYDAR, S. Emville: a gamification-based approach to address the equivalent mutant problem. In: INTERNATIONAL CONFERENCE ON COMPUTER AND KNOWLEDGE ENGINEERING, ICCKE 2017, 2017., 2017. **Anais...** [S.l.: s.n.], 2017. v. 2017-January, p. 326–332.

PARIZI, R. M. On the gamification of human-centric traceability tasks in software testing and coding. In: IEEE/ACIS 14TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING RESEARCH, MANAGEMENT AND APPLICATIONS, SERA 2016, 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p. 193–200. Cited By :11.

PEDREIRA, O. et al. Gamification in software engineering - a systematic mapping. **Information and Software Technology**, [S.l.], v. 57, n. 1, p. 157–168, 2015. Cited By :116.

ROJAS, J. M.; FRASER, G. Code defenders: a mutation testing game. In: IEEE INTERNATIONAL CONFERENCE ON SOFTWARE TESTING, VERIFICATION AND VALIDATION WORKSHOPS, ICSTW 2016, 2016., 2016. **Proceedings...** [S.l.: s.n.], 2016. p. 162–167. Cited By :11.