

Test.wiki: Plataforma Crowdsourcing para Testes Funcionais

Camila Munzlinger¹, Tobias De Vargas¹, Mateus Henrique Dal Forno¹

¹Instituto Federal de Educação, Ciência e Tecnologia Farroupilha (IFFar)
Campus Frederico Westphalen

Caixa Postal 169 – 98.400-000 – Frederico Westphalen – RS – Brasil

{camila.2019001356, tobias.2018001709}@aluno.iffar.edu.br,

mateus.dalforno@iffarroupilha.edu.br

***Abstract.** Testing is one of the main steps in software development. Despite this, small development companies find it difficult to carry out it, as they do not have a demand to hire a professional for this purpose. In this sense, initiatives that allow more flexible work demand can benefit such companies. This work describes the development of a platform for the execution of functional tests, which uses crowdsourcing as a strategy for recruiting testers. In addition to on-demand hiring, the platform can also help companies document the tests performed. Currently, the platform is in the acceptance test stage, which is being carried out through a case study.*

***Resumo.** O teste é uma das principais etapas do desenvolvimento de software. Apesar disso, pequenas empresas de desenvolvimento têm dificuldade de realizá-lo, pois não possuem demanda para contratar um profissional para esta finalidade. Neste sentido, iniciativas que permitam a flexibilização de demanda de trabalho podem beneficiar tais empresas. Este trabalho descreve o desenvolvimento de uma plataforma para a execução de testes funcionais, que utiliza o crowdsourcing como estratégia para o recrutamento de testadores. Além de permitir a contratação por demanda, a plataforma também pode auxiliar as empresas a documentar os testes realizados. Atualmente, a plataforma está em estágio de testes de aceitação, sendo realizados através de um estudo de caso.*

1. Introdução

Durante o desenvolvimento de um software, existem várias atividades que são realizadas, para que o resultado deste trabalho possa proporcionar ao cliente uma solução que atenda às suas necessidades. Dentre as atividades, pode-se citar a validação do software, a qual engloba – dentre outras técnicas – os testes do sistema, como o teste funcional e os testes de usabilidade, visando aprimorar a interface do sistema. Na etapa de testes, pode-se revelar erros e omissões na definição dos requisitos do sistema, o que evidencia a importância desta etapa do processo de desenvolvimento de software [Sommerville 2011, Zanatta et al. 2016].

Quando a etapa de testes é aplicada em um sistema da maneira apropriada, o sistema fica menos suscetível a erros, o que passa maior segurança aos clientes e usuários do mesmo, já que os erros foram encontrados pelo testador e corrigidos, antes do usuário utilizar as funções.

Dentre as empresas de pequeno e médio porte que lançam normalmente de 2 a 4 atualizações de um software por mês, torna-se inviável a contratação de um profissional apenas para execução de testes do sistema. Conseqüentemente, esta tarefa fica a cargo dos demais profissionais da empresa, como o suporte técnico ou até mesmo o próprio programador. Por não haver um profissional dedicado a esta função, a realização dos testes geralmente acaba sendo feita às pressas, de maneira informal e exploratória, sem a devida atenção e cuidados necessários para que realmente os testes sejam efetivos.

O uso do crowdsourcing permite a flexibilidade da mão de obra [Howe 2006]. Desta forma é possível, por exemplo, que várias pessoas possam executar o mesmo teste. A sua aplicação pode abranger uma vasta diversidade de dispositivos, já que o software é testado nos equipamentos pessoais de cada testador. O crowdsourcing também é vantajoso no quesito de remuneração proporcional à atividade, visto que a remuneração dos testadores é equivalente ao acordo feito antecipadamente entre o contratador e o testador, estipulando metas que o testador deve atingir para que este receba a quantia estabelecida. Com isso, o crowdsourcing pode se tornar uma alternativa viável para essas empresas.

Neste sentido, este trabalho tem como objetivo geral propor uma alternativa à atual maneira como as empresas de desenvolvimento de software de pequeno porte realizam testes nos softwares desenvolvidos, através da construção de uma plataforma para documentação e suporte à execução de testes funcionais, utilizando o crowdsourcing como estratégia para o recrutamento de testadores.

Para atingir este objetivo, foi necessário o estudo do Crowdsourcing Process for Functional Tests (CPFT), que é um modelo de processo de crowdtesting genérico, proposto por Dal Forno (2016), bem como o estudo de plataformas já existentes, buscando identificar as dificuldades de adoção de uso das mesmas por empresas de desenvolvimento de software de pequeno porte. Na sequência, desenvolveu-se uma plataforma de testes funcionais, que utiliza o crowdsourcing como estratégia de recrutamento da mão de obra de testes (testadores). Por fim, é realizada a validação da plataforma, através da condução de um estudo de caso, com a participação de empresas de desenvolvimento de software de pequeno porte da região do Médio Alto Uruguai Gaúcho e do Oeste Catarinense.

O restante deste artigo está organizado conforme descrito a seguir. A Seção 2 apresenta a fundamentação teórica, onde será abordado o conceito de crowdsourcing e sua aplicação para teste de software. Na Seção 3 são apresentados trabalhos relacionados, realizando um comparativo com sistemas similares existentes no mercado. A Seção 4 traz as etapas metodológicas executadas neste trabalho. Os resultados obtidos são apresentados na Seção 5. Por último, a Seção 6 apresenta as considerações finais, acompanhada na sequência pelas referências bibliográficas.

2. Fundamentação Teórica

A definição da palavra “crowdsourcing” e seu significado surgiu no ano de 2006, a partir de um artigo escrito pelo jornalista norte-americano Jeff Howe. O mesmo conceitua o termo como sendo o ato de selecionar uma tarefa até então realizada internamente em uma empresa, e terceirizá-la a uma rede indefinida de pessoas, mobilizada por meio de um convite aberto [Howe 2006].

Os indivíduos que possuem interesse em contribuir, normalmente estão distribuídos geograficamente, o que mescla habilidades e costumes diferentes, gerando as-

sim ideias inovadoras perante a interação destes indivíduos. O que também pode gerar dificuldades, como a comunicação efetiva quando indivíduos falam diferentes línguas. Mesmo que a ideia não seja totalmente nova, o crowdsourcing ganhou popularidade com o uso da internet (especialmente após o advento da web 2.0) [Peng et al. 2014, Stol and Fitzgerald 2014] e das redes sociais [Slack 2018, Kogler et al. 2018].

Slack (2018) ainda dá como exemplo a empresa de produtos de consumo Procter & Gamble (P&G), a qual mobilizou cientistas amadores para explorarem ideias sobre corantes de detergente cuja cor mudasse quando despejado em uma máquina lava-louças. Ele também apresenta o crowdsourcing aplicado nos órgãos governamentais, quando estes convocam os cidadãos para poder definir prioridades ou cortes em gastos de projetos.

Pinochet (2014) apresenta como exemplo o Foldit, que é um jogo experimental de computador desenvolvido pela Universidade de Washington que reúne voluntários para decifrar a estrutura de uma proteína, diretamente relacionada com o vírus HIV. Foi produzido um modelo 3D, resultado alcançado em apenas 10 dias.

Uma das atividades do processo de desenvolvimento de software que podem ser realizadas com o uso de crowdsourcing é o teste. Kogler et al. (2018) conceituam crowdtesting como o termo que é aplicado no desenvolvimento dos testes de software através do modelo crowdsourcing. Os autores descrevem que, quando uma pessoa inicia a função de testador em uma plataforma de crowdtesting, ela participa de uma etapa classificatória, baseada em reputação, pautada na qualidade e no desenvolvimento de atividades anteriormente realizadas. Dessa maneira, no crowdtesting se aplica o conceito de crowdsourcing, já que os usuários efetuam os testes de maneira remota em seu ambiente. Ao contratar esse tipo de serviço, obtemos pontos positivos como a execução de testes em várias plataformas, dispositivos, versões e idiomas, dado que os testadores utilizam seus próprios dispositivos para executarem os testes.

De acordo com Delamaro et al. (2007), durante o teste funcional ocorre a projeção dos casos de testes de um software. Nele, são fornecidas as entradas e analisadas as saídas, verificando se estas estão conforme o esperado, segundo o ponto de vista do usuário.

O teste funcional de software é a técnica onde verifica-se as funções do sistema, desconsiderando os detalhes internos de implementação [Kolm 2001]. Para que se tenha um bom teste funcional, é necessária uma especificação estruturada e coerente com as solicitações do usuário, pois as funções do software são criadas a partir das especificações. Bartié (2002, p.113) considera que o teste funcional é a “categoria mais prioritária dentre as demais, pois representa a aderência do software em relação às regras de negócio”.

Comparando o teste de software tradicional e o crowdtesting, Mao et al. (2017) defendem que o crowdtesting possui uma vantagem, onde é possível fazer com que, além de testadores profissionais apliquem os testes, os usuários finais também tenham participação nessa etapa do desenvolvimento de software. Após submetidos e concluídos, os testes passam por uma análise. Se esta é aprovada, então o testador adquire uma pontuação, elevando os níveis de reputação. Dessa maneira, crescem as chances de que o testador seja convidado para outros novos projetos. Caso a análise seja reprovada, o testador vai adquirindo uma reputação negativa. Consequentemente, haverá menos convites de novos projetos para testes, o que posteriormente poderá causar a desistência do testador de utilizar a plataforma.

3. Trabalhos relacionados

As subseções a seguir apresentam algumas plataformas crowdsourcing existentes, destacando-se suas características e estruturas de funcionamento.

3.1. Crowdtest

A plataforma Crowdtest¹ foi desenvolvida com o objetivo de poder unir a mão de obra de testadores freelancers com empresas que necessitam de testes para validar aplicações e serviços. Esta plataforma utiliza o crowdsourcing, já que várias pessoas testam a mesma funcionalidade para encontrar falhas. Este modelo contém o benefício da variedade de dispositivos e configurações. Dessa maneira traz rapidez nos resultados retornados pelos indivíduos testadores. A plataforma também oferece um plano gratuito, porém este não permite a definição de testadores, não permite a definição de dispositivos alvo a serem testados e também não permite o registro de evidências em vídeo. Além disso, a versão gratuita não traz atratividade do projeto aos testadores, tendo em vista que estes não serão remunerados com o uso da plataforma.

3.2. uTest

UTest² é uma plataforma que pode ser usada para que sejam feitos testes em praticamente qualquer projeto, envolvendo desde encontrar bugs em sistemas até testes que necessitam ser feitos em ambientes físicos específicos, como restaurantes ou lojas. Os testadores são selecionados através da qualificação que cada um demonstra na plataforma, às vezes o requisito para ser um testador pode ser simplesmente possuir um aparelho Android, em outros casos necessita-se de qualificações mais específicas como falar determinado idioma, ou ter um objeto que cumpra totalmente com as especificações solicitadas. A remuneração dos testadores é feita com base no projeto testado. Ao concluir os testes, a plataforma revisa o trabalho feito e estabelece a remuneração conforme o trabalho julgado correto. Os detalhes sobre os pagamentos de cada projeto podem ser consultados pelos testadores antes de aceitar a solicitação de teste. O principal ponto negativo do uso desta plataforma é que há dificuldades em encontrar testadores habilitados para execução de testes em português, o que inviabiliza sua adoção para pequenas empresas, que produzem predominantemente softwares para uso na região, e sem internacionalização de idioma.

3.3. Plataforma desenvolvida

Empresas desenvolvedoras de software de pequeno porte, por terem poucos funcionários, acabam testando seus sistemas superficialmente ou de forma exploratória, ou sequer executam testes antes de lançar uma atualização. Com isso, acabam deixando brechas para que os clientes possam encontrar erros no sistema e reclamarem a cada atualização de algo que não funciona corretamente.

Nesse contexto, foi desenvolvida a plataforma Test.wiki, focada em empresas locais com dificuldade de manter um indivíduo que execute a função de teste para as aplicações da empresa, como por exemplo funções de módulos desenvolvidos, na qual a empresa pode cadastrar solicitações de testes a serem feitos em seus sistemas. Após

¹Disponível em <https://crowdtest.me/>

²Disponível em <https://www.utest.com/>

isso, os indivíduos interessados em fazer os testes se candidatam para a vaga e, então, a empresa contrata quem ela definir que seria a melhor opção para o teste.

Descrevendo, os diferenciais possuídos pela Test.wiki são: ser uma plataforma brasileira, possuir um custo mais acessível para empresas de desenvolvimento de pequeno porte e ser construída a partir do processo adaptável CPFT [Dal Forno 2016], ou seja, o seu uso tem a possibilidade de ser totalmente autônomo, dessa maneira não é necessária a participação de analistas da plataforma (como é o caso da crowdtest).

Também é importante ressaltar que a plataforma Test.wiki também possui a vantagem de, além de ser uma plataforma para recrutamento, também ser uma ferramenta para a documentação e registro das evidências de teste, uma vez que os planos de teste cadastrados permanecem salvos na plataforma, mesmo após concluídos os testes, podendo ser excluídos apenas pelo analista de teste que os cadastrou.

A Tabela 1 apresenta um comparativo entre as principais características existentes nas plataformas anteriormente descritas.

Tabela 1. Comparativo entre as plataformas

Crítérios	Crowdtest	Utest	Test.wiki
Área para cadastro de casos de teste	X	X	X
Área para cadastro de resultados de teste	X	X	X
Área para registro de bugs	X	X	X
Foco em EPP			X
Baixo custo	X		X
Plataforma brasileira	X		X
Especificação do teste em português	X		X

Conforme demonstrado na Tabela 1, as três plataformas contém uma área para que a empresa possa cadastrar os casos de teste. Todas as plataformas possuem uma área para que após a execução dos testes, os testadores possam retornar com os feedbacks e relatando os bugs, caso eles existam.

A plataforma Crowdtest e a Utest não apresentaram flexibilidade na gestão dos resultados de teste, sendo necessário a intervenção de analistas das respectivas plataformas no retorno dos feedbacks. As mesmas plataformas também tem sua aplicação focada para o uso por médias e grandes empresas, não tendo foco em empresas de pequeno porte (EPP), como é o caso da Test.wiki. Relacionado a contratação de planos, a plataforma Crowdtest e a Test.wiki apresentam um menor custo para utilização, em comparativo com a plataforma Utest, especialmente devido ao fato da moeda padrão da plataforma ser o Dólar americano. Por fim, as plataformas Crowdtest e Test.wiki, se destacam no uso do português como idioma padrão, o que permite que os usuários possam cadastrar os testes e os feedbacks nesta linguagem, pois nas plataformas internacionais são direcionadas ao uso do idioma inglês, tanto para a especificação de testes e resultados de teste, quanto para o software propriamente dito.

4. Metodologia

Os itens a seguir detalham as etapas metodológicas desenvolvidas neste trabalho.

- **Análise da documentação** do CPFT, processo de teste proposto por Dal Forno (2016). Esta etapa permitiu a compreensão e organizado os próximos passos do desenvolvimento deste trabalho, bem como facilitou a estruturação das rotinas do sistema.
- **Levantamento e especificação de requisitos** da plataforma Test.wiki. Os requisitos definidos fundamentam-se na documentação analisada, e com eles foi possível prosseguir com o processo de desenvolvimento da plataforma de maneira mais objetiva e organizada.
- **Desenvolvimento**, envolvendo o projeto do sistema e codificação da plataforma. O projeto do sistema foi realizado utilizando-se modelagem UML. Para a codificação foram utilizadas as linguagens de programação Python e o Framework Django para codificar o backend. Já para o frontend utilizou-se o Framework Angular, juntamente com as linguagens de marcação HTML e CSS. O armazenamento dos dados utilizado é o SGBD PostgreSQL.
- **Validação** da plataforma através da execução de testes funcionais caixa preta. Os casos de teste foram definidos a partir dos requisitos especificados anteriormente.
- **Publicação da plataforma** em um servidor web.
- Condução de **testes de aceitação** na plataforma Test.Wiki, por meio de um estudo de caso (em andamento).
- **Análise dos resultados** obtidos com o teste de aceitação, e realização de modificações na plataforma, caso necessário.

5. Resultados

A seguir, são apresentados os resultados obtidos com este trabalho a partir das etapas metodológicas anteriormente citadas.

5.1. Detalhamento do sistema

O sistema é composto por 3 módulos, sendo:

- I **Administrador:** possui controle amplo da aplicação e é responsável por gerenciar o saldo proveniente da integração com o intermediador de pagamento.
- II **Analista de Testes:** responsável pelo planejamento dos testes crowdsourcing, publicação das tarefas de teste, seleção dos testadores e análise dos resultados obtidos com a execução dos testes.
- III **Testador:** faz parte da multidão e é o responsável por executar as tarefas de teste e registrar o trabalho realizado, recebendo ao final a remuneração pelo seu trabalho, caso esteja satisfatório.

Para a publicação de um plano de teste, o analista precisa escolher quanto deseja pagar por cada caso de teste e o saldo é descontado da sua conta virtual para garantir a segurança do processo, permanecendo retido na plataforma. Para adicionar o saldo, a plataforma conta com integração com o intermediador de pagamento do Mercado Pago.

No cadastro de um plano de testes, o analista deve fornecer informações sobre o sistema, sobre os dispositivos, as etapas a serem seguidas e deve ser definido ao menos um dispositivo no qual os testes devem ser executados, escolhendo o dispositivo e o sistema operacional. Caso necessário, devem ser fornecidos os navegadores e operadoras de celular nos quais os testes devem ser executados. A Figura 1 apresenta a tela de cadastro de plano de teste.

Figura 1. Tela de cadastro de plano de teste.

Relacionado ao plano de testes, o analista de teste deve definir os cenários de teste e os casos de teste. Um plano de teste pode conter vários cenários de teste, e cada cenário de teste pode conter a ele associado vários casos de teste. O cadastro do caso de teste permite que sejam incluídas informações básicas, como o título e a descrição do caso de teste, além de informações mais detalhadas, como por exemplo as pré e pós condições, os passos que devem ser realizados e também o resultado esperado ao ser executado cada passo. Uma captura de tela do cadastro de um caso de teste pode ser visualizada na Figura 2.

Figura 2. Tela de cadastro de caso de teste.

Após publicar o plano de teste, a publicação fica visível em um feed e os testadores podem manifestar interesse na execução de uma determinada tarefa de teste. Cada manifestação indica ao analista de teste a reputação do testador na plataforma, permitindo que este selecione o testador melhor qualificado. A Figura 3 exemplifica uma solicitação de execução de uma tarefa de teste.

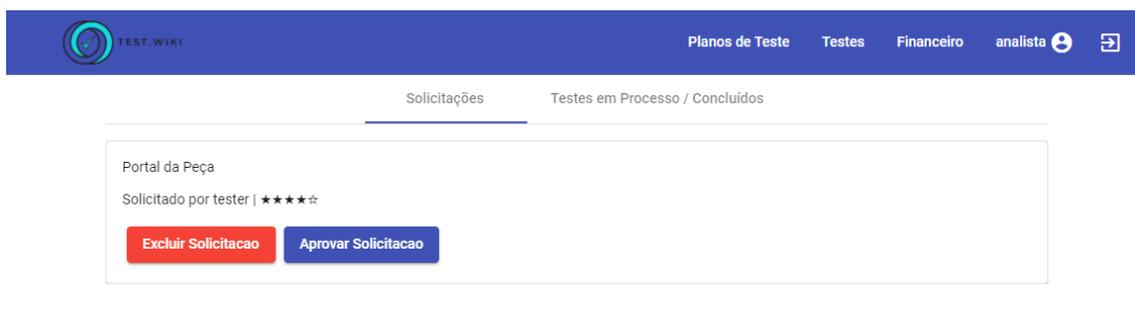


Figura 3. Manifestação de interesse de testador.

Após a aprovação, o testador tem acesso às informações e pode iniciar a execução dos casos de teste, relatando o resultado obtido e registrando as falhas encontradas. A Figura 4 ilustra o registro de uma falha identificada durante a execução do caso de teste. Para cada falha encontrada, devem ser obrigatoriamente detalhados os passos para a reprodução dos erros, também podendo ser anexadas imagens para facilitar o entendimento e visualização.

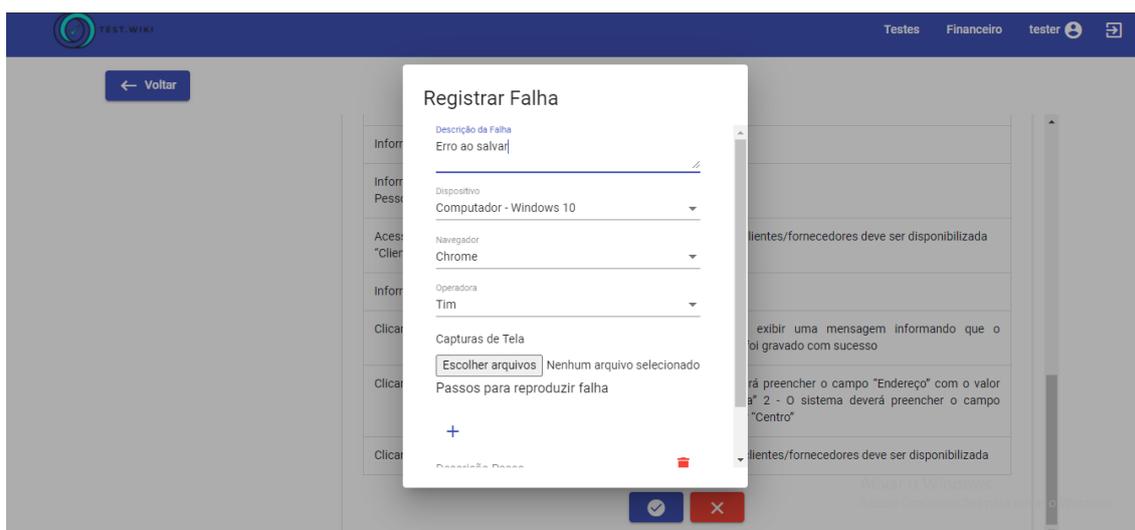


Figura 4. Tela de registro de falha.

Após o analista de testes aprovar a execução da tarefa de testes, é necessário aguardar 30 dias para que a plataforma permita a retirada de saldo para alguma conta que o testador possua no Mercado Pago e o administrador possa fazer a transferência para o testador, conforme solicitado.

5.2. Hospedagem e publicação na web

A plataforma Test.wiki encontra-se hospedada no endereço web <http://inf2.fw.iffarroupilha.edu.br/~camila2019001356/frontend/> e permite o cadastro dos analistas de teste das empresas participantes do estudo de caso, bem como dos testadores, que serão responsáveis pela execução dos casos de teste.

A etapa de execução de testes de aceitação encontra-se em andamento, sendo que a abordagem metodológica escolhida para sua execução é o estudo de caso. O estudo de

caso definido é do tipo integrado, sendo composto por um único caso e duas unidades de análise.

A primeira unidade de análise é uma empresa desenvolvedora de software de pequeno porte, localizada no Oeste Catarinense. A empresa possui uma equipe de 8 funcionários e possui enfoque no desenvolvimento de sistemas ERP que atendem a clientes também de pequeno porte que possuem necessidades, como: manter o cadastro de pessoas e de produtos armazenados e acessados de qualquer lugar, emitir notas fiscais, manifestos, ou até mesmo a criação de um site para e-commerce. Nesta empresa a escrita dos casos de teste já foi iniciada, bem como também foram definidos os valores de cada caso de teste.

A segunda unidade de análise é uma empresa localizada na região do Médio Alto Uruguai Gaúcho. A empresa possui 13 funcionários e atua em 16 estados do país. O ramo principal de desenvolvimento de software são sistemas para centros de desmontes veiculares, sendo credenciada pelo Detran RS. Além disso, a empresa desenvolve sistemas ERP para empresas em geral e é responsável pelo desenvolvimento e manutenção do primeiro marketplace de comércio de peças usadas de desmontes credenciados pelo Detran. Nesta empresa o processo de escrita dos testes também já foi iniciado.

Paralelamente a elaboração dos planos de teste nas unidades de análise, está sendo realizada a divulgação da plataforma em instituições de ensino técnico e superior da região, para que estudantes em um primeiro momento, possam se candidatar como testadores.

6. Considerações Finais

Este trabalho teve como principal objetivo o desenvolvimento de uma plataforma para suporte à execução de testes funcionais, utilizando o crowdsourcing como estratégia para o recrutamento de testadores. Assim, foi proposta uma alternativa à atual maneira como as empresas de desenvolvimento de software de pequeno porte realizam testes nos softwares desenvolvidos.

A execução das etapas metodológicas definidas encontra-se em fase final, já tendo o estudo de caso sendo conduzido nas empresas participantes. A adoção das empresas como participantes do teste de aceitação possibilita que a validação da plataforma aconteça em ambiente real de uso, utilizando softwares comerciais para a execução do processo de teste, recrutando testadores reais, que executarão os casos de testes especificados e irão reportar falhas efetivamente existentes nos softwares.

Ao final, espera-se que o estudo de caso confirme ou refute as seguintes proposições:

- A plataforma permitirá a execução de testes envolvendo uma maior diversidade de dispositivos utilizados para execução dos casos de teste.
- A plataforma possibilitará menores custos, obtidos com a mobilização de testadores eventuais, se comparado a um profissional em tempo integral na empresa.
- A plataforma possibilitará ampliação da qualidade dos sistemas a serem desenvolvidos pelas empresas, isso devido as diferentes perspectivas alcançadas por diferentes testadores e também pela ampliação do uso das práticas de teste.

Além disso, com a conclusão do estudo de caso, poderá ser observado se a plataforma Test.wiki atende adequadamente às empresas participantes, demonstrando a eficácia do modelo de processo CPFT proposto por Dal Forno (2016), assim como a da plataforma desenvolvida.

Após a conclusão do estudo de caso e da análise dos resultados obtidos, pretende-se realizar o registro do software junto ao Instituto Nacional da Propriedade Intelectual (INPI), o que trará maior visibilidade e relevância à plataforma. Também pretende-se expandir a divulgação da plataforma Test.wiki a nível nacional, junto a empresas desenvolvedoras de software e também potenciais testadores, ampliando e conseqüentemente qualificando a base de usuários da plataforma e a qualidade dos testes desenvolvidos através da mesma.

Referências

- Bartié, A. (2002). *Garantia da qualidade de software: adquirindo maturidade organizacional*. Elsevier, Rio de Janeiro, 1 edition.
- Dal Forno, M. H. (2016). *CPFT: Uma Proposta de Processo Adaptável Para Testes Funcionais Utilizando Crowdsourcing*. Dissertação (Mestrado em Computação Aplicada) – Universidade de Passo Fundo.
- Delamaro, M. E., Maldonado, J. C., and Jino, M. (2007). *Introdução ao Teste de Software*. Elsevier, Rio de Janeiro, 1 edition.
- Howe, J. (2006). The rise of crowdsourcing. *Wired magazine*, (14):1–5.
- Kogler, R., Costela, F., and Zanatta, A. (2018). Sistemas de reputação: Uma análise em plataformas de crowdtesting. In *Anais da II Escola Regional de Engenharia de Software*, pages 97–104, Porto Alegre, RS, Brasil. SBC.
- Kolm, E. L. (2001). *Sistema para gerenciamento de testes funcionais de software*. Santa Catarina: Universidade Regional de Blumenau.
- Mao, K., Capra, L., Harman, M., and Jia, Y. (2017). A survey of the use of crowdsourcing in software engineering. *Journal of Systems and Software*, 126:57–84.
- Peng, X., Ali Babar, M., and Ebert, C. (2014). Collaborative Software Development Platforms for Crowdsourcing. *IEEE Software*, 31(2):30–36.
- Pinochet, L. H. C. (2014). *Tecnologia da informação e comunicação*. São Paulo: GEN Atlas.
- Slack, N. (2018). *Administração da produção*. São Paulo: Atlas.
- Sommerville, I. (2011). *Engenharia de Software*. Pearson Prentice Hall, São Paulo, 9 edition.
- Stol, K.-J. and Fitzgerald, B. (2014). Two’s company, three’s a crowd: a case study of crowdsourcing software development. In *Proceedings of the 36th International Conference on Software Engineering - ICSE 2014*, pages 187–198, New York, New York, USA. ACM Press.
- Zanatta, A., Machado, L., Pereira, G., Prikladnicki, R., and Carmel, E. (2016). Software crowdsourcing platforms. *IEEE Software*, 33(06):112–116.