

Faculdade de Tecnologia de São José do Rio Preto
Informática para Negócios

TESTES DE SOFTWARE EM EQUIPES ÁGEIS

Gabriel Fajersztajn, Igor Isnardi Barreto

e-mail:

gabriel.fajersztajn@fatec.sp.gov.br; igorisnardibarreto@gmail.com

Resumo: O artigo verifica o impacto na produtividade de equipes ágeis de desenvolvimento de software ao implementar técnicas de qualidade como teste de software. Para responder a problemática de como os testes de software impactam na produtividade de equipes ágeis de desenvolvimento de software aplicou-se uma Prova de Conceito (POC) em um estudo de caso realizado no mercado de trabalho da tecnologia. Essa pesquisa mensura de forma qualitativa a eficiência que pode ser adquirida ao automatizar tarefas de validação que eram feitas manualmente. Esse estudo foi realizado utilizando uma metodologia baseada em uma revisão bibliográfica em conjunto de uma aplicação prática de uma ferramenta de testes chamada *Cypress.js*, buscando a melhora da qualidade produzida e aumento da produtividade. Os resultados desse estudo demonstram que utilizar técnicas de qualidade no desenvolvimento como TDD (*Test-drive development*) indicam uma economia significativa do tempo de trabalho da equipe, aumento na qualidade do produto final entregue, destacando a importância dos custos que podem ser evitados, do profissional capacitado para executar essas técnicas, da percepção do cliente ao utilizar um produto de qualidade, dos desafios enfrentados pelas empresas para implementar essas técnicas e assim garantir competitividade no mercado exigente da indústria de software.

Palavras-chave: Qualidade de software. Desenvolvimento de software. Equipes ágeis. Testes de software. Produtividade em equipes ágeis.

Abstract: *This article verifies the impact on the productivity of agile software development squads when implementing quality assurance techniques such as software testing. To answer the central question of how can software tests impacts on the productivity of software development agile teams, was applied a Proof of Concept (POC) in a case study practiced out in a technology market job. This research qualitatively measures the efficiency that can be gained by automating validation tasks that were previously done manually. This study case was done using literature review as methodology research together with a practical application of a testing framework called Cypress.js, seeking to improve the quality produced and increase productivity. The results of this study demonstrates that using quality assurance techniques in development such as TDD (Test-driven development) indicates significant savings in the teams working hours, an increase in the quality of the final product delivered, highlighting the importance of costs that can be avoided, the professional qualified to execute these techniques, the customer's perception when using a quality product, the challenges faced by companies to implement these techniques and the way it guarantee competitiveness in the demanding market of the software industry.*

Keywords: *Software quality assurance. Software development. Agile teams. Software Tests. Productivity in agile teams.*

1 Introdução

Nos últimos anos, as práticas e abordagens do setor de desenvolvimento de software experimentaram uma transformação notável. As metodologias ágeis se tornaram cada vez mais comuns, as equipes de desenvolvimento de software estão constantemente buscando formas de melhorar seus procedimentos, assegurando entregas rápidas e de alta qualidade. A utilização de testes automatizados de software é uma ótima técnica para ser utilizada, quando o objetivo é buscar uma melhoria na qualidade do software (BERNARDO; KON, 2008). Os testes automatizados têm um papel essencial para aumentar a eficiência e garantir uma confiança maior no desenvolvimento de software em equipes ágeis.

A tecnologia baseada em software, nos dias atuais, é relevante e em muitos casos imprescindível, com crescente aplicação nos mais variados segmentos. O mercado consumidor tem sido cada vez mais exigente, levando as organizações que atuam no segmento de tecnologia da informação, a entregar softwares de qualidade com valor reduzido e dentro do prazo combinado (DE SOUZA, 2004). Os métodos ágeis proporcionam uma abordagem flexível e adaptável frente a esses desafios atuais, permitindo que as equipes respondam de forma rápida às mudanças de requisitos durante o desenvolvimento. Porém a agilidade não é atingida apenas com práticas rápidas de desenvolvimento, é necessária uma estratégia que garanta a qualidade do software e que acompanhe o ritmo do ciclo de desenvolvimento da equipe. Para Silva (2023, p.6)“Este tema está diretamente associado à qualidade de um produto proposto, visto que agilidade e qualidade são pilares essenciais para uma entrega de sucesso”.

Nesse cenário, os testes automatizados de software surgem como uma ferramenta fundamental no desenvolvimento de software para as equipes ágeis. Testes automatizados são programas ou scripts simples que exercitam funcionalidades do sistema sendo testado e fazem verificações automáticas nos efeitos colaterais obtidos. A grande vantagem desta abordagem, é que todos os casos de teste podem ser facilmente e rapidamente repetidos a qualquer momento e com pouco esforço. Os testes de software permitem ser aplicados inúmeras vezes, simulando de forma idêntica diversos cenários, de acordo com a necessidade desejada, dessa forma garante que etapas importantes do processo não sejam descartadas por conta de falha humana, sendo assim, mais simples de identificar não conformidades ou comportamentos indesejados que possam ocorrer (BERNARDO; KON, 2008).

Diante disso o que nos leva a pergunta principal a ser respondida: Como os testes de software impactam na produtividade de times ágeis no desenvolvimento de software? Para orientar práticas eficientes, aprimorar a eficiência e a qualidade do processo de desenvolvimento, assim garantindo a satisfação do cliente em um mercado altamente competitivo, é fundamental compreender o papel e o impacto dos testes de software dentro do contexto ágil. Nesse artigo iremos abordar os impactos, os desafios encontrados e as melhores práticas na implementação do uso de testes de software em equipes ágeis, visando proporcionar informações valiosas para profissionais e pesquisadores desse segmento de desenvolvimento de software.

2 Justificativa

É de grande importância para uma entrega bem-sucedida de um produto, quando ao desenvolver softwares, aplicar boas práticas de teste para garantir a qualidade do software em todas as fases do ciclo de desenvolvimento em um projeto utilizando metodologias ágeis. Essas práticas envolvem testes em diversas etapas do processo, desde a concepção até a entrega do produto final. A importância está na capacidade de detectar e corrigir bugs antecipadamente, reduzir riscos e garantir que o programa satisfaça os requisitos do usuário de maneira confiável e consistente. A partir do momento que é possível reduzir a frequência com que os problemas aparecem, o tempo que é utilizado para verificar tais problemas é minimizado, dessa forma, o esforço utilizado que se necessita para corrigir esses problemas no código fonte do software durante o desenvolvimento do produto, faz com que se torne mais viável o aumento da qualidade do software desenvolvido (DE MATOS, 2020).

A implementação dessa cultura de qualidade ajuda a identificar falhas em todas as etapas do ciclo de desenvolvimento, em especial nos estágios iniciais, ao identificar essas falhas no começo do projeto, permite que a equipe de desenvolvimento responda rapidamente para resolver os problemas evitando que se tornem problemas mais complexos e custosos no futuro, podendo causar consequências mais graves, pois quanto mais tempo esse defeito demora para ser descoberto, maior é o custo, maior é a complexidade pra resolver, além de carregar uma dívida técnica durante todo os ciclos posteriores do projeto de desenvolvimento. Em muitos cenários é inevitável o acúmulo de dívida técnica, porém, essa dívida pode ser minimizada com testes de software, haja vista que corrigir um defeito na fase inicial do projeto é relativamente simples e barato, pois com algumas correções nas linhas de código é possível resolver o problema. Supondo que esse mesmo defeito fosse encontrado após o lançamento de um software, poderia acarretar em consequências graves, como falhas no sistema, insatisfação do cliente, perda de dados ou até danos à reputação da empresa. Conforme o projeto vai sendo desenvolvido, quanto mais tarde um defeito no software é percebido, mais caro ele se torna para ser corrigido e, maior é o esforço necessário para solucionar esse defeito (SARCHESI; DINIZ, 2021) conforme ilustrado na Figura 1.

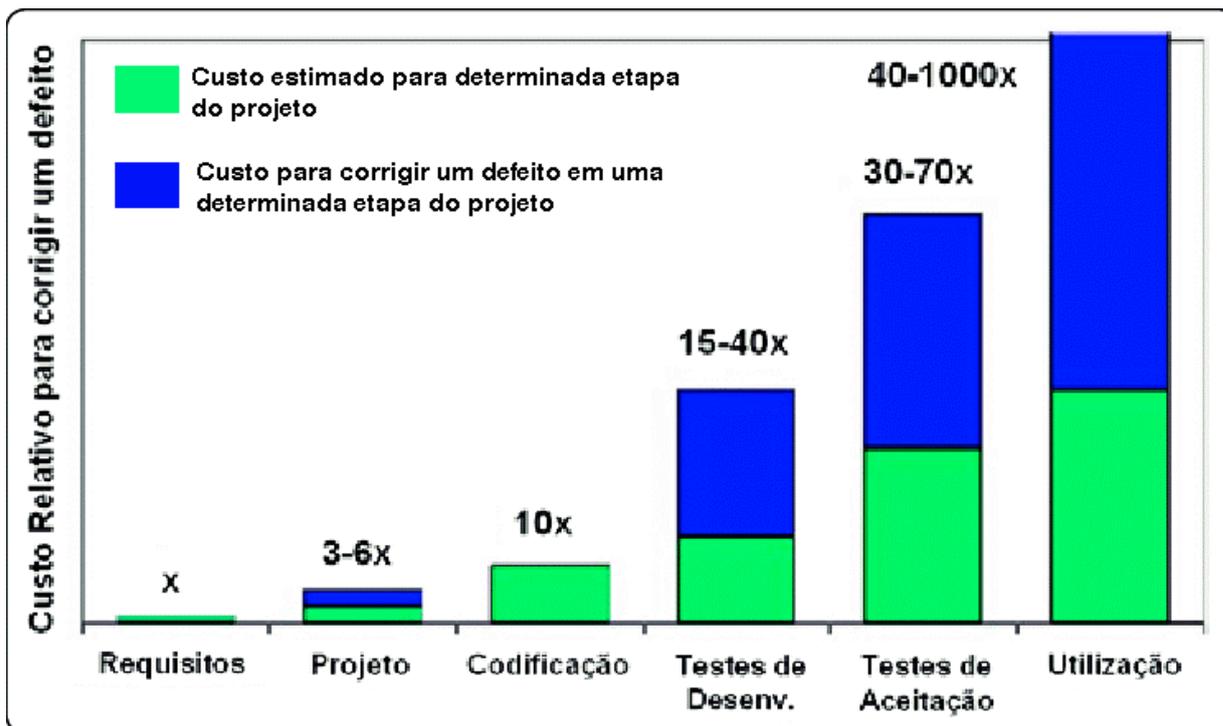


Figura 1 – Custo relativo para corrigir um defeito (Adaptado da Fonte).

Fonte: <https://odonodoproduto.com/o-papel-do-product-owner-na-qualidade-do-produto/>

A busca pela melhoria da qualidade do produto faz com que as equipes de desenvolvimento possam garantir que a arquitetura e estrutura do software sejam confiáveis e sólida, além de evitar que problemas iniciais ou que ocorram durante o decorrer do projeto, possam comprometer a estabilidade do sistema ou funcionalidades futuras, resulta em um produto final de maior qualidade. No decorrer do projeto, uma presença de soluções abrangentes de testes permite que a equipe mantenha um ritmo de desenvolvimento constante e eficiente. Quando problemas capturados por testes de rotina são resolvidos logo que surgem, evita-se que o fluxo de trabalho seja interrompido e a necessidade de retroceder etapas anteriores do desenvolvimento em um projeto ágil, que podem custar tempo de trabalho precioso da equipe, além de aumentar a dívida técnica do projeto. Estes erros podem acontecer em qualquer atividade humana e são muito evidentes na área de desenvolvimento de software. Para Sandhof esses erros têm como causa “Falta de especificações, falta de especialização de profissionais nas tecnologias exigidas pelo mercado, falta de conhecimento do tipo de sistema desenvolvido, tudo isto são evidências do dia-a-dia de empresas produtoras de software” (SANDHOF, 2004 p.62).

O desenvolvimento de software guiado por testes é uma técnica conhecida como TDD (*Test-driven development*) que provê aos desenvolvedores a possibilidade de escrever pequenos pedaços de software baseado em requisitos de software e iterações curtas. Implementar casos de testes no código fonte durante o decorrer do projeto é importante, além de efetuar melhorias no código fonte com certa frequência. Essa maneira de desenvolver softwares faz com que os desenvolvedores da equipe consigam concentrar sua atuação nos requisitos que o projeto exige, sendo assim cada parte do código fonte sendo coberto por casos de teste, consequentemente o código fonte fica mais limpo, garantindo um código fonte produzido de qualidade (DE MATOS, 2020). Dessa forma é mais simples de manter e evoluir o produto, pois a cada etapa de produção, minimiza as chances de acontecer o acúmulo de uma dívida técnica no projeto,

tornando a implementação de novas funcionalidades e a manutenção desse software mais simples ao longo do tempo.

A promoção da cultura de qualidade dentro da equipe de desenvolvimento incentiva os colaboradores a adotarem boas práticas de codificação junto aos testes, tornando um ambiente de trabalho focado na excelência e na melhoria contínua do produto. A entrega de um produto final aonde os clientes e usuários finais são beneficiados diretamente, visto que, o produto passou por um rigoroso controle de qualidade desde os primeiros estágios do projeto. Ao usar um produto, percebendo-se que não há recorrência de falhas ou interrupções durante a sua utilização, acarreta em uma melhor experiência do usuário final, sendo assim, maior é a sua receptibilidade, podendo se dizer que esse produto é de qualidade, aumentando a reputação da empresa que desenvolveu, pois um cliente satisfeito com um produto entregue, é um cliente que tem a sensação de uma boa experiência de acordo com o dinheiro investido para adquirir e utilizar tal produto ou serviço (RAMOS et al., 2016).

3 Objetivo

Este trabalho tem por objetivo demonstrar o impacto na produtividade das equipes de desenvolvimento de software que utilizam métodos ágeis, ao implementar técnicas de qualidade de software, como por exemplo TDD. Foi elaborada uma prova de conceito que foi aplicada em um cenário real do mercado de trabalho de tecnologia, com o intuito de mensurar o quanto a equipe poderia aumentar sua produtividade de forma qualitativa, caso fosse implementado testes automatizados de software para realizar uma validação em uma jornada do cliente no site da empresa, que era feita de forma manual pelos membros da equipe. Em busca de melhorar a qualidade do software entregue, assim como entregar mais em menos tempo, como essas soluções podem ser aplicadas em outros projetos, pois, da mesma forma que essa solução trouxe benefícios para o projeto onde foi aplicado essa prova de conceito, ela abrange o escopo de outros produtos que podem ser beneficiados com essa solução também. Um dos principais tipos de problemas ao fazer a revisão de um código fonte de software é a baixa cobertura de testes automáticos, pois é utilizado como medida de qualidade de um produto, já que a cobertura do código fonte por meio de um caso de teste garante a verificação e a validação desse código (DE MATOS et al., 2018).

4 Metodologia

O embasamento teórico desse estudo sobre o impacto na produtividade de equipes ágeis utilizando técnicas voltadas para qualidade de software, foi escolhida uma abordagem metodológica aplicada, com foco na orientação qualitativa, estudar de forma objetiva o impacto de acordo como ele ocorre no contexto abordado (LÖSCH; RAMBO; FERREIRA, 2023). Foi aplicada uma revisão bibliográfica como método de pesquisa, buscando ampliar o conhecimento sobre o tema e identificar as causas dos problemas relacionados à produtividade no desenvolvimento de software utilizando métodos ágeis. Para o desenvolvimento dessa pesquisa foram utilizadas diferentes bases científicas, como por exemplo Google Scholar e Portal Capes.

A natureza aplicada dessa pesquisa é gerar conhecimentos que possam ser diretamente aplicados na solução dos problemas identificados no contexto do desenvolvimento de software em equipes ágeis. A pesquisa teve como foco o atingimento de resultados que possam ser implementados no ambiente de trabalho para melhorar a qualidade do software entregue e aumentar a produtividade da equipe.

Foi adotada uma abordagem qualitativa para medir de forma comportamental, para que fosse possível coletar informações e o impacto da implementação de técnicas de qualidade de software na produtividade das equipes ágeis. Isso envolve analisar as horas trabalhadas, os recursos envolvidos e das melhorias alcançadas ao implementar as resoluções propostas desse estudo.

Buscando responder os objetivos de forma explicativa, essa pesquisa visa também identificar as causas que comprometem a produtividade e entender como as soluções propostas podem contribuir para melhorar a eficiência das equipes ágeis. Essa abordagem busca analisar os eventos ocorridos, mas também explicar como que métodos influenciam na produtividade no desenvolvimento de software ágil, já que metodologias baseadas em processos preditivos conseguem oferecer melhor soluções e resultados frente à metodologias orientadas a processos (CONCEIÇÃO; SILVEIRA, 2018).

A realização de uma Prova de Conceito (POC) permite a estimação do tempo de horas trabalhadas e o potencial de economia do tempo que poderia ser alcançado com a execução das soluções propostas. Essa análise fornece dados que podem ser usados para avaliar o impacto das técnicas de qualidade de software na produtividade das equipes ágeis em um cenário real de mercado de trabalho voltado para tecnologia.

Em resumo, a metodologia aplicada combina uma abordagem orientada para resultados alcançáveis, uma análise para medir o impacto e objetivos que buscam explicar as causas dos problemas e as estratégias para melhorar de forma qualitativa a produtividade das equipes ágeis no desenvolvimento de software.

5 Desenvolvimento

Foi realizada uma POC (Prova de conceito) com o intuito de verificar se a solução proposta de implementar testes automatizados de software em uma atividade rotineira da equipe, poderia ser viabilizada, se atenderia a necessidade da equipe e fosse trazer qualquer tipo de retorno para a empresa, como por exemplo, poupar horas de trabalho dos colaboradores de uma equipe ágil. O motivo que gerou essa solução foi de que uma tarefa que era realizada com certa recorrência, acabava impactando a produtividade da equipe, pois se tratava de uma tarefa que era feita de forma manual, que após sugestão da liderança imediata, verificou que esse processo manual poderia ser automatizado.

A tarefa objetivo desse caso de estudo, é realizada de forma manual, onde precisa-se fazer a verificação de todos os campos do formulário da venda de um determinado produto da empresa. Essa jornada do cliente possui 9 telas diferentes com dezenas de campos a serem verificados, conforme Figura 2, cada campo de informação do cliente precisa ser testado, para garantir que não seja inserido nenhum tipo de caractere inválido para o campo, se o campo não foi preenchido, verificar se a mensagem de erro está sendo exibida e, necessário verificar se está sendo exibida a mensagem de erro correta para cada erro que pode acontecer no campo.

Queremos saber mais sobre você!
Preencha os campos abaixo com seus dados:

!
Por favor, digite um nome válido

!
O email é inválido.

!

!
Celular inválido.

!

!
O valor mínimo é de R\$ 40.000,00.

!

Li e concordo com o(a) **Aviso de Privacidade**.
Também estou ciente que a pode enviar ofertas e atualizações por e-mail, WhatsApp, SMS e realizar ligações telefônicas.
O campo para aceitar a política de privacidade é obrigatório.

VER VALOR DE PARCELAS

Figura 2 – Primeira tela do formulário da jornada do cliente.
Fonte: Website da empresa objeto de estudo

Durante o desenvolvimento desse estudo de caso, foi realizado um levantamento junto à equipe ágil de desenvolvimento de software que, por se tratar de uma jornada de cliente que era comum a dezenas de produtos oferecidos pela própria empresa, poderia ser replicado essa mesma solução de usar testes automatizados de software para realizar essa mesma tarefa manual, com todos os outros produtos da empresa que utilizam essa mesma jornada do cliente. A ferramenta utilizada para desenvolver o teste automatizado de software nesse estudo de caso foi o *Cypress.js*, que é um *framework* (ferramentas prontas usadas para auxiliar o desenvolvimento de software) de testes automatizados *end-to-end* (de ponta a ponta da aplicação) que usa a linguagem de programação *Javascript*.

A escolha da ferramenta foi motivada pelo fato de que a tarefa era realizada no website da empresa, dessa forma, após reunião com os desenvolvedores da equipe, foi comentado que o *Javascript* é bastante utilizado pelos navegadores de website para carregar as páginas, o que se tornou bastante interessante utilizar o *Cypress.js*, pois como o objetivo do estudo de caso era aplicar testes de software para realizar validações em um website, além de ser uma ferramenta muito intuitiva, possui uma documentação completa e, também recebe muitas atualizações de melhoria.

O teste automatizado de software foi desenvolvido para que a ferramenta selecionasse os elementos da página que seriam validados, no caso desse estudo, são os campos do formulário da jornada do cliente. Para cada campo era realizado diversos testes de validação por meio da ferramenta de forma automática, com o intuito de verificar se o preenchimento do campo tinha sido feito da forma correta, conforme a necessidade da demanda da equipe citada anteriormente nesse artigo.

Conforme pode ser observado na Figura 3, essa é a parte esquerda da interface visual da ferramenta *Cypress*, podemos ver o relatório que é gerado enquanto o teste acontece, marcado em vermelho no relatório é o elemento selecionado e a verificação que está sendo feita em um campo.

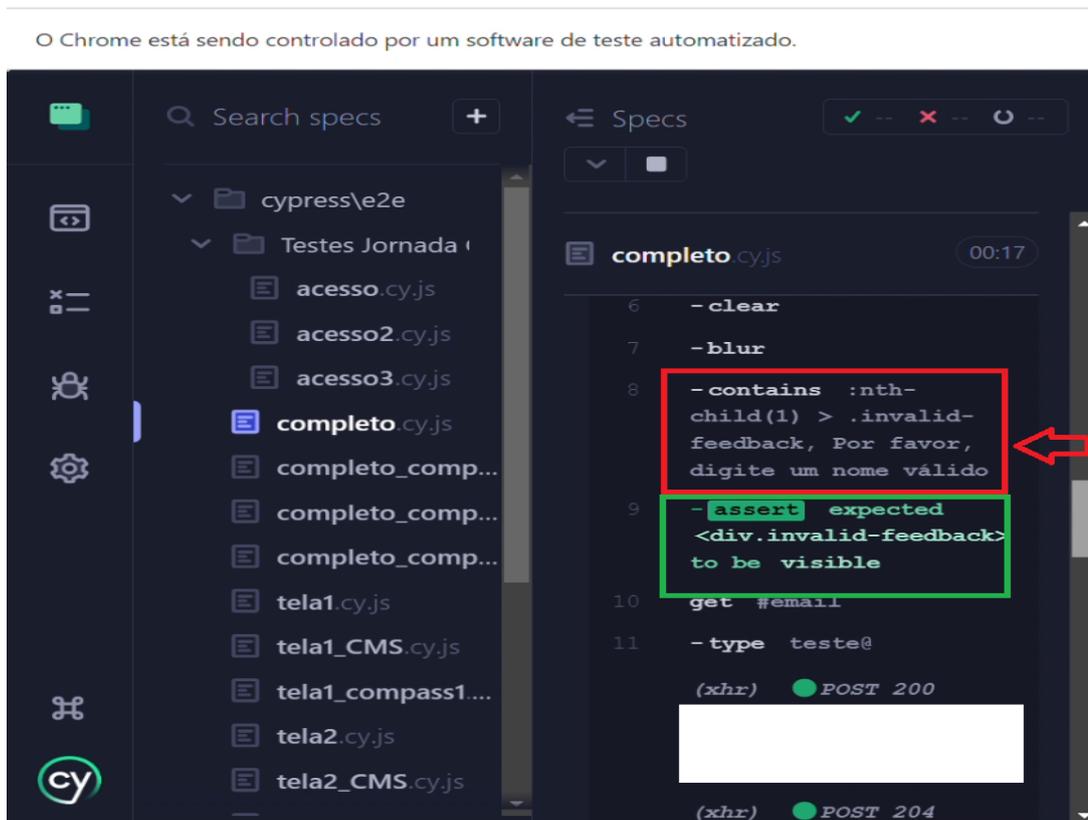


Figura 3 – Interface gráfica da ferramenta.
Fonte: Ferramenta utilizada nesse caso de estudo.

Conforme Figura 4 pode-se observar a parte direita da interface, o que está marcado em vermelho é o campo selecionado e a mensagem de erro que está sendo validada. Nessa validação, como o resultado esperado do teste foi alcançado, é exibido no relatório uma mensagem de sucesso que está marcada em verde na Figura 3. Caso aconteça alguma inconformidade, incompatibilidade ou algum comportamento diferente do resultado esperado de um teste ou validação, a ferramenta aponta no relatório que o teste falhou e faz uma descrição sobre o erro que aconteceu.

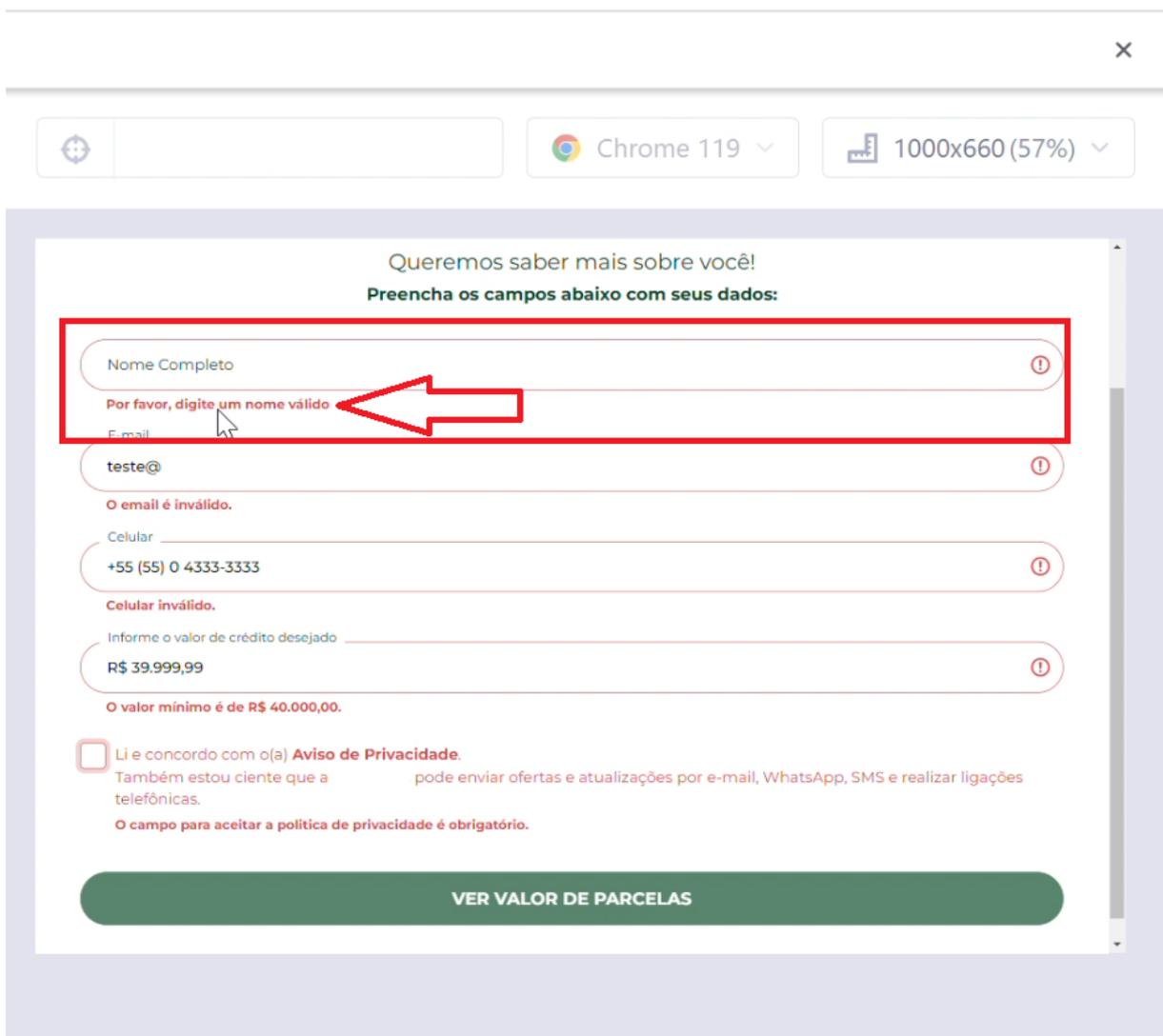


Figura 4 – Interface gráfica da ferramenta.
Fonte: Ferramenta utilizada nesse caso de estudo.

6 Resultados e Discussões

Ao analisar os impactos da implementação de testes de software na rotina de desenvolvimento de software em equipes ágeis, é importante discutir também sobre os desafios que podem surgir ao introduzir técnicas de qualidade de software na rotina das empresas, como por exemplo, o tamanho da equipe de desenvolvimento, se a empresa está disposta a inovar seus processos, a tecnologia que está sendo desenvolvida o software, se existe mão de obra qualificada para realizar essas implementações na empresa, se existem ferramentas suficientes para implementar esses testes de acordo com a tecnologia que a empresa utiliza, além de outros possíveis desafios.

Considerando a necessidade de um profissional qualificado para realizar a implementação dessas técnicas, pesquisas confirmam essa demanda por especialistas certificados no mercado de tecnologia da informação, apesar de muitos profissionais estarem buscando certificações, a demanda por esses profissionais não acompanha a oferta dos mesmos profissionais. Dessa forma pode se observar a importância de ter um profissional qualificado responsável por tais implementações, já que a falta do conhecimento e competência são um dos

maiores riscos para um projeto de desenvolvimento de software (DE SOUZA; LUCIANO, 2004).

Respondendo à problemática de como os testes de software impactam na produtividade do desenvolvimento de software em equipes ágeis, após todas as etapas de pesquisa e desenvolvimento desse trabalho, foi possível identificar os seguintes aspectos, dentre eles, foi observado que após esse estudo de caso, a cada mês de trabalho da equipe ágil de desenvolvimento de software, poderiam ser economizadas 4 horas por *Sprint* (período de tempo definido o qual uma equipe ágil se concentra em trabalhar em um conjunto específico de tarefas e objetivos) implementando a solução proposta de teste automatizado de software para cobrir a validação principal do caso de estudo, sendo que a cada mês ocorrem duas *Sprints*, no total de 8h de trabalho por mês, levando em conta os 12 meses do ano, podendo ser economizados até 96h de trabalho da equipe por ano. Essas horas de trabalho economizadas da equipe, poderiam ser utilizadas em outras frentes, funcionalidades e podendo também trabalhar em outras demandas diferentes que estão aguardando as próximas *Sprints*.

Automatizar testes traz segurança à equipe para realizar tarefas como alteração de código, adicionar novas funcionalidades, tornando a manutenção mais simples, podendo aumentar o tempo que é utilizado para verificar o sistema, porém, o tempo utilizado com a identificação e correção de erros diminui (BERNARDO; KON, 2008). Existem outras pesquisas como que comprovam que implementar técnicas de testes automatizados de software em paralelo com o desenvolvimento do código fonte, podem comprometer a produtividade no início do projeto (ALVES, 2017) quando os testes não são feitos, porém, podemos observar com o estudo de caso dessa pesquisa que substituir testes manuais por automatizados, poderia gerar ganho de tempo em horas produtivas de trabalho para a equipe caso fosse implementado. Apesar de algumas empresas não terem rotina alguma de teste, nestes casos, no começo do projeto existe uma ilusão de estar entregando mais, por conta das primeiras etapas do projeto serem mais simples. Porém, os pontos de valor das *Sprints* reduzem em função do tempo por conta de um acúmulo de dívida técnica que vai sendo carregado durante as etapas do desenvolvimento do projeto.

Os resultados dessa pesquisa ajudam a compreender como os times ágeis de desenvolvimento de software podem aumentar de forma qualitativa a produtividade da equipe, a importância da utilização de técnicas de qualidade de software durante o projeto de desenvolvimento de um software, a relação de custo-benefício de utilizar tais técnicas, haja vista que encontrar um defeito no software num momento tardio do projeto pode aumentar exponencialmente o custo para solucioná-lo, a importância de entregar um produto de qualidade para seu cliente e a importância de um profissional capacitado e competente, especializado em qualidade de software na equipe.

7 Conclusões

Ao observar todos os desafios que as empresas que atuam no ramo de tecnologia da informação encontram, esse estudo demonstra a importância da inovação de processos e da implementação de novas soluções para elevar a eficiência das equipes de desenvolvimento, além de aumentar a qualidade no desenvolvimento de software. Foi demonstrado através do estudo de caso que a implementação de testes automatizados de software pode economizar tempo significativo nas *Sprints*, além de otimizar a produtividade e de certa forma garantir um produto final melhor. Além dos resultados citados acima, deve-se considerar que essa implementação contribui para uma entrega de produtos com menos falhas e com maior

qualidade, dessa forma alcançando a satisfação do cliente e garantindo sua competitividade no mercado.

A pesquisa traz a atenção para a importância de profissionais qualificados e certificados em técnicas de qualidade de software, visto que a falta de conhecimento especializado pode trazer sérios riscos aos projetos de desenvolvimento de software. A utilização de ferramentas como o *Cypress.js* demonstrou-se eficaz não apenas pela sua compatibilidade com tecnologias web, mas também por ser intuitiva, versátil e de fácil adoção pelos desenvolvedores.

As reflexões trazidas por essa pesquisa, demonstram que a implementação de testes de software em equipes ágeis não só aumenta a qualidade do produto final entregue ao cliente, mas também oferece um excelente custo-benefício ao reduzir possíveis custos associados a encontrar um defeito no software de forma tardia nos ciclos de desenvolvimento. Dessa forma, investir em técnicas de qualidade de software e na aquisição de profissionais capacitados da área de qualidade para a equipe, são fundamentais para o sucesso e a competitividade das empresas de tecnologia da informação que atuam nesse segmento da indústria de software.

Agradecimentos

Gostaria de agradecer a Deus por me permitir alcançar essa conquista, pois, desde novo sempre sonhei em cursar uma graduação em tecnologia da informação, sem Ele nada disso seria possível. Deixo aqui meus sinceros agradecimentos ao meu orientador Igor Isnardi Barreto, por toda sua dedicação e empenho ao longo desse trabalho, por nos ensinar a importância de entregar um trabalho de qualidade e de sempre buscar dar o nosso melhor, tanto na vida acadêmica quanto na vida profissional. Agradeço por toda paciência, instrução, orientação que foram fundamentais para a conclusão desse trabalho

À minha família, gostaria de agradecer aos meus pais que sempre fizeram de tudo para me proporcionar as melhores oportunidades de estudo e sempre investiram na minha educação. Em especial à minha mãe Maria Izabel Fajersztajn que foi uma verdadeira guerreira, por tudo que passou e mesmo assim nunca se deixou abalar por causa das circunstâncias, foi um exemplo de mãe, pessoa e será pra sempre minha heroína. Tenho certeza que ela estaria muito orgulhosa dessa conquista. Ao meu irmão Raphael Fajersztajn que sempre me apoiou nos momentos mais difíceis, sempre esteve ao meu lado e foi meu maior exemplo. Me ensinou desde pequeno o quanto importante é estudar e se não fosse por ele ter me chamado para vir morar em São José do Rio Preto, eu não estaria concluindo esse curso. À minha madrinha Adalgisa Ferreira da Silva sou muito grato por todo apoio que ela sempre prestou a nossa família, por ter sempre nos acompanhado, protegido e nos guiado desde pequenos. Nos momentos mais difíceis sempre se fez presente e foi essencial para que nós superássemos esses momentos.

Gostaria de agradecer ao meu gestor Paulo Colucci por ter me acompanhado no projeto da POC e pela ideia do artigo, ao meu líder Denys Rodrigues pela ideia da POC e ao Rafael Loures de Souza que me ensinou a programar os testes automatizados e a importância sobre o tema da qualidade de software que despertou o meu interesse na área.

Referências

ALVES, Gabriella Mayara Tavares. **Automatização de testes em equipes ágeis: um estudo qualitativo usando teoria fundamentada**. 2017. 82 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Campina Grande, 2017. Cap. 6. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/xmlui/handle/riufcg/1384>. Acesso em: 10 jun. 2024.

BERNARDO, Paulo Cheque; KON, Fabio. A importância dos testes automatizados. **Engenharia de Software Magazine**, v. 1, n. 3, p. 54-57, 2008. Disponível em: <https://www.academia.edu/download/74461621/EngSoftMagazine-IntroducaoTestes.pdf>. Acesso em: 03 fev. 2024.

CONCEIÇÃO, Juliano; SILVEIRA, Sidnei Renato. Aplicação de Metodologias Ágeis para Desenvolvimento de Software: um estudo de caso na empresa Alliance Software. **Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica**, v. 9, n. 1, 2018. Disponível em: <http://periodicos.unifacef.com.br/resiget/article/view/1520>. Acesso em: 20 mai. 2024.

DE MATOS, Altieres et al. Estudo preliminar sobre os impactos do desenvolvimento orientado a testes no processo de revisão de código. In: **Anais da II Escola Regional de Engenharia de Software**. SBC, 2018. p. 147-156. Disponível em: <https://sol.sbc.org.br/index.php/eres/article/view/10068> Acesso em: 17 mai. 2024.

DE MATOS, Altieres. **Efeitos do uso do desenvolvimento orientado por testes em conjunto com critérios de teste na indústria de software ágil**. 2020. 73 f. Dissertação – Mestrado, Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020. Disponível em: <https://magsilva.pro.br/publications/Matos-MscThesis-2020.pdf>. Acesso em: 28 jan. 2024.

DE SOUZA, Eduardo Paulo. **Elementos fundamentais na melhoria da qualidade de software**. 2004. Tese de Doutorado. [sn]. Disponível em: <https://repositorio.unicamp.br/Busca/Download?codigoArquivo=472895>. Acesso em: 03 fev. 2024.

DE SOUZA, Alessandro Nunes; LUCIANO, Edimara Mezzomo. O papel da certificação do profissional de TI na qualidade do software: Uma investigação preliminar. In: **Anais do III Simpósio Brasileiro de Qualidade de Software**. SBC, 2004. p. 74-87. Disponível em: <https://sol.sbc.org.br/index.php/sbqs/article/view/16185>. Acesso em: 01 jun. 2024.

LÖSCH, Silmara; RAMBO, Carlos Alberto; FERREIRA, Jacques Lima. A pesquisa exploratória na abordagem qualitativa em educação. **Revista Ibero-Americana de Estudos em Educação**, p. e023141-e023141, 2023. Disponível em: <https://periodicos.fclar.unesp.br/iberoamericana/article/download/17958/17250?inline=1>.

Acesso em: 20 mai. 2024.

RAMOS, Mayara et al. Design de Serviços e Experiência do Usuário (UX): uma análise do relacionamento das áreas. **DAPesquisa**, v. 11, n. 16, p. 105-123, 2016. Disponível em: <https://www.revistas.udesc.br/index.php/dapesquisa/article/view/6378>. Acesso em: 20 mai. 2024.

SANDHOF, Karen. **Fatores humanos no processo de desenvolvimento de software: um estudo visando qualidade**. 2004. Tese de Doutorado. Universidade de São Paulo. Disponível em: <https://www.teses.usp.br/teses/disponiveis/3/3141/tde-15122004-075221/en.php>. Acesso em: 17 mai. 2024.

SARCHESI, Giovani França; DINIZ, Débora Pelicano. **SOFTWARE PARA OTIMIZAR GERAÇÃO DE COMBINAÇÕES UTILIZANDO O MÉTODO PAIRWISE**. 2021. 12 f. TCC (Graduação) - Curso de Informática, Faculdade de Tecnologia de Ribeirão Preto, Ribeirão Preto, 2021. Cap. 2. Disponível em: <http://www.fatecrp.edu.br/WorkTec/edicoes/2021-1/trabalhos/III-Worktec-SOFTWARE%20PARA%20OTIMIZAR%20GERAC%CC%A7A%CC%83O%20DE%20COMBINAC%CC%A7O%CC%83ES%20UTILIZANDO%20O%20ME%CC%81TODO%20PAIRWISE.pdf>. Acesso em: 17 maio 2024.

SILVA, Aylma Benício da. Estratégia na qualidade de software: um comparativo sobre as ferramentas gerenciais: jira x ALM. 2023. Disponível em: <https://repositorio.ifpe.edu.br/xmlui/bitstream/handle/123456789/817/Tcc%20Aylma%20Ben%C3%ADcio-%20incluso%20ficha%20catalogr%C3%A1fica.pdf?sequence=1>. Acesso em: 17 mai. 2024.