

A eficácia dos testes automatizados para melhorar a satisfação do cliente em projetos de desenvolvimento de software

Michele Ramos Borowski¹, Kassya C. R. de Andrade¹

¹Faculdade de Computação e Informática (FCI)
Universidade Presbiteriana Mackenzie – São Paulo, SP – Brasil

10390565@mackenzista.com.br,

kassyachristina.andrade@mackenzie.br

Resumo. Atualmente os testes de usabilidade são fundamentais para o desenvolvimento de software, especialmente diante de um mercado altamente competitivo, onde as expectativas dos usuários em relação à qualidade e à experiência de uso são cada vez mais elevadas. O objetivo deste trabalho é avaliar a eficácia dos testes de usabilidade como um fator determinante para a melhoria da satisfação do cliente, identificando práticas e estratégias que contribuam para a criação de softwares mais alinhados com as expectativas dos usuários. Foi realizada uma pesquisa bibliográfica acompanhada de uma análise qualitativa de dois artigos relevantes sobre o tema. Identificou-se que a adoção de estratégias, como o planejamento antecipado dos testes, a definição clara de objetivos e a avaliação da efetividade das tarefas ou da satisfação dos usuários, contribui significativamente para a eficiência dos processos de teste. Os resultados também indicam que a automação, quando aplicada de forma estratégica, não apenas melhora a qualidade do produto, mas também aprimora a percepção do cliente em relação à solução desenvolvida.

Palavras-chave: Teste de usabilidade; Satisfação do cliente; Automação de teste; Qualidade de software.

Abstract. Currently, usability testing is fundamental for software development, especially in a highly competitive market where user expectations regarding quality and user experience are increasingly high. The objective of this work is to evaluate the effectiveness of usability testing as a determining factor for improving customer satisfaction, identifying practices and strategies that contribute to the creation of products more aligned with user expectations. Bibliographic research was conducted along with a qualitative analysis of two relevant articles on the topic. It was identified that the adoption of strategies such as early planning of tests, clear definition of goals, and the evaluation of task effectiveness or user satisfaction significantly contributes to the efficiency of testing processes. The results also show that automation, when strategically applied, not only improves product quality but also enhances the customer's belief of the developed solution.

Keywords: Usability test; Customer satisfaction; Test automation; Software quality.

1. Introdução

Conforme Silvestre e Scombatti Pinto (2023), os softwares são recursos valiosos desenvolvidos para atender a diversas necessidades, como simplificar tarefas diárias, proporcionar entretenimento, automatizar processos industriais, gerenciar informações nas empresas e melhorar a comunicação, entre outras funções. De todo modo, é comum que surjam desafios durante o desenvolvimento, como problemas com o tempo, qualidade, desempenho e custo, estes podem impactar diretamente a satisfação dos clientes (Kaplan; Norton, 2005).

É fundamental o papel que os testes de usabilidade desempenham para garantir que o produto funcione bem e atenda às necessidades do usuário, a autora Barnum (2021) informa que os testes são descritos como o processo de obter *insights* sobre os usuários diretamente por meio da observação deles ao utilizarem um produto para realizar tarefas reais e alcançar objetivos de interesse pessoal. É possível identificar se o desenvolvimento do produto atende às expectativas do cliente, por meio dos testes de software (CAMPOS, 2024). Para que o software realmente atenda a essas necessidades e traga satisfação aos usuários, é fundamental que opere como previsto, livre de erros e defeitos (CAMPOS, 2024).

A falta de testes de usabilidade bem realizados agrava esses desafios, pois descobrir estes problemas tarde demais acaba prejudicando os prazos, fazendo com que tenha maior tempo para a finalização do projeto, aumentando os custos e comprometendo a qualidade do produto final, colocando em risco tanto a experiência do usuário quanto a credibilidade do produto, tornando essencial a aplicação de métodos eficazes para assegurar que o cliente esteja satisfeito.

Esse trabalho tem como objetivo geral avaliar a eficácia dos testes de usabilidade como um fator determinante para a melhoria da satisfação do cliente em projetos de desenvolvimento de software, identificando práticas e estratégias que contribuam para a criação de produtos mais eficientes e alinhados com as expectativas dos usuários.

Os objetivos específicos desta pesquisa são:

- Analisar o impacto da ausência ou falta de qualidade de testes de usabilidade em projetos de software e como isso afeta a satisfação do cliente;
- Identificar os principais problemas decorrentes da falta de testes de usabilidade;
- Investigar as melhores práticas de testes de usabilidade em projetos de desenvolvimento de software, destacando ferramentas e abordagens mais eficazes;
- Avaliar a relação entre a qualidade dos testes de usabilidade realizados durante o desenvolvimento e o nível de satisfação do cliente com o produto final;
- Identificar os principais desafios enfrentados pelas equipes de desenvolvimento na implementação de testes de usabilidade;
- Análises e considerações finais encontrados na aplicação de testes de usabilidade;

A escolha deste tema se justifica pela crescente importância dos testes de usabilidade no cenário atual de desenvolvimento de software, devido a ampla concorrência e o aumento das expectativas dos usuários, garantir a boa experiência do cliente é um fator determinante para o sucesso de produtos digitais, compreender como os testes de usabilidade podem melhorar a satisfação do cliente é essencial para empresas que buscam desenvolver softwares de alta qualidade e competitividade no mercado.

Em um estudo de Barboza et al. (2024), analisam a eficiência do aplicativo Avazum, focando nos aspectos de navegabilidade e o nível de esforço dos usuários para atingirem seus objetivos. Eles destacam a visibilidade dos elementos do aplicativo, a utilização de uma linguagem adequada e a clareza das informações durante a realização das tarefas como fatores fundamentais para a satisfação dos usuários, já que isso está diretamente relacionado à eficiência, uma vez que aplicativos fáceis de usar e que permitem conclusão de tarefas com sucesso geram maior satisfação.

Os autores concluem que o aplicativo Avazum atinge seus objetivos de forma eficiente, apresentando uma linguagem clara e confortável durante o uso. Dessa forma, a pesquisa de Barboza et al. (2024) evidencia que, ao conduzir testes de usabilidade eficazes, as empresas podem aumentar a satisfação do cliente, tornando o software mais atrativo e competitivo no mercado.

2. Referencial Teórico

Este referencial teórico tem como objetivo apresentar os assuntos e conceitos mais importantes que serão tratados na pesquisa.

A qualidade no desenvolvimento de software é um fator crítico para o sucesso de qualquer produto ou projeto, uma vez que impacta diretamente na satisfação do cliente e na eficiência do sistema (Pressman, 2019). O teste de software é uma das principais práticas que garantem qualidade, ao identificar e corrigir defeitos antes que o produto alcance o usuário final. Dessa forma, além de reduzir custos de manutenção, o teste aumenta a confiabilidade do sistema, contribuindo para uma experiência mais satisfatória ao utilizador.

Nesse contexto, Silvestre e Scombatti Pinto (2023), destacam que a aplicação correta dos testes de software é essencial para avaliar a prontidão do produto no mercado. Testes bem estruturados oferecem uma oportunidade de melhorar o processo de desenvolvimento, proporcionando uma experiência de excelência ao usuário. Ao aplicar metodologias rigorosas de teste, as organizações são capazes de encontrar problemas e inconsistências que poderiam comprometer a experiência do usuário e, conseqüentemente, a satisfação do cliente. Isso se torna ainda mais relevante em um mercado competitivo, onde a percepção do cliente em relação à qualidade do software pode influenciar diretamente a sua decisão de uso e recomendação do produto.

Seguindo essa linha, a prática de testes de usabilidade surge como uma ferramenta essencial para assegurar a qualidade em projetos de software, focando especialmente no que diz respeito à interação do usuário com o sistema. A usabilidade é uma medida que indica como os usuários podem utilizar produtos para alcançar objetivos específicos com eficácia, eficiência e satisfação. O padrão ISO 9241-10 detalha os benefícios de avaliar a usabilidade em relação à satisfação e ao desempenho do usuário durante a realização de tarefas. Dado que o objetivo deste trabalho é analisar a usabilidade, esse padrão serve como um guia fundamental. Além disso, conforme o padrão ISO 9241-11, a satisfação dos usuários também envolve a ausência de desconforto, o prazer no uso do produto e a aceitação da carga mental imposta pela tarefa. Dessa forma, a satisfação do usuário está

associada aos objetivos de usabilidade, como a eficiência e a facilidade de aprendizado do sistema.

Complementando essa perspectiva, os princípios de usabilidade descritos por Nielsen (2020) são amplamente reconhecidos e aplicados no design de interfaces. Conhecidos como "10 Usability Heuristics for User Interface Design", essas heurísticas oferecem uma base para a criação de sistemas que priorizam a facilidade de uso, como a visibilidade do status do sistema e a prevenção de erros, que garantem que o usuário tenha uma experiência mais fluida e satisfatória. Ao realizar testes de usabilidade com base nesses princípios, é possível identificar e corrigir problemas de interface de forma precoce, resultando em maior satisfação do cliente (NIELSEN, 2020).

Dessa forma, a usabilidade é um componente central para o desenvolvimento de software moderno, com influência direta sobre a satisfação do usuário final. A integração de testes de usabilidade ao longo do processo de desenvolvimento permite a identificação precoce de problemas e a validação de decisões de design com base em feedbacks reais de usuários.

No livro *Usability Testing Essentials: Ready, Set...Test!*, Barnum (2020, p. 17) também contribui para essa discussão ao demonstrar que mesmo com um número reduzido de participantes é possível aprender muito sobre a usabilidade de um sistema. A autora argumenta que com apenas cinco usuários é possível identificar cerca de 85% dos problemas mais críticos de usabilidade. Isso enfatiza que, mesmo com poucos testes, é possível obter insights significativos sobre as áreas problemáticas de uma interface, tornando os testes de usabilidade altamente eficientes e acessíveis, especialmente em projetos de software com orçamentos e prazos limitados.

No entanto, Barnum (2020) discute as forças e fraquezas dos testes de usabilidade. Entre os pontos fortes, ela destaca a capacidade de identificar problemas que os desenvolvedores e designers não haviam previsto, tornando o feedback dos usuários um elemento essencial para o refinamento do produto. No entanto, ela também menciona algumas fraquezas, como a limitação do número de usuários, que pode não ser suficiente para capturar a diversidade completa de interações e preferências dos usuários. Além disso, os resultados podem ser influenciados pelo comportamento subjetivo dos participantes, o que significa que diferentes grupos de usuários podem fornecer diferentes feedbacks para o mesmo sistema. Os testes de usabilidade podem ser classificados em duas categorias principais:

- Teste formativo: realizado durante o desenvolvimento do produto, com o objetivo de identificar e corrigir problemas. Envolve um número reduzido de usuários e é repetido ao longo do processo de design.
- Teste somativo: aplicado quando o produto está concluído ou em fase final de desenvolvimento, com o propósito de validar se ele atende aos requisitos estabelecidos. Esse teste envolve um número maior de usuários e os resultados são geralmente tratados de forma estatística.

Uma abordagem que tem ganhado destaque no desenvolvimento de software é o *Behavior-Driven Development* (BDD). Segundo Nicchio, Leal e Martins (2021), essa

técnica busca auxiliar as equipes a desenvolverem o software com mais eficácia, destacando que, além de ter o objetivo de fazer uma entrega de qualidade, também visa alinhar o software com a estratégia de negócio. Dessa forma, o BDD garante que o software atenda não só às necessidades técnicas, mas também às expectativas do cliente, promovendo uma experiência mais intuitiva e alinhada aos objetivos da empresa. Para conseguir realizar uma prática colaborativa, em que todas as partes interessadas consigam se comunicar sobre as funcionalidades do sistema, o BDD utiliza a linguagem Gherkin para criar cenários, por ser fluida e oferecer maior clareza na obtenção de requisitos e melhora a comunicação (Santos, Soares e Rocha, 2024). Essa linguagem utiliza palavras-chaves para garantir que o cenário seja compreendido de forma coerente. As palavras em inglês são: *Title*, *Given*, que indica a pré-condição para rodar o cenário; *When* que se refere a condição de quando o usuário executa a ação; e *Then* é o resultado esperado do sistema (RAHARJANA; HARRIS; JUSTITIA, 2020). A Figura 1 é um exemplo da linguagem Gherkin, onde o lado (a) mostra um modelo das palavras-chaves e o lado (b) mostra exemplos com dois cenários.

Figura 1: Exemplo da linguagem Gherkin (a) modelo das palavras-chaves (b) Exemplo do modelo com 2 cenários.

<pre>Title : (one line describing the story) Narrative : As a [role], I want [feature] So that [benefit] Acceptance Criteria : (presented as Scenarios) Scenario 1 : Title Given [context] and [some more context] ... When [event] Then [outcome] and [another outcome] Scenario 2 : Title Given [context] and [some more context] ... When [event] Then [outcome] and [another outcome]</pre>	<pre>Title : Registration Form Narrative : As a Facebook user, I want to be able to log in into my account so that I can share my photos Acceptance Criteria : (presented as Scenarios) Scenario 1 : Log in success Given I go to "log in" page visit path_to (login_page) When I fill "Ujicoba@gmail.com" in "inputName" field Fill_in ("inputName" with "Ujicoba@gmail.com") and I fill "1234" in "inputPassword" field Fill_in ("inputPassword" with "1234") and I submit the form click_button("Log in") Then I go to Home login visit path_to (user_home) Scenario 2 : Wrong Username Given I go to "log in" page visit path_to (login_page) When I fill "error@gmail.com" in "inputName" field Fill_in ("inputName" with "error@gmail.com") and I fill "1234" in "inputPassword" field Fill_in ("inputPassword" with "1234") and I submit the form click_button("Log in") Then I go to Error Login page visit path_to (error_login)</pre>
(a)	(b)

Fonte: RAHARJANA; HARRIS; JUSTITIA (2020)

O BDD e o *Test Driven Development* (TDD) são abordagens complementares no desenvolvimento de software: enquanto o TDD é direcionado a testes de unidade e integração, o BDD se concentra em testes de aceitação, ajudando a validar se o software realmente atende às necessidades do usuário final. Em projetos complexos, onde a

satisfação do cliente é um critério essencial para o sucesso, o BDD oferece uma vantagem adicional ao priorizar o ponto de vista do usuário, resultando em produtos mais alinhados com a expectativa do cliente e menos suscetíveis a problemas de usabilidade (NICCHIO; LEAL; MARTINS, 2021).

Por fim, a satisfação do cliente em projetos de software pode ser definida como o grau em que o produto atende ou excede as expectativas e necessidades dos usuários finais. Em termos práticos, a satisfação do cliente é influenciada por uma série de fatores, incluindo a qualidade técnica do software, sua facilidade de uso, confiabilidade. O suporte oferecido pela equipe de desenvolvimento. Segundo Albert e Tullis (2022), clientes satisfeitos tendem a adotar o produto a longo prazo, recomendá-lo a outros e se tornarem defensores da marca, o que reforça a importância de uma experiência de usuário positiva em projetos de software.

A qualidade do software e a satisfação do cliente estão intimamente ligadas. Um produto que apresenta altos padrões de qualidade, incluindo baixa incidência de defeitos, facilidade de manutenção e desempenho estável, tende a gerar maior satisfação entre os usuários. A qualidade percebida do software afeta diretamente a confiança do cliente no produto e, por consequência, a sua fidelidade. Por outro lado, softwares que apresentam falhas constantes ou que não atendem às expectativas do usuário podem levar à frustração e ao abandono do produto. Dessa forma, garantir a qualidade do software é essencial para manter a confiança do cliente e assegurar a sua satisfação ao longo do tempo.

A usabilidade tem um papel crítico na satisfação do cliente. Um software que é fácil de usar, intuitivo e eficiente torna a experiência do usuário mais agradável, o que contribui diretamente para uma avaliação positiva do produto. Como Barnum (2020) aponta, problemas de usabilidade podem ser detectados mesmo com um pequeno grupo de participantes, e ao corrigir esses problemas, as empresas podem evitar frustrações futuras e aumentar a satisfação. Quando os usuários conseguem completar suas tarefas de forma eficiente e sem dificuldades, a probabilidade de sucesso do projeto de software é significativamente maior. Portanto, a usabilidade não só melhora a experiência imediata do usuário, mas também fortalece a relação de longo prazo entre o cliente e o produto.

As métricas de usabilidade são ferramentas fundamentais para avaliar a qualidade de uma interface, especialmente em relação à eficácia, eficiência e satisfação dos usuários. Entre os tipos mais comuns de métricas estão o tempo necessário para completar uma tarefa, a taxa de sucesso, a quantidade de erros cometidos e a satisfação do usuário. Essas métricas são essenciais para medir o desempenho de um sistema e identificar áreas de melhoria, especialmente no que se refere à satisfação dos clientes.

Quanto à definição e importância das métricas, uma métrica eficaz inclui a taxa de erros e a taxa de conclusão de tarefas. O tempo necessário para finalizar uma tarefa, o número de cliques e a relação custo-benefício são indicados como métricas adequadas para avaliar a eficiência. Já a satisfação pode ser mensurada tanto pelas taxas gerais de satisfação quanto pela proporção de usuários que relatam reclamações. Embora a eficácia, a eficiência e a satisfação sejam frequentemente avaliadas de forma quantitativa, essas métricas também podem ser analisadas qualitativamente, como discutindo os erros e

sucessos diretamente com os participantes, permitindo que descrevam sua satisfação com o sistema em suas próprias palavras (AIYEGBUSI, 2019).

3. Metodologia

A metodologia utilizada nesta pesquisa baseia-se em uma análise qualitativa de dois artigos relevantes sobre testes de usabilidade e sua influência na satisfação do cliente em projetos de desenvolvimento de software. Essa escolha visa proporcionar uma investigação aprofundada das técnicas e práticas apresentadas nos estudos, com o objetivo de compreender como esses resultados podem ser aplicados no contexto da pesquisa atual.

Inicialmente, foi realizada uma revisão da literatura com o propósito de estabelecer uma base teórica sólida sobre os principais conceitos relacionados ao tema, servindo como base para a fundamentação e compreensão das questões exploradas. Para isso foram selecionados livros e artigos que discutem a importância da usabilidade e apresentam metodologias de aplicação desses testes. Esse embasamento teórico oferece uma visão mais ampla e sólida, permitindo um entendimento consistente sobre o assunto.

A análise central da pesquisa é composta por dois estudos de caso que examinam a aplicação de testes de usabilidade em diferentes contextos de desenvolvimento de software. Esses estudos foram escolhidos devido à sua relevância e detalhamento sobre a implementação de práticas de usabilidade e os respectivos impactos na satisfação do cliente.

Para encontrar os estudos de caso utilizados nessa pesquisa foi realizada uma pesquisa avançada utilizando combinações estratégicas e palavras-chave. Termos como “testes de software”, “satisfação do cliente”, “projetos de desenvolvimento de software” e “eficácia dos testes de software” foram empregados, com o uso de operadores booleanos para refinar os resultados. Esse processo teve como objetivo localizar estudos alinhados aos objetivos da pesquisa e que trouxessem contribuições relevantes ao contexto analisado.

A partir da análise dos dois estudos de caso, será conduzida uma comparação detalhada para identificar semelhanças e diferenças nas abordagens, observando os impactos de cada método na satisfação do cliente. Os dados foram coletados por meio da leitura e interpretação detalhada dos métodos, desafios e soluções relatados em cada estudo. A análise focou nos seguintes aspectos:

- Qualidade dos Testes: Avaliação da eficácia dos testes em identificar falhas e garantir uma experiência de usuário satisfatória.
- Eficiência e Economia de Tempo: Comparação do tempo de execução e dos benefícios proporcionados pela automação em relação aos métodos manuais.
- Consistência e Estabilidade: Observação da frequência e impacto de falhas nos testes automatizados, além da consistência dos resultados.

As pesquisas realizadas em livros e artigos citados neste referencial teórico foram de suma importância para a compreensão dos conceitos e ferramentas que envolvem os testes de usabilidade. Esta etapa visa identificar a consistência dos resultados observados nos estudos de caso com teorias e práticas recomendadas em usabilidade e automação de

testes. A validação ajudará a contextualizar as conclusões e a oferecer recomendações práticas sobre o uso de testes de usabilidade para a melhoria da satisfação do cliente.

4. Resultados e discussão

Esta seção busca comparar dois estudos de caso que abordam, sob diferentes perspectivas, os desafios e as soluções para garantir um processo de teste eficiente, capaz de entregar um produto de qualidade e que atenda às expectativas e satisfaça o cliente. O primeiro estudo explora a importância da qualidade nos testes automatizados, enquanto o segundo se aprofunda na comparação entre testes manuais e automatizados, revelando os benefícios da automação. A partir dessa análise comparativa, é possível identificar como esses processos impactam diretamente a experiência final do usuário e o sucesso do produto.

Antes de apresentar os estudos de casos, é necessário conhecer as ferramentas de automação de testes que foram utilizadas nesses estudos de caso: o Cypress é um *framework* de testes criado com o propósito de realizar testes frontend, é muito mais rápido ao rodar no browser do que os outros frameworks de teste (MWAURA, 2021). O Selenium também é um framework com um conjunto de ferramentas de teste funcional, possui mais de 10 anos no mercado, tendo uma grande comunidade para dar suporte a novos desenvolvedores (DA SILVA FERREIRA et. al., 2022). Por fim, o Cucumber é uma ferramenta de automação de testes que utiliza a linguagem Gherkin, considerado como uma documentação de requisitos funcionais precisa, compreensível tanto para desenvolvedores quanto para outras partes envolvidas no projeto. (CASTRO; HERCULINO; MENDONÇA, 2023).

O primeiro estudo de caso, de Blascke, Farina e Florian (2023) em seu artigo *Testes de software como garantia de qualidade e eficiência: estudo de caso com uso do selenium*, aborda os desafios de instabilidade e baixa qualidade nos testes automatizados. Os autores identificaram problemas significativos como testes instáveis, que falharam sem uma razão aparente, implicando a confiança nos resultados e a estabilidade do processo de testes; lentidão na execução dos testes, que prolongava o ciclo de desenvolvimento e tornava o processo de entrega menos ágil, prejudicando a eficiência do projeto e a satisfação do cliente; os testes cobriam apenas alguns cenários limitados, deixando de fora casos importantes que poderiam revelar falhas críticas no software, dificultando a identificação de problemas, especialmente em situações de uso real.

A falta de estabilidade nos testes automatizados, como mencionada, pode causar uma entrega de software que não atende às expectativas do cliente, refletindo a importância da execução adequada dos testes de usabilidade. A relação entre a qualidade dos testes e a experiência do usuário final reforça a necessidade de atenção cuidadosa ao planejamento e à execução dos testes, uma premissa central para garantir a satisfação do cliente. Por exemplo, tempos de espera explícitos são úteis em cenários onde o carregamento de componentes depende de variáveis externas, como em testes de interface com Selenium, pois evitam falhas em testes devido a problemas de sincronização. A ampliação da cobertura de testes também é essencial para identificar problemas antes que cheguem ao usuário final. Essas medidas mostram como ajustes relativamente simples podem melhorar significativamente a qualidade do software entregue.

A paralelização dos testes também se destacou como uma estratégia eficaz para aumentar a velocidade de entrega, permitindo um ciclo de feedback mais rápido e possibilitando que múltiplos testes sejam executados simultaneamente em diferentes ambientes e dispositivos. Essa conexão entre a melhoria dos processos de teste e a entrega mais ágil e precisa de software é primordial para entender o impacto positivo na satisfação do cliente.

Sendo assim, o estudo destaca que a simples automação dos testes, sem a devida atenção à sua qualidade e estabilidade, pode não alcançar os resultados desejados. Isso implica que, além da automação, o planejamento, a execução cuidadosa e as correções são fatores-chave para assegurar que o produto final seja satisfatório em termos de usabilidade.

O segundo estudo de caso de Campos (2024) intitulado como *Proposta para implantação de automação de testes de software usando behavior driven development (BDD) - Estudo de caso*, complementa essa visão ao comparar diretamente os testes manuais e automatizados, mostrando os benefícios quantitativos da automação em termos de economia de tempo e consistência nos resultados. A autora identificou um problema inicial de tempo elevado na configuração da automação, que exigiu um esforço considerável para desenvolver os scripts de teste e integrar a metodologia BDD ao processo. Esse desenvolvimento inicial foi mais demorado em comparação aos testes manuais, mas, uma vez configurado, a automação trouxe benefícios significativos em termos de economia de tempo, reduzindo o tempo de execução dos testes de 10 horas (testes manuais) para 5 horas e 1 minuto. Essa economia de tempo impacta diretamente o processo de desenvolvimento, permitindo uma entrega mais rápida, o que é uma das principais expectativas dos clientes em projetos de software.

Outro ponto relevante do estudo foi a consistência na identificação de bugs em ambos os métodos, o que indica que a automação pode garantir resultados precisos sem perder a eficiência, algo essencial para a manutenção da qualidade e estabilidade do software. A rapidez na execução dos retestes, realizada em apenas 33 segundos, reforça ainda mais essa vantagem, tornando o ciclo de correção de bugs mais ágil e eficiente, que é fundamental para reduzir o tempo de resposta e manter a satisfação do cliente durante o desenvolvimento do software.

A automação também contribuiu para a liberação de recursos humanos que antes eram dedicados a testes repetitivos, possibilitando que esses profissionais fossem direcionados a tarefas mais estratégicas. Isso não apenas otimiza a alocação de equipe, mas também reduz custos a longo prazo, uma vez que os testes automatizados podem ser executados múltiplas vezes sem custos adicionais de mão de obra, aumentando a produtividade do time como um todo.

Além disso, a integração da metodologia BDD aos testes automatizados trouxe um benefício adicional ao processo de automação, pois possibilita a identificação de problemas com maior antecedência e a ação proativa para evitar falhas maiores. Essa abordagem se mostrou benéfica para o cliente, pois o software final tem menor chance de regressões ou erros que comprometam a experiência do usuário, reduzindo também os custos de correção ao longo das etapas do desenvolvimento, e possibilitar um aumento da cobertura de testes, gerando mais segurança no produto final. Conforme mencionado por Luna (2024), a usabilidade nos projetos de software contribui para a redução de custos de desenvolvimento e manutenção ao melhorar a eficiência operacional e diminuir as

despesas com suporte e documentação, fatores que favorecem tanto a experiência do usuário quanto os resultados das organizações.

No segundo estudo de caso, foram implementadas diversas soluções que visam otimizar o processo de validação de funcionalidades. Entre as principais iniciativas, destaca-se o desenvolvimento de scripts de automação baseados na sintaxe Gherkin. Esta escolha permitiu que os scripts fossem escritos em uma linguagem acessível, compreensível tanto para desenvolvedores quanto para colaboradores sem formação técnica, facilitando o entendimento e o alinhamento de expectativas entre as diferentes equipes envolvidas, promovendo uma comunicação mais clara e colaborativa no desenvolvimento dos testes e a integração de ferramentas específicas. O Cypress foi utilizado para a execução dos testes automatizados, enquanto o Cucumber contribuiu para a definição dos testes em uma linguagem natural, de acordo com a metodologia BDD. Essa combinação permitiu a criação de um ambiente de testes robusto, alinhado com as melhores práticas de automação e colaborativo, otimizando as etapas de verificação do sistema.

Comparando os dois estudos, fica evidente que ambos os métodos, manual e automatizado, têm seus méritos, mas a automação se destaca em termos de economia de tempo e eficiência, especialmente com o suporte de metodologias como o BDD. No entanto, o primeiro estudo salienta a importância da qualidade e estabilidade na automação, sugerindo que a automação por si só não resolve todos os problemas de usabilidade e satisfação do cliente. A automação precisa ser bem planejada e implementada para trazer os resultados desejados.

Por outro lado, o segundo estudo mostra como a automação pode otimizar o ciclo de desenvolvimento, melhorando a experiência do usuário e acelerando o processo de entrega, desde que mantenha a precisão e consistência dos resultados. A comparação sugere que, embora os testes manuais ainda tenham seu lugar, especialmente em cenários complexos e específicos, a automação oferece uma clara vantagem em termos de eficiência e feedback contínuo.

Após a leitura dos estudos, foi desenvolvido o Quadro 1 a seguir, comparando os estudos de caso que abordam desafios e soluções nos testes de usabilidade em projetos de software. O primeiro foca na estabilidade e cobertura dos testes automatizados, enquanto o segundo destaca a eficiência e os ganhos de consistência proporcionados pela automação com BDD. Essa comparação resume os principais pontos de cada estudo, ressaltando os impactos na qualidade do produto e na satisfação do cliente.

Quadro 1: Comparação entre estudos de caso

Aspectos	Estudo de Caso 1: Testes de software como garantia de qualidade e eficiência: estudo de caso com uso do selenium	Estudo de Caso 2: Proposta para implantação de automação de testes de software usando behavior driven development (BDD) - Estudo de caso
Objetivo	Analisar a qualidade e estabilidade nos testes automatizados usando Selenium.	Comparar testes manuais e automatizados com ênfase na automação usando BDD.

Contexto	Simulação de teste de um formulário de registro	Teste de um sistema de precificação de produtos
Principais Problemas	Instabilidade, baixa cobertura de testes e lentidão.	Tempo inicial elevado para configuração da automação, mas ganho de eficiência após a implementação.
Ferramentas Utilizadas	Selenium para testes automatizados.	Cypress para execução de testes e Cucumber para definição de testes em linguagem natural com BDD.
Soluções Implementadas	Ampliação da cobertura de testes, paralelização de testes para ganho de velocidade.	Uso de BDD para linguagem de testes mais acessível, scripts Gherkin para maior colaboração entre equipes.
Benefícios Identificados	Melhoria na estabilidade e confiança nos resultados dos testes.	Economia de tempo (de 10 horas em testes manuais para 5 horas e 1 minuto em testes automatizados), consistência e precisão nos resultados, retestes rápidos (33 segundos).
Desafios Enfrentados	Falhas intermitentes sem causa aparente, falta de cobertura ampla, baixa velocidade de execução.	Esforço inicial na criação de scripts de automação e integração de BDD.
Impacto na Satisfação do Cliente	Melhoria na experiência do usuário ao reduzir problemas de qualidade e falhas nos testes.	Redução no tempo de entrega, maior rapidez no ciclo de correção de bugs, e melhor experiência final para o usuário devido à cobertura e consistência dos testes.
Comparação de Tempo	Tempo de execução otimizado pela paralelização, mas ainda lento devido a problemas de sincronização.	Automação com BDD reduziu o tempo de execução significativamente, e retestes passaram a ser executados em apenas 33 segundos.
Conclusão	Qualidade e estabilidade dos testes automatizados são essenciais, e a automação precisa ser bem planejada para ser eficaz.	Automação com BDD traz eficiência e economia de tempo, melhorando o ciclo de desenvolvimento e a experiência do usuário.
Vantagem da Automação	Contribui para a estabilidade, mas enfrenta desafios de implementação e manutenção.	Clara vantagem em termos de economia de tempo e consistência, especialmente com o suporte do BDD.

Fonte: autoria própria.

Essa análise comparativa valida a importância de investir em processos de teste eficientes, e os resultados mostram que a automação, quando bem executada, é uma ferramenta poderosa para elevar a qualidade e a satisfação do cliente em projetos de desenvolvimento de software.

5. Conclusão

Esta pesquisa foi fundamental para sustentar o objetivo principal deste artigo, destacando que a ausência de testes de usabilidade pode impactar o projeto de desenvolvimento de software de diversas maneiras. Entre os impactos, estão o aumento de bugs que afetam diretamente os usuários e tornam a interface confusa, dificultando uma navegação fluida e intuitiva. Esse cenário provoca desinteresse, levando ao abandono da aplicação e à perda de oportunidades de negócios, já que usuários insatisfeitos reduzem as chances de retenção e fidelização.

Além disso, a falta de testes de usabilidade pode gerar custos elevados de correção. Um exemplo apresentado no primeiro estudo de caso mostra que a criação de scripts de teste demanda um tempo significativo. Quando realizada em estágios mais avançados do desenvolvimento, essa atividade pode consumir ainda mais tempo e aumentar substancialmente os custos de produção. Realizar o planejamento antecipado dos testes, permitindo que os desenvolvedores tenham feedbacks dos usuários e apliquem as funcionalidades desde o início, definir dos objetivos, identificando os problemas de navegação, avaliando a efetividade das tarefas, ou a satisfação dos usuários, obtendo assim um direcionamento maior dos testes, são práticas de teste de usabilidade que quando aplicadas de maneira eficaz fazem com que o cliente obtenha uma satisfação maior.

Esta pesquisa destaca que ferramentas com linguagens mais claras, como Cypress, Cucumber, tornam os testes mais rápidos e eficientes, além de contribuir para a comunicação da equipe com pessoas que não tem habilidade técnica necessária, o que é muito importante já que a comunicação é um fator determinante para a satisfação do cliente. Enquanto o Selenium por ser uma ferramenta mais versátil e flexível se torna mais complexa, não tão rápida e precisa ser utilizada por desenvolvedores. O uso da automação com o BDD traz economia de tempo e eficiência, melhorando o ciclo de desenvolvimento e a experiência do usuário.

Uma das limitações da pesquisa foi a falta de dados numéricos relacionados à satisfação dos clientes e como ela é influenciada pela eficácia dos testes de usabilidade. Desse modo restringindo a possibilidade realizar análises estatísticas que poderiam ter enriquecido os resultados. De todo modo foi feita uma abordagem baseada na literatura recente, a qual forneceu *insights* valiosos.

Essa pesquisa contribui para a sociedade, destacando a importância dos testes de usabilidade para a satisfação do cliente, indicando que a aplicação de metodologias como o BDD pode otimizar a qualidade dos softwares, mas também reduzir custos e o tempo de desenvolvimento, o que resulta em soluções mais rápidas e acessíveis ao público, proporcionando uma experiência de uso aprimorada, essencial em um mercado digital cada vez mais competitivo. Este trabalho também colabora para a Academia apresentando os benefícios e desafios na implementação de testes de usabilidade de forma eficaz, enriquecendo ainda mais as pesquisas que existem sobre o assunto.

Para trabalhos futuros, sugere-se a realização de estudos que integrem abordagens quantitativas para medir com maior precisão os impactos da automação nos testes de usabilidade. Tais estudos podem contribuir para aumentar a satisfação do cliente em

relação aos softwares desenvolvidos, além de aprimorar o desempenho e a eficiência das equipes de desenvolvimento.

6. Referências bibliográficas

AIYEGBUSI, O. L. Key methodological considerations for usability testing of electronic patient-reported outcome (ePRO) systems. *Quality of Life Research*, v. 29, p. 325-333, 2020. Disponível em: <https://link.springer.com/article/10.1007/s11136-019-02329-z>. Acesso em: 02 de outubro de 2024.

BARBOZA, Hionara Nascimento; LIMA, Marcillyo Carneiro de; FERREIRA, Rubens Jonatha dos Santos; ROSA, Marine Raquel Diniz da; ARAÚJO, Ana Loísa de Lima e Silva; ACOLY, Angélica de Souza Galdino. Teste de usabilidade do aplicativo Avazum. Disponível em: <https://www.scielo.br/j/codas/a/xwDXkHbhwwhHBfbhMhpFwbC/?lang=pt>. 02 de outubro de 2024.

BARNUN, C. M. *Usability testing essentials: ready, set... test!* Burlington (MA): Morgan Kaufmann, 2020. Acesso em: 18 set. 2024.

BLASCKE, Francis Fumagalli Megda; FARINA, Renata Mirella; FLORIAN, Fabiana. Testes de software como garantia de qualidade e eficiência: estudo de caso com uso do selenium. *Revista Científica Multidisciplinar Núcleo do Conhecimento*, ano 08, ed. 09, vol. 03, pp. 26-55, setembro de 2023. ISSN: 2448-0959. Disponível em: <https://www.nucleodoconhecimento.com.br/engenharia-da-computacao/testes-de-software>. Acesso em: 17 set.2024

CAMPOS, Ana Gabriela de Abreu. Proposta para implantação de automação de testes de software usando behavior driven development (BDD) - Estudo de caso. 2024. 67 f. Trabalho de Conclusão de Curso (Sistemas de Informação) - Universidade Federal de Uberlândia, Uberlândia, 2024. Disponível em: <https://repositorio.ufu.br/handle/123456789/41987>. Acesso em: 17 set. 2024.

CASTRO, Ana Paula Ferreira Barroso de; HERCULINO, Gabriel Cardoso dos Santos; MENDONÇA, Victor Hugo Bustamonte. Padronização e reciclagem de códigos em um Marketplace utilizando Gherkin: um estudo de caso na Agro2Business com testes automatizados RSpec, Capybara e Page Objects. 2023. Disponível em: <https://repositorio.ufms.br/handle/123456789/5967>. Acesso em: 15 nov. 2024.

DA SILVA FERREIRA, Ana Carolina et al. SELENIUM, ROBOT E CYPRESS: UM ESTUDO COMPARATIVO ENTRE FERRAMENTAS DE AUTOMAÇÃO DE TESTE. Disponível em: https://multiversa.edu.br/docs/revista-cientifica/ARTIGO%205%20-%20Estudo%20Comparativo_testes_ANA_CAROLINA_DARIELSON.pdf. Acesso em: 14 nov. 2024.

LUNA, Rafael Bezerra da Silva. Usabilidade em e-commerce: uma análise no aplicativo da Dafiti. 2024. Disponível em: <https://repositorio.ufpe.br/handle/123456789/58316>. Acesso em: 09 de nov. 2024.

MWAURA, Waweru. End-to-End Web Testing with Cypress: Explore techniques for automated frontend web testing with Cypress and JavaScript. Packt Publishing Ltd, 2021.

NICCHIO, Fabiana; LEAL, Lucas; MARTINS, Eliane. Aplicação de técnicas de testes de interface com BDD nos testes de API. 2021. Disponível em: <https://www.ic.unicamp.br/~reltech/PFG/2021/PFG-21-20.pdf>. Acesso em: 24 out. 2024.

NIELSEN, J. 10 Usability Heuristics for User Interface Design. Nielsen Norman Group, 2020. Disponível em: <https://www.nngroup.com/articles/ten-usability-heuristics/>. Acesso em: 25 de setembro de 2024.

PRESSMAN, Roger S. Engenharia de Software: Uma Abordagem Profissional. 8. ed. Porto Alegre: AMGH, 2019. Acesso em: 13 set. 2024.

RAHARJANA, I. K.; HARRIS, F.; JUSTITIA, A. Tool for Generating Behavior-Driven Development Test-Cases. Journal of Information Systems Engineering and Business Intelligence, [S. l.], v. 6, n. 1, p. 27–36, 2020. DOI: 10.20473/jisebi.6.1.27-36. Disponível em: <https://e-journal.unair.ac.id/JISEBI/article/view/18082>. Acesso em: 14 nov. 2024.

SANTOS, Shexmo; SOARES, Michel dos Santos; ROCHA, Fabio Gomes. Perception of Professionals Regarding the Adoption of Behavior-Driven Development (Bdd): A Descriptive and Statistical Study Through a Survey. Disponível em: <https://dx.doi.org/10.2139/ssrn.4933832>. Acesso em: 14 nov. 2024.

SILVESTRE, B. H.; SCOMBATTI PINTO, G. A importância dos testes de software. Revista Interface Tecnológica, v. 20, n. 2, p. 122-133, 2023. DOI: 10.31510/inf.v20i2.1775. Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/1775>. Acesso em: 17 set. 2024.