



A IMPORTÂNCIA ESTRATÉGICA DOS TESTES DE SOFTWARE NO CICLO DE DESENVOLVIMENTO

Tiago F. MORAES¹; **Paulo C. dos SANTOS²**

RESUMO

Este artigo apresenta uma revisão bibliográfica sobre a importância estratégica dos testes de software, reposicionando-os de um centro de custo para um pilar de criação de valor e mitigação de risco. A análise demonstra que a negligência com os testes resulta em impactos financeiros severos, quantificados pelo Custo da Não Qualidade (CoNQ) e pela Dívida Técnica. Evidencia-se que um processo de teste maduro, integrado a metodologias ágeis e alavancado por automação, não apenas protege a reputação e a segurança da organização, mas também otimiza a entrega de valor e garante a satisfação do usuário. A pesquisa consolida o argumento de que o investimento em qualidade de software gera um retorno sobre o investimento (ROI) mensurável, sendo indispensável para a sustentabilidade de negócios na economia digital. A conclusão aponta para uma cultura de qualidade como diferencial competitivo.

Palavras-chave:

Custo da Não Qualidade; Dívida Técnica; Gerenciamento de Risco.

1. INTRODUÇÃO

A sociedade contemporânea opera sobre uma infraestrutura digital cada vez mais complexa e onipresente. Desde sistemas financeiros e de saúde até serviços de energia e transporte, o software tornou-se o alicerce de operações críticas e da vida cotidiana (BARBOSA *et al.*, 2004; BASTOS *et al.*, 2006). Essa dependência crescente eleva exponencialmente os riscos associados a falhas, tornando a qualidade e a confiabilidade do software não apenas um requisito técnico, mas um imperativo estratégico.

Nesse contexto, a disciplina de testes de software emerge como um processo fundamental para a gestão de riscos, criação de valor e garantia da continuidade dos negócios.

A questão problema que norteia esta investigação é: em um cenário de crescente complexidade e dependência de sistemas de software, como a negligência ou a aplicação inadequada de processos de teste impacta a viabilidade financeira, a segurança e a reputação das organizações? A percepção do teste como um custo, em vez de um investimento estratégico, gera riscos sistêmicos que ameaçam não apenas projetos individuais, mas a própria sustentabilidade do negócio (BURNSTEIN; SUWANNASART; CARLSON, 1999; PRESSMAN, 2016).

O objetivo deste artigo é analisar e demonstrar a importância estratégica dos testes de

¹Discente do Bacharelado em Ciência da Computação, IFSULDEMINAS – Campus Muzambinho. E-mail: tiago.fernandes@alunos.ifsuldeminas.edu.br.

²Docente do Bacharelado em Ciência da Computação, IFSULDEMINAS – Campus Muzambinho. E-mail: paulo.santos@muz.ifsuldeminas.edu.br.

software como um pilar fundamental do desenvolvimento. Por meio de uma revisão bibliográfica e da análise de estudos de caso notórios, busca-se evidenciar a correlação direta entre a maturidade do processo de teste e a mitigação de riscos operacionais e financeiros.

2. FUNDAMENTAÇÃO TEÓRICA

O teste de software é o processo investigativo de executar um programa para encontrar erros (PRESSMAN, 2016; BARBOSA *et al.*, 2004), garantindo que ele atenda às necessidades dos usuários (SOMMERVILLE, 2019).

A qualidade de software possui implicações financeiras diretas, mensuradas pelo Custo da Não Qualidade (CoNQ) — custos de falhas internas e externas — e pela Dívida Técnica, uma metáfora para o retrabalho futuro gerado por soluções apressadas (PRESSMAN, 2016; SOMMERVILLE, 2019). A Dívida Técnica é a causa; o CoNQ, o efeito.

Padrões como a ISO/IEC 29119 e metodologias ágeis transformaram o processo de teste em uma atividade contínua (BASTOS *et al.*, 2006; PRESSMAN, 2016). A automação, com ferramentas como Selenium, permite feedback rápido e sustentação de qualidade em ciclos curtos, consolidando sua importância estratégica (BURNSTEIN; SUWANNASART; CARLSON, 1999).

3. MATERIAL E MÉTODOS

Este estudo caracteriza-se como uma revisão bibliográfica descritiva e qualitativa, centrada na análise de literatura científica e técnica sobre testes de software e seu papel estratégico no ciclo de desenvolvimento. A pesquisa foi realizada nas bases Scopus, IEEE Xplore, ACM Digital Library e Google Scholar, utilizando a string: ("software testing" OR "quality assurance") AND ("return on investment" OR "ROI" OR "cost of quality"), complementada por suas versões em português para ampliar o escopo nacional.

Foram incluídos artigos revisados por pares, publicados entre 2015 e 2025, em inglês ou português, com aderência direta à temática. Excluíram-se publicações não científicas, literatura cinzenta e documentos anteriores a 2015. Após a triagem inicial de 42 documentos, 24 foram selecionados para análise aprofundada.

A etapa de síntese temática estruturou os achados em quatro eixos analíticos: imperativos econômicos (CoNQ, Dívida Técnica e ROI), impactos nos stakeholders (segurança e experiência do usuário), processos e tecnologias (metodologias ágeis e automação), e evidências empíricas (falhas de alto impacto). Essa abordagem permitiu conectar teoria e prática, consolidando uma argumentação robusta ao longo do artigo.

4. RESULTADOS E DISCUSSÃO

Do ponto de vista econômico, a negligência em testes de software gera impactos financeiros significativos. Estudos demonstram que um defeito não detectado nas fases iniciais do desenvolvimento pode custar até 100 vezes mais quando identificado em produção, devido ao retrabalho, perda de clientes e riscos legais (SILVA; MÜLLER; BERNARDI, 2011; PRESSMAN, 2016). A ausência de uma abordagem estruturada de testes contribui diretamente para o acúmulo de Dívida Técnica, que, quando não controlada, transforma-se em Custo da Não Qualidade (CoNQ). Esse custo se manifesta por meio de penalidades contratuais, degradação da imagem da empresa e redução da competitividade.

O impacto da qualidade de software também se estende diretamente aos stakeholders, especialmente aos usuários finais. A experiência do usuário (UX) é afetada não apenas por falhas funcionais, mas também por problemas de segurança e usabilidade. A exposição de dados sensíveis, por exemplo, compromete a confiança no produto e pode acarretar severas consequências jurídicas e reputacionais. Para mitigar tais riscos, práticas como o *Static Application Security Testing* (SAST) e o *Dynamic Application Security Testing* (DAST) são utilizadas com o objetivo de identificar vulnerabilidades ainda durante o desenvolvimento. Complementarmente, o teste de aceitação pelo usuário (UAT) assegura que o software entregue esteja alinhado às expectativas e às necessidades reais do público-alvo.

No que se refere aos processos e tecnologias, observa-se que a maturidade em testes está diretamente relacionada à adoção de modelos estruturados e ao uso estratégico da automação. Modelos como o TMMi (*Test Maturity Model Integration*) oferecem uma trilha evolutiva para a melhoria contínua do processo de testes. A automação, por sua vez, desempenha um papel essencial na garantia de testes de regressão consistentes e na redução do tempo dos ciclos de desenvolvimento em ambientes ágeis. A utilização de métricas como a densidade de defeitos, a taxa de cobertura de testes e o tempo médio de correção contribui para a gestão baseada em dados, fortalecendo a tomada de decisão e promovendo a previsibilidade dos projetos.

Por fim, evidências empíricas extraídas de casos emblemáticos reforçam de forma contundente a tese central deste artigo. A missão Mars Climate Orbiter, da NASA, fracassou devido a um erro de conversão de unidades não detectado por testes adequados, resultando na perda total da sonda. O caso da Knight Capital Group exemplifica o custo do código obsoleto e não testado, ocasionando um prejuízo de 440 milhões de dólares em apenas 45 minutos. Já o escândalo envolvendo o sistema Horizon, do Correio britânico, levou à condenação injusta de dezenas de funcionários por erros não identificados em testes de aceitação. Esses eventos revelam o custo oculto da negligência com a qualidade e evidenciam que o teste de software não deve ser encarado como um luxo técnico, mas como uma disciplina estratégica indispensável à sustentabilidade de

qualquer solução digital.

5. CONCLUSÃO

Este estudo evidenciou que os testes de software são um fator estratégico, e não apenas operacional. Sua ausência ou subvalorização gera riscos traduzidos pelo Custo da Não Qualidade e pela Dívida Técnica. A consolidação dos testes como pilar estratégico exige ações combinadas: educação sobre impactos financeiros, uso de métricas alinhadas ao negócio e automação inteligente. O futuro aponta para a integração da Inteligência Artificial, ampliando a confiabilidade e o valor entregue pelo software.

REFERÊNCIAS

BARBOSA, E. F. *et al.* **Introdução ao Teste de Software**. Rio de Janeiro: Campus, 2004. Disponível em: <[https://www.kufunda.net/publicdocs/Introdu%C3%A7%C3%A3o%20ao%20Teste%20de%20Software%20\(Marcio%20Eduardo%20Delamaro\).pdf](https://www.kufunda.net/publicdocs/Introdu%C3%A7%C3%A3o%20ao%20Teste%20de%20Software%20(Marcio%20Eduardo%20Delamaro).pdf)>. Acesso em: 2 jul. 2025.

BASTOS, A. *et al.* **MPT.BR: Melhoria de Processo de Teste**. Rio de Janeiro: Campus, 2006. Disponível em: <https://www.mpt.org.br/mpt_br>. Acesso em: 2 jul. 2025.

BURNSTEIN, I.; SUWANNASART, T.; CARLSON, C. R. A path to test automation. **STQE Magazine**, v. 1, n. 5, p. 28-34, 1999.

PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 8. ed. Porto Alegre: AMGH, 2016. Disponível em: <<https://archive.org/details/pressman-engenharia-de-software-uma-abordagem-profissional-8a>>. Acesso em: 2 jul. 2025.

SILVA, D. R. S.; MÜLLER, F. M.; BERNARDI, G. Teste de software: estudo de caso em uma empresa de desenvolvimento de software. **Revista Eletrônica de Sistemas de Informação**, v. 10, n. 2, 2011.

SOMMERVILLE, Ian. **Engenharia de Software**. 10. ed. São Paulo: Pearson Education, 2019. Disponível em: <<https://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/softwareEngineeringSomerville.pdf>>. Acesso em: 2 jul. 2025.