

Testes de Aceitação em Engenharia de Software

Euclides Palma Paim*

*Curso Técnico em Informática

Escola de Educação Básica Elfrida Cristino da Silva
Secretaria de Estado da Educação de Santa Catarina (SED-SC)
Itajaí, SC, Brasil
euclidespaim@gmail.com

Resumo—Os testes de aceitação representam uma etapa crucial na validação de sistemas, sendo o último filtro antes da entrega ao cliente. Este trabalho apresenta os conceitos, tipos, processos e boas práticas relacionados aos testes de aceitação, destacando sua importância na garantia da qualidade de software.

Index Terms—Teste de Aceitação, Qualidade de Software, Engenharia de Software, Validação, Verificação.

I. INTRODUÇÃO

O desenvolvimento de software requer uma série de atividades de verificação e validação que visam garantir que o produto final atenda às expectativas dos usuários e às especificações estabelecidas. Nesse contexto, os testes de aceitação assumem papel fundamental, uma vez que são responsáveis por validar o sistema do ponto de vista do cliente, antes de sua liberação para uso operacional.

Os testes de aceitação estão diretamente relacionados aos requisitos do usuário e aos critérios de sucesso do projeto, sendo geralmente conduzidos pelos próprios clientes, usuários finais ou equipes independentes de qualidade. [1]

II. DEFINIÇÃO DOS TESTES DE ACEITAÇÃO

Testes de aceitação são atividades que visam verificar se o sistema atende às necessidades e expectativas dos usuários. Segundo Sommerville [2], os testes de aceitação correspondem à validação dos requisitos funcionais e não funcionais, e servem como base para a decisão de aceitar ou rejeitar o software.

De acordo com o IEEE [3], testes de aceitação são “um processo formal para verificar se um sistema satisfaz critérios de aceitação e permite que o cliente determine se aceita o sistema”.

III. OBJETIVOS DOS TESTES DE ACEITAÇÃO

Os principais objetivos dos testes de aceitação incluem:

- Confirmar que o sistema atende aos requisitos acordados;
- Garantir que o produto está pronto para ser colocado em operação;
- Reduzir riscos associados a falhas em produção;
- Assegurar que o software oferece valor de negócio esperado.

IV. TIPOS DE TESTES DE ACEITAÇÃO

A. Teste de Aceitação do Usuário (UAT)

Realizado pelos próprios usuários finais para verificar se o sistema atende às suas necessidades.

B. Teste de Aceitação Operacional

Avalia se o sistema é capaz de operar no ambiente de produção, considerando infraestrutura, backups e procedimentos operacionais. O objetivo principal é garantir que o software funcione corretamente e de forma intuitiva no ambiente real de uso, antes da sua implementação final.

C. Teste de Aceitação Contratual

Visa verificar se o software cumpre as obrigações contratuais estabelecidas entre fornecedor e cliente. Normalmente, envolve uma análise detalhada do contrato e do software em si, para garantir que todas as cláusulas e especificações técnicas sejam respeitadas.

D. Teste de Aceitação Regulatório

Garante que o sistema está em conformidade com normas, regulamentações e legislações aplicáveis. Aqui o objetivo último é prevenir que o produto final possa incorrer em formas ilegais de uso.

V. PROCESSO DE EXECUÇÃO DOS TESTES

O processo de testes de aceitação geralmente inclui as seguintes etapas:

- 1) Planejamento: Definição dos critérios, escopo e responsáveis.
- 2) Design dos Casos de Teste: Baseados nos requisitos e critérios de aceitação.
- 3) Preparação do Ambiente: Montagem do ambiente que simula o real.
- 4) Execução: Realização dos testes conforme os casos definidos.
- 5) Registro dos Resultados: Documentação dos testes executados.
- 6) Análise e Decisão: Avaliação dos resultados para aceitação ou recusa do sistema.

VI. CRITÉRIOS DE ACEITAÇÃO

Os critérios de aceitação são condições específicas, quantificáveis e verificáveis que um software deve atender para que um requisito, uma funcionalidade ou até mesmo um produto inteiro seja considerado aceito pelos stakeholders. Critérios de aceitação servem como a ponte entre os requisitos do negócio e a definição concreta de pronto (Definition of Done) no desenvolvimento de software. [4]

Critérios de aceitação funcionam como uma forma de contrato informal entre desenvolvedores e usuários, especificando de maneira clara e objetiva quais são as expectativas associadas a uma entrega de software. Eles não apenas orientam os testes de aceitação, como também ajudam na validação de requisitos e no alinhamento entre os envolvidos no projeto. [2]

Além de validar requisitos funcionais, os critérios de aceitação muitas vezes abrangem também requisitos não funcionais, como desempenho, segurança, acessibilidade e conformidade com normas. Ignorar requisitos não funcionais nos critérios de aceitação é uma das principais fontes de insatisfação dos clientes após a entrega. [5]

A. Tipos de Critérios de Aceitação

Os critérios podem assumir diferentes formatos, tais como:

- Critérios Funcionais: Validam se uma funcionalidade específica executa corretamente suas tarefas.
- Critérios Não Funcionais: Avaliam aspectos como tempo de resposta, escalabilidade, segurança e usabilidade.
- Critérios de Negócio: Estão alinhados com objetivos estratégicos, metas de negócio e regras operacionais.
- Critérios Técnicos: Relacionam-se a aspectos como compatibilidade com plataformas, integração com APIs, entre outros.

B. Exemplos de Critérios Bem Definidos

- O sistema deve permitir que um usuário cadastrado realize login com autenticação em menos de 3 segundos, 95% das vezes.
- O botão de envio de formulário deve permanecer inativo enquanto campos obrigatórios não forem preenchidos.
- O relatório financeiro deve ser gerado em PDF e conter todas as transações dos últimos 12 meses.
- O sistema deve estar em conformidade com a Lei Geral de Proteção de Dados (LGPD) no Brasil. [6]

C. Papel dos Critérios no Desenvolvimento Ágil

No contexto ágil, critérios de aceitação são frequentemente elaborados juntamente com os User Stories. Como definido em [4], “se uma história de usuário não tem critérios de aceitação, ela não está pronta para desenvolvimento”. Ferramentas como o Behavior-Driven Development (BDD) formalizam esses critérios em formato dado-quando-então (Given-When-Then), facilitando a automação e a rastreabilidade dos testes.

D. Riscos de Critérios Mal Definidos

Critérios de aceitação ambíguos, subjetivos ou ausentes são uma das principais causas de retrabalho, aumento de custos e insatisfação dos clientes. Beizer (1995) destaca que “a ausência de critérios de aceitação claros torna impossível determinar se um teste foi bem-sucedido”.

Portanto, a construção de critérios de aceitação robustos deve envolver não apenas os desenvolvedores, mas também analistas de negócio, representantes dos usuários e testadores, promovendo uma visão compartilhada dos objetivos do software.

VII. FERRAMENTAS E TÉCNICAS

Entre as ferramentas e técnicas utilizadas nos testes de aceitação destacam-se:

- Gerenciamento de testes: TestRail, Zephyr, Xray.
- Automação de testes de aceitação: Cucumber, Robot Framework, Behave.
- BDD (Behavior-Driven Development): Foco em cenários escritos em linguagem natural, facilitando a comunicação entre equipe técnica e stakeholders.

VIII. DESAFIOS E BOAS PRÁTICAS

A. Desafios

- Definir critérios de aceitação claros e objetivos.
- Alinhar expectativas entre clientes e desenvolvedores.
- Disponibilidade de usuários para participar dos testes.

B. Boas Práticas

- Envolver os usuários desde o início na definição dos testes.
- Documentar claramente os critérios e os resultados.
- Utilizar ferramentas que facilitem o rastreamento e automação dos testes.

IX. CONCLUSÃO

Os testes de aceitação são essenciais para garantir que o software atenda às necessidades dos usuários e aos requisitos de negócio. Sua aplicação correta reduz riscos, melhora a qualidade e fortalece a confiança entre clientes e fornecedores. A adoção de boas práticas, aliada a ferramentas adequadas, é fator crítico para o sucesso desta etapa.

REFERÊNCIAS

- [1] G. J. Myers, C. Sandler, and T. Badgett, *The Art of Software Testing*, 3rd ed. John Wiley & Sons, 2011.
- [2] I. Sommerville, *Engenharia de Software*, 9th ed. Pearson, 2011.
- [3] IEEE, “Ieee standard for software and system test documentation,” 2008, IEEE Std 829-2008.
- [4] L. Crispin and J. Gregory, *Agile Testing: A Practical Guide for Testers and Agile Teams*. Addison-Wesley Professional, 2009.
- [5] C. Kaner, J. Falk, and H. Q. Nguyen, *Testing Computer Software*, 2nd ed. John Wiley & Sons, 1999.
- [6] Presidência da República do Brasil, “Lei nº 13.709, de 14 de agosto de 2018. lei geral de proteção de dados pessoais (lgpd),” https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm, 2018, diário Oficial da União, 15 ago. 2018.