



Material de Apoio para Estudo

Quadrantes de Teste

Versão 1.1

Direitos Autorais

Copyright© Brazilian Software Testing Qualifications Board (doravante denominado BSTQB®)

BSTQB® é uma marca registrada da ABRAMTI Associação Brasileira de Melhoria em Ti.

BSTQB® é o Conselho Membro do ISTQB® International Software Testing Qualifications Board representando o Brasil nesta instituição.

Copyright©2023 autores da versão inicial (ordem alfabética): Irene Nagase, George Fialkovitz, Osmar Higashi e Stênio Viveiros.

Todos os direitos reservados. Os autores transferem os direitos autorais para o Brazilian Software Testing Qualifications Board (BSTQB®). Os autores (como detentores atuais de direitos autorais) e o BSTQB® (como futuro detentor dos direitos autorais) concordaram com as seguintes condições de uso:

- Este material foi produzido para apoiar o estudo do candidato interessado nos exames de certificação em Teste de Software do ISTQB®.
- Este material não pode ser comercializado.
- Extratos deste documento podem ser copiados se a fonte for reconhecida.
- Qualquer indivíduo ou grupo de indivíduos pode usar este material como base para artigos e livros, se os autores e o BSTQB® forem reconhecidos como a fonte e os proprietários dos direitos autorais

Histórico

Versão	Data	Observação
0.0	02/08/2023	Versão inicial
0.1	22/08/2023	Correções e implementações diversas
1.0	30/08/2023	Lançamento
1.1	29/01/2024	Correção nos tipos de teste em Q3.

Sumário

Direitos Autorais	2
Histórico	3
Sumário	3
1 Introdução.....	4
2 Quadrantes de Teste.....	5
2.1 Quadrante de Teste 1 (Q1).....	6
2.1.1 Principais características	6
2.1.2 Níveis e Tipos de Teste	6
2.1.3 Benefícios.....	6
2.2 Quadrante de Teste 2 (Q2).....	7
2.2.1 Principais características	7
2.2.2 Níveis e Tipos de Teste	7
2.2.3 Benefícios.....	7
2.3 Quadrante de Teste 3 (Q3).....	8
2.3.1 Principais características	8
2.3.2 Níveis e Tipos de Teste	8
2.3.3 Benefícios.....	8
2.4 Quadrante de Teste 4 (Q4).....	9
2.4.1 Principais características	9
2.4.2 Níveis e Tipos de Teste	9
2.4.3 Benefícios.....	10
3 Recapitulando.....	11
4 Referências.....	12

1 Introdução

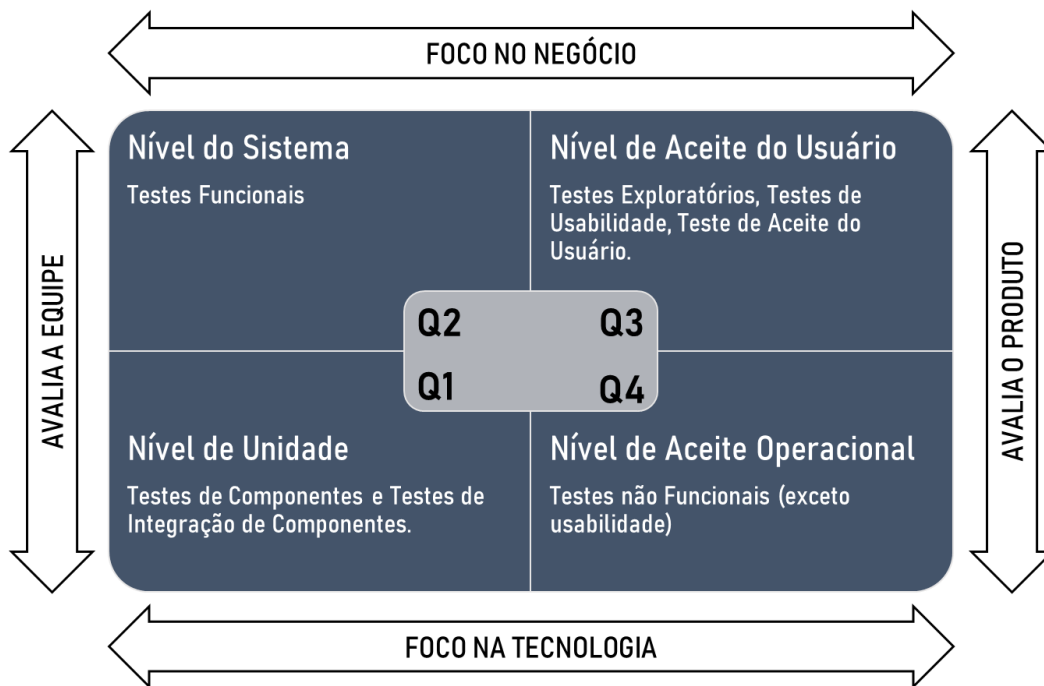
O desenvolvimento de software é um processo complexo e desafiador que exige garantias sólidas de que o produto final atenderá aos requisitos, será confiável e proporcionará uma excelente experiência aos usuários. Nesse contexto, o teste de software desempenha um papel crucial para identificar e corrigir defeitos, garantir a qualidade e mitigar riscos associados ao lançamento de um novo produto ou atualização.

Uma abordagem estruturada para o planejamento e execução dos testes é essencial para que sejam abordados os diversos aspectos do sistema em desenvolvimento. O Quadrante de Teste, proposto por Brian Marick, é uma técnica amplamente utilizada na indústria de desenvolvimento de software para categorizar e organizar os diferentes tipos de testes em quatro quadrantes distintos, garantindo a cobertura adequada em todas as dimensões do projeto de software.

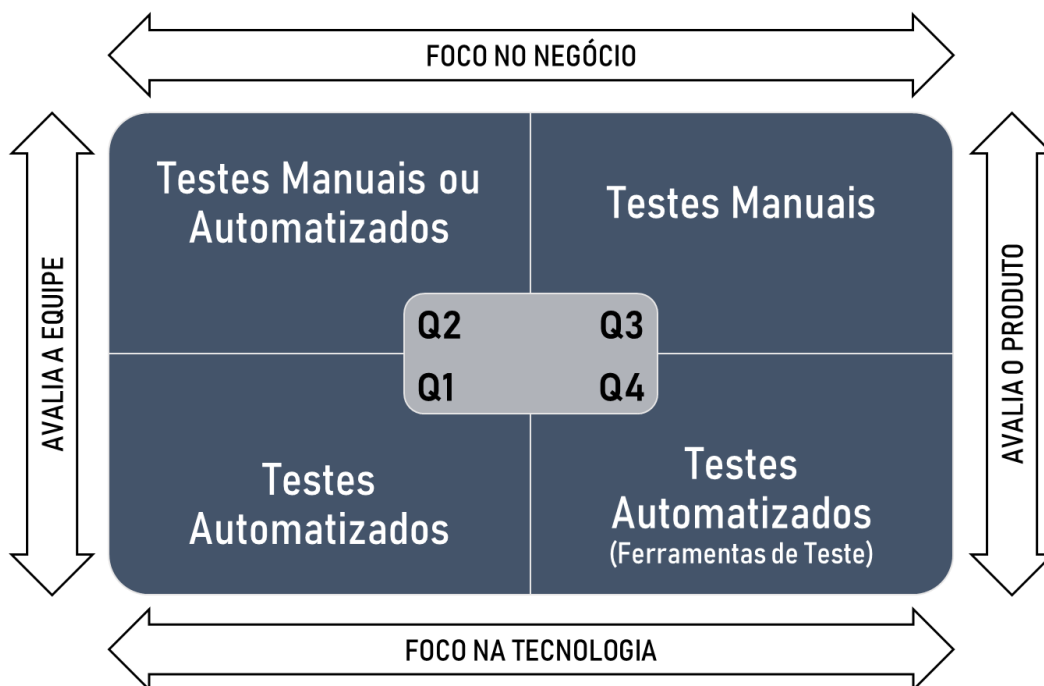
Nesta abordagem, os testes são organizados em quadrantes relacionando os níveis de teste com os tipos de teste, as atividades, as técnicas de teste e os produtos de trabalho apropriados no desenvolvimento ágil de software. Esses critérios fornecem uma estrutura clara para identificar os objetivos e escopos de cada teste, permitindo que as equipes de desenvolvimento e teste atuem de forma colaborativa e eficiente.

2 Quadrantes de Teste

Nos quadrantes de teste, os testes podem focar no negócio ou em uma tecnologia. Alguns testes avaliam o trabalho realizado pela equipe ágil e confirmam o comportamento do software, outros testes podem avaliar o produto.



Os testes podem ser totalmente manuais, totalmente automatizados, uma combinação de ambos, ou ainda, apoiados por ferramental específico.



2.1 Quadrante de Teste 1 (Q1)

O Q1 representa o grupo de testes que tem como foco a execução de testes que verificam a tecnologia do software em desenvolvimento. Em outras palavras, esses testes são realizados para validar o comportamento do código e das funcionalidades do sistema em nível técnico.

2.1.1 Principais características

Os testes neste quadrante são voltados para avaliar aspectos específicos da tecnologia utilizada no desenvolvimento do software, como unidades de código, componentes e integrações entre eles. É uma categoria fundamental para garantir que o código funcione conforme o esperado em nível mais atômico.

Geralmente é executado pelos próprios desenvolvedores do software, pois eles estão mais familiarizados com o código e têm o conhecimento necessário para criar e executar testes unitários, de componentes e de integração. Esses testes são essenciais para garantir que o código funcione corretamente em nível de detalhe e que as partes individuais do sistema interajam de maneira adequada.

É comum encontrar testes automatizados neste quadrante, especialmente testes unitários, que são escritos pelos desenvolvedores para verificar a funcionalidade das unidades de código de forma rápida e repetitiva. A automação ajuda a garantir uma execução consistente dos testes sempre que há alterações no código, possibilitando a identificação precoce de defeitos. Todos os testes automatizados devem ser incluídos na CI (Integração Contínua).

Com o avanço da inteligência artificial o desenvolvimento de testes de unidade pode ser mais eficiente e mais efetivo. Por exemplo, o *Copilot* pode ajudar a gerar código de testes de unidade com base nas funcionalidades que estão sendo implementadas.

2.1.2 Níveis e Tipos de Teste

Teste de Componente: verifica as menores unidades de código, como funções, métodos ou classes, isoladamente.

Teste de Integração de Componente: avalia a interação entre diferentes componentes do sistema.

Teste de Integração de Sistemas: verifica a integração de diferentes módulos ou sistemas para garantir que funcionem juntos corretamente.

2.1.3 Benefícios

O Q1 permite detectar e corrigir defeitos no código em estágios iniciais do desenvolvimento, facilitando a manutenção e evolução do software. Além disso, a automação dos testes nessa categoria ajuda a obter feedback rápido sobre a qualidade do código, contribuindo para o desenvolvimento ágil e iterativo do projeto.

É importante ressaltar que o Q1 não é suficiente para garantir a qualidade geral do software. Ele aborda principalmente questões técnicas e de baixo nível. Para uma abordagem completa, é necessário combinar os testes do Q1 com os outros quadrantes, especialmente o Q2, que trata dos testes relacionados aos requisitos de negócio. Juntos, esses quadrantes fornecem uma estratégia abrangente de testes que cobre todos os aspectos importantes do software em desenvolvimento.

2.2 Quadrante de Teste 2 (Q2)

O Q2 representa a categoria de testes que foca na execução de testes relacionados aos aspectos de negócio do software em desenvolvimento. Em outras palavras, esses testes têm como objetivo validar se o sistema está atendendo aos requisitos de negócio e entregando o valor desejado aos usuários e stakeholders.

2.2.1 Principais características

O Q2 aborda os testes que se concentram em validar a funcionalidade do software em relação aos requisitos de negócio. É essencial para garantir que o produto final atenda às expectativas dos usuários e cumpra os objetivos do negócio.

Embora desenvolvedores possam contribuir com testes no Q2, a execução é frequentemente conduzida por Testadores ou Analistas de Teste. Esses profissionais possuem habilidades específicas para validar o software em relação aos requisitos estabelecidos.

No Q2 você pode encontrar uma combinação de testes manuais e automatizados. Os testes manuais são úteis para explorar cenários complexos e para verificar a experiência do usuário, enquanto a automação pode ser aplicada para garantir a consistência e eficiência dos testes.

Nesse quadrante, é muito comum ver a automação de testes de UI utilizando ferramentas como Selenium, Cypress, Robot Framework dentre outras. As novas ferramentas de automação que utilizam a IA com visão computacional proporcionam que os scripts sejam gerados por pessoas que conheçam muito bem o negócio, mas que não possuem conhecimento de linguagem de programação.

2.2.2 Níveis e Tipos de Teste

Teste Funcional: valida as funcionalidades do sistema conforme especificado nos requisitos, confirmando se o sistema atende aos critérios de aceite definidos pelos stakeholders.

2.2.3 Benefícios

O objetivo principal do Q2 é garantir que o software entregue valor de negócio aos usuários. Isso significa que os testes são direcionados para garantir que o software esteja cumprindo suas finalidades e proporcionando benefícios reais.

O Q2 é essencial para garantir a qualidade do software de acordo com as expectativas dos usuários e stakeholders. Ao combiná-lo com os outros quadrantes de teste, a equipe de

desenvolvimento pode obter uma abordagem mais completa para garantir a qualidade do software, abrangendo desde a validação técnica até a conformidade com os requisitos de negócio e a entrega de valor para os usuários.

2.3 Quadrante de Teste 3 (Q3)

Este quadrante concentra-se em validar os aspectos não funcionais do software e aborda preocupações específicas de valor para os usuários e stakeholders, como performance, segurança, escalabilidade e outros atributos críticos para o sucesso do sistema.

2.3.1 Principais características

O Q3 aborda testes que exploram e validam riscos, preocupações éticas e valores importantes para os usuários e stakeholders.

Nesse quadrante, é comum o time de testes acompanhar e apoiar o time de PO ou de usuários finais na validação das histórias de usuário que estão sendo entregues na sprint ou na release do software. Esses testes podem ser realizados no ambiente pré-produção criada pela empresa desenvolvedora do sistema ou até mesmo no ambiente de produção do cliente final.

2.3.2 Níveis e Tipos de Teste

Teste de Usabilidade: avalia a facilidade de uso e a experiência do usuário com o software.

Teste de Aceite: neste quadrante é validado o aceite do usuário voltado para o negócio, através de testes que avaliam o produto, utilizando cenários e dados realistas. O teste mais comum executado neste quadrante é o Teste de Aceite do Usuário (UAT), Teste de Aceite Contratual e Normativo, Teste Alfa e Teste Beta.

Teste de Sistema: Teste funcional de tarefas de ponta a ponta.

2.3.3 Benefícios

O Q3 é crucial para garantir que o software não apenas funcione corretamente, mas também atenda a todos os critérios de aceite baseados nos requisitos críticos. Ignorar esses testes pode levar a problemas graves no produto final e afetar negativamente a experiência do usuário.

Neste quadrante, os usuários finais do sistema participam ativamente dos testes para garantir que o sistema atenda às suas necessidades e expectativas. O foco é validar a usabilidade, a funcionalidade e a experiência geral do usuário através de Testes de Aceite do Usuário (UAT).

O Q3 é crucial para garantir que o sistema esteja pronto para ser implantado em um ambiente de produção ou entregue ao cliente. É importante notar que, mesmo que os requisitos e resultados esperados sejam conhecidos, podem ocorrer problemas ou discrepâncias durante o teste de aceitação, levando a ajustes finais antes da implantação final.

Ao combinar os testes do Q3 com os outros tipos de testes (Q1, Q2 e Q4), a equipe pode obter uma visão abrangente e equilibrada da qualidade do software, abordando tanto a

funcionalidade quanto os aspectos não funcionais que são essenciais para o sucesso e a satisfação dos usuários.

2.4 Quadrante de Teste 4 (Q4)

O Q4 é voltado para a tecnologia e contém testes que visam avaliar o produto. Também busca melhorar o processo de teste e a qualidade do produto. Este quadrante contém *smoke tests* e testes não funcionais (exceto testes de usabilidade), que são geralmente automatizados, com o uso efetivo de ferramentas específicas para teste.

2.4.1 Principais características

Este quadrante aborda testes que envolvem a equipe de desenvolvimento trabalhando para melhorar o processo de teste, explorando novas abordagens e técnicas e identificando formas de aumentar a eficiência do processo. Embora o foco esteja em aspectos técnicos, a equipe também busca agregar valor ao produto.

O objetivo principal é a melhoria contínua do processo de teste e, por extensão, da qualidade do produto. A equipe procura identificar e adotar melhores práticas, ferramentas e abordagens para otimizar a eficácia dos esforços de teste.

A maioria dos testes são automatizados no quadrante Q4 e por isso, existe um grande investimento de ferramentas utilizando IA. Por exemplo, ferramentas para realizar testes de segurança (*pentest*), ferramentas para executar testes de performance bem como ferramentas para automação de testes de UI e API.

2.4.2 Níveis e Tipos de Teste

Teste de Aceite: Teste de Aceite Operacional,

Teste Não Funcional (exceto Teste de Usabilidade): avalia os atributos que não sejam características funcionais de um componente ou sistema. O principal objetivo do Q4 é verificar as características não funcionais da qualidade do software, que são igualmente críticos para o sucesso do projeto.

Testes de Carga: verifica a capacidade do sistema de lidar com uma grande quantidade de usuários ou transações.

Testes de Segurança: avalia a resistência do sistema a ataques e protegem os dados contra acesso não autorizado.

Testes de Performance: mede a performance e a velocidade do sistema em diferentes condições de carga.

2.4.3 Benefícios

O Q4 encoraja a inovação e o aprendizado contínuo. A equipe busca aprender com as experiências passadas e busca aprimorar suas habilidades, abordagens e conhecimentos sobre testes de software.

É importante compreender que o Q4 complementa as outras categorias de testes (Q1, Q2 e Q3). Ao combinar todas as quatro categorias, a equipe de desenvolvimento pode obter uma visão abrangente do processo de teste, cobrindo a validação técnica, a conformidade com os requisitos de negócio, a verificação dos atributos não funcionais e a busca constante por melhorias e inovações no processo de teste.

Lembre-se de que a ênfase e os detalhes das categorias de teste podem variar dependendo do contexto do projeto e das práticas adotadas pela equipe. O Quadrante de Testes é uma ferramenta útil para organizar e planejar os esforços de teste de forma equilibrada e eficiente.

3 Recapitulando...

O Quadrante de Teste é uma abordagem proposta por Brian Marick para categorizar e organizar os diferentes tipos de testes em projetos de software. Ele consiste em quatro quadrantes distintos, cada um com um propósito específico:

Quadrante Q1 (voltado para a tecnologia, avalia a equipe). Esse quadrante contém testes de componentes e de integração de componentes. Esses testes devem ser automatizados e incluídos no processo de CI.

Quadrante Q2 (voltado para os negócios, avalia a equipe). Esse quadrante contém testes funcionais, exemplos, testes de histórias de usuários, protótipos de experiência do usuário, testes de API e simulações. Esses testes verificam os critérios de aceite e podem ser manuais ou automatizados.

Quadrante Q3 (voltado para os negócios, avalia o produto). Esse quadrante contém testes exploratórios, testes de usabilidade e testes de aceite do usuário. Esses testes são orientados para o usuário e geralmente manuais.

Quadrante Q4 (voltado para a tecnologia, avalia o produto). Esse quadrante contém *smoke tests* e testes não funcionais (exceto testes de usabilidade). Esses testes geralmente são automatizados.

O Quadrante de Teste é uma ferramenta útil para ajudar as equipes de desenvolvimento e teste a entenderem e planejarem seus esforços de teste de forma abrangente, garantindo que todos os aspectos relevantes do sistema sejam devidamente abordados. Ao combinar as técnicas de teste em diferentes quadrantes, a equipe pode obter uma visão completa da qualidade do software e identificar áreas de melhoria para entregar um produto de alta qualidade e valor para os usuários finais.

4 Referências

Marick, B. (2003)

Exploration through Example, <http://www.exampler.com/oldblog/2003/08/21.1.html#agile-testing-project-1>

Crispin, L. and Gregory, J. (2008)

Agile Testing: A Practical Guide for Testers and Agile Teams, Pearson Education: Boston MA

Whittaker, J. (2009)

Exploratory Software Testing: Tips, Tricks, Tours, and Techniques to Guide Test Design, Addison Wesley

ISTQB Syllabus CTFL

Certified Tester Foundation Level, v4.0, bstqb.org.br

ISTQB Syllabus CTFL-AT

Certified Tester Foundation Level, Agile Tester, v1.0, bstqb.org.br