



Níveis de teste



Revisão

Qualidade de *software*

“Qualidade de software é a conformidade com requisitos funcionais e de desempenho explicitamente declarados, normas de desenvolvimento explicitamente documentadas e características implícitas que são esperadas em todo software desenvolvido profissionalmente”

Pressman (PRESSMAN, 2006)



Revisão

Qualidade de *software*

“Qualidade de software é a conformidade com requisitos funcionais e de desempenho explicitamente declarados, normas de desenvolvimento explicitamente documentadas e características implícitas que são esperadas em todo software desenvolvido profissionalmente”

Pressman (PRESSMAN, 2006)



Revisão

Teste de software

“Teste é a simulação de um artefato executável com entradas e pré-condições conhecidas e a comparação entre a saída atual e a saída esperada para determinar se o artefato está em conformidade com os requisitos” (FIRESMITH, 2013).

“Teste é o processo de executar um programa ou sistema com a intenção de encontrar erros” (MYERS, 2004).



Revisão

Casos de teste

Casos de teste é uma sequência de passos que devem ser executados no sistema, sendo que os dados de entrada e saída esperada para cada passo são especificados. Os casos de teste devem “direcionar” as ações do testador em uma determinada funcionalidade e fazer com que ele observe se o resultado que ele obtém em cada passo é o resultado esperado, de acordo com os requisitos



Revisão

Cenário de Testes

De acordo com a norma IEEE 829-2008 (IEEE 829-2008, 2008), cenário de teste é um conjunto de casos de teste relacionados, que comumente testam o mesmo componente ou a mesma funcionalidade do sistema.



Revisão

Problema

Ao identificar um problema, o testador deve reportar o defeito de forma a permitir que o time de desenvolvimento identifique facilmente como eliminar o defeito do sistema.



Revisão

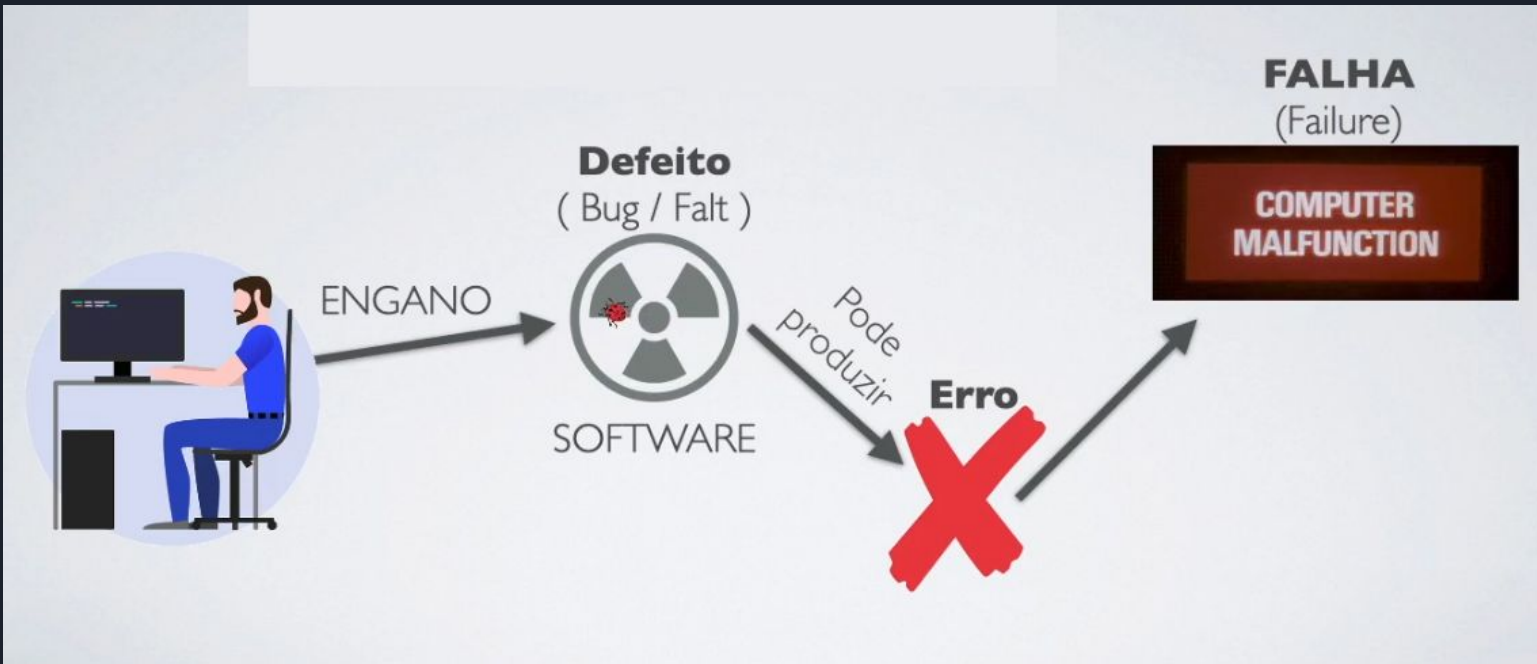
ENGANO (<i>MISTAKE</i>)	Uma ação humana que produz um resultado incorreto, como uma codificação ou modelagem errada;
DEFEITO (<i>FAULT</i>)	Uma imperfeição ou deficiência em um artefato que faz com que este não esteja em conformidade com os requisitos ou especificações, sendo necessária sua correção ou substituição. Termos como "erro" e " <i>bug</i> " comumente são usados para expressar defeitos;



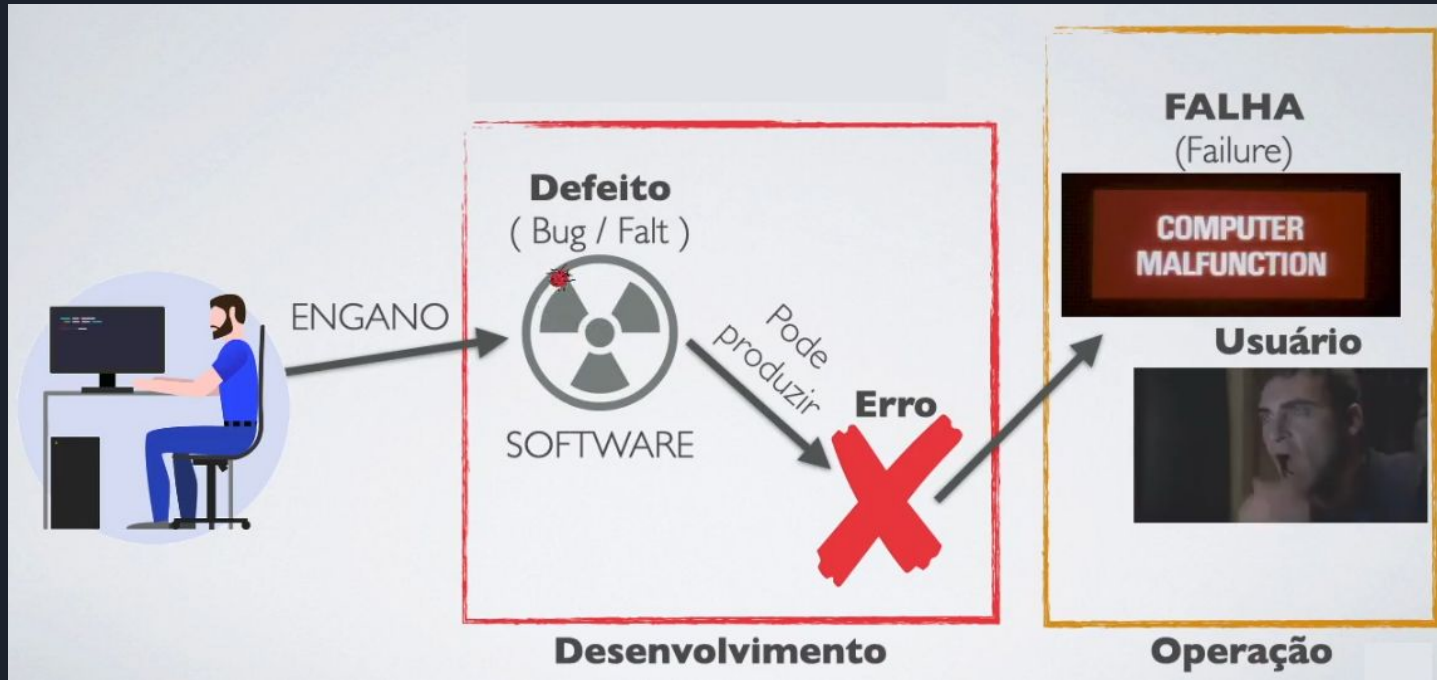
Revisão

ERRO (<i>ERROR</i>)	Resultado incoerente produzido por meio de uma ação no sistema;
FALHA (<i>FAILURE</i>)	Incapacidade do <i>software</i> exercer a função para a qual foi desenvolvido.

Revisão



Revisão





Revisão

Desenvolvimento de *software* e teste de *software*

O programa de estudo da ISTQB (*International Software Testing Qualifications Board*) categoriza os modelos mais comuns de ciclo de vida de desenvolvimento de software em :

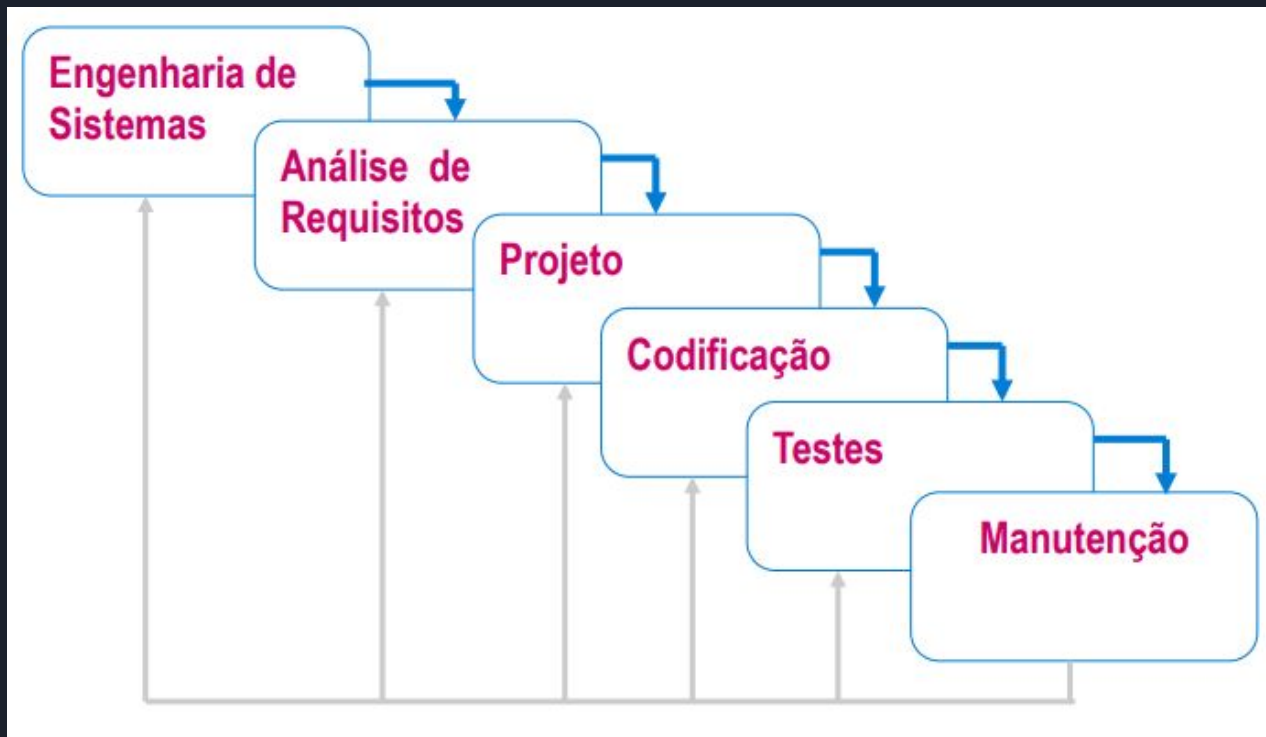
- Modelos de desenvolvimento sequencial;
- Modelos de desenvolvimento iterativo;
- Modelos de desenvolvimento incremental.



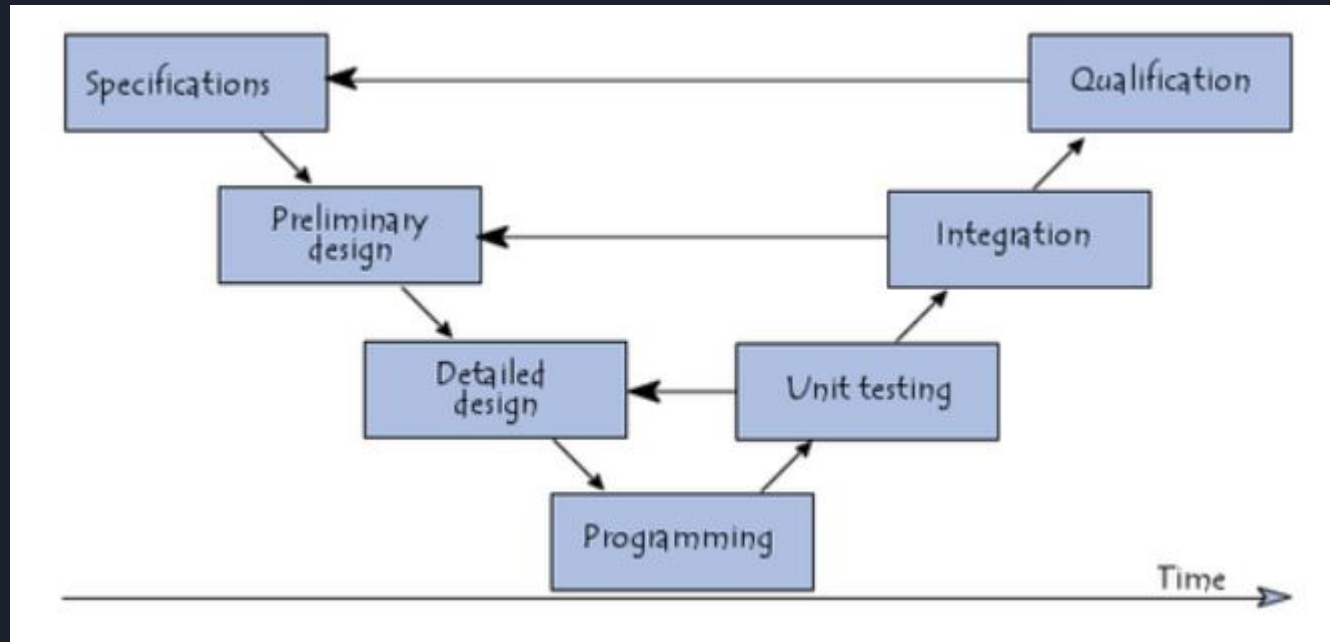
Revisão

Modelo de Desenvolvimento Sequencial: descreve o processo de desenvolvimento de software como um fluxo sequencial e linear de atividades. Isso significa que qualquer fase do processo de desenvolvimento deve começar quando a fase anterior estiver concluída. Em teoria, não há sobreposição de fases, mas, na prática, é benéfico ter antecipadamente o feedback da fase seguinte.

Revisão



Revisão





Revisão

Modelo V

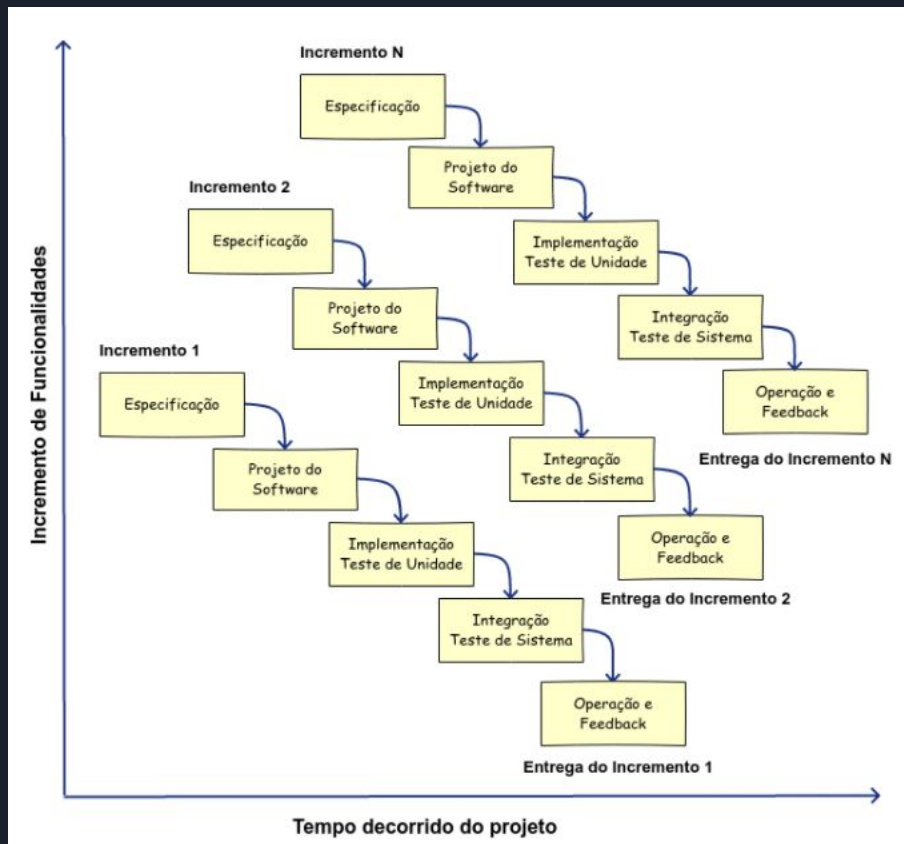
A conexão entre os lados esquerdo e direito do modelo em V implica que, caso sejam encontrados problemas durante a verificação e a validação, o lado esquerdo do V pode ser executado novamente para corrigir e melhorar os requisitos, o projeto e a codificação, antes da execução das etapas de testes que estão no lado direito.



Revisão

Modelo de **desenvolvimento incremental** envolve o estabelecimento de requisitos, a modelagem, a construção e o teste de um sistema em partes, o que significa que os recursos do software crescem de forma incremental.

Revisão





Revisão

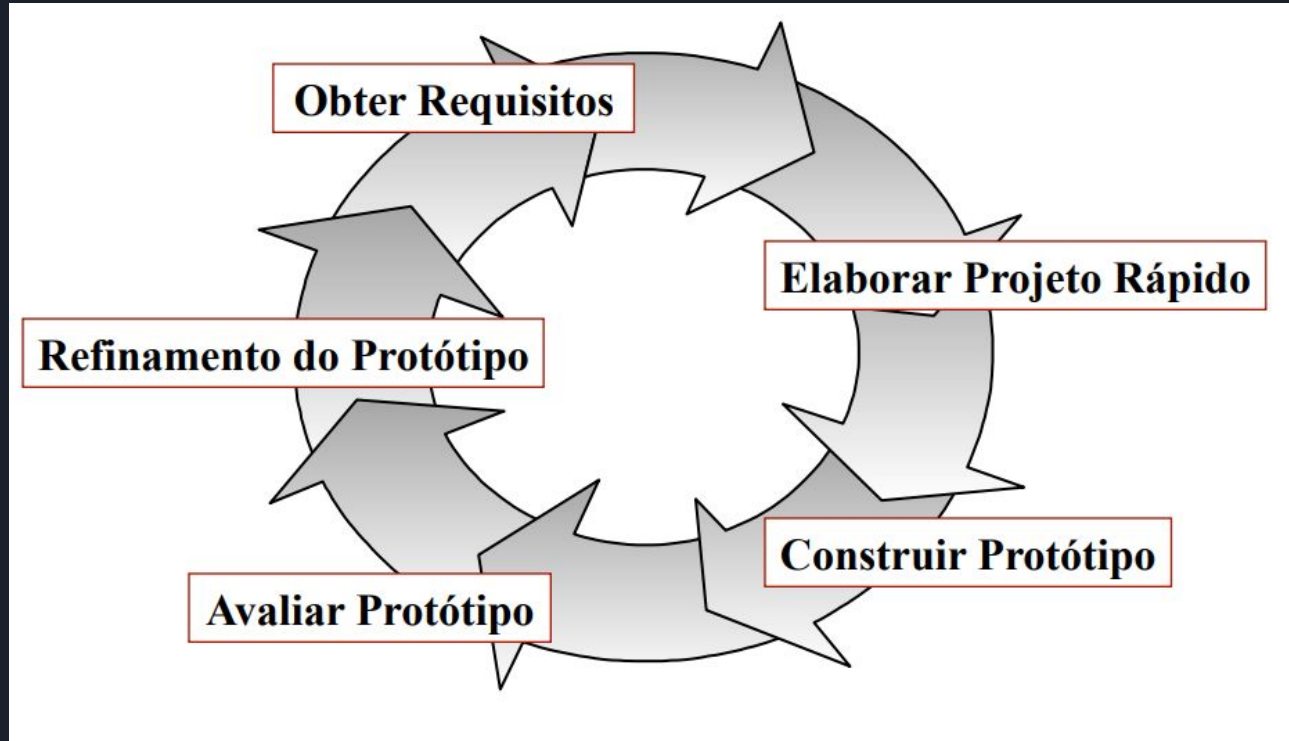
O primeiro incremento é frequentemente chamado de “núcleo do produto” (PRESSMAN, 2006) e contém a implementação dos requisitos básicos para que o sistema possa funcionar e atender minimamente às necessidades do cliente.



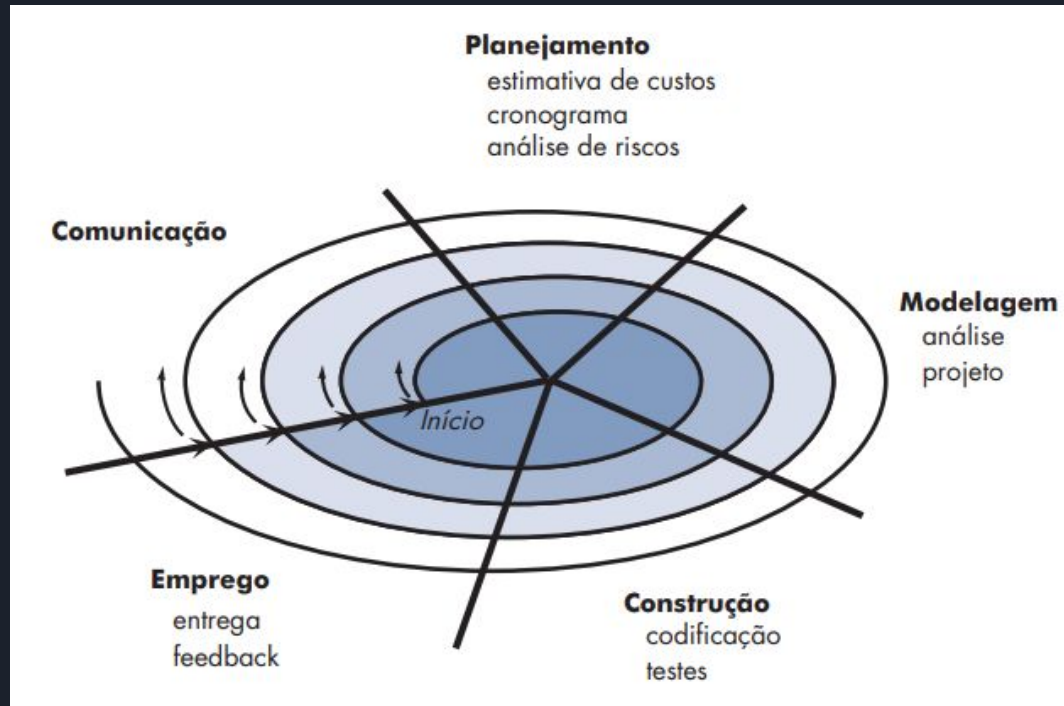
Revisão

O **desenvolvimento iterativo** ocorre quando grupos de recursos são especificados, projetados, construídos e testados juntos em uma série de ciclos, geralmente com duração fixa. Iterações podem envolver mudanças em recursos desenvolvidos em iterações anteriores, juntamente com mudanças no escopo do projeto.

Revisão



Revisão



Revisão





Processos de Teste

“Não existe um processo universal de teste de software, mas há conjuntos comuns de atividades de teste sem as quais os testes terão menor probabilidade de atingir seus objetivos estabelecidos”

(ISTQB, 2019)



Processos de Teste

O processo de teste de software específico e apropriado em qualquer situação depende de muitos fatores:

- Quais atividades de teste estão envolvidas nesse processo de teste;
- Como essas atividades são implementadas;
- Quando essas atividades ocorrem.



Processo de teste no contexto

Fatores contextuais que influenciam o processo de teste de uma organização:

- Modelo de ciclo de vida de desenvolvimento de software e metodologias de projeto utilizados;
- Níveis de teste e tipos de teste considerados;
- Riscos de produto e projeto;
- Domínio do negócio;



Processo de teste no contexto

Fatores contextuais que influenciam o processo de teste de uma organização:

- Algumas restrições operacionais:
 - Orçamentos e recursos;
 - Escalas de tempo;
 - Complexidade;
 - Requisitos contratuais e regulamentares.



Processo de teste no contexto

Fatores contextuais que influenciam o processo de teste de uma organização:

- Políticas e práticas organizacionais;
- Normas internas e externas necessárias.



Atividades e tarefas de teste

Um processo de teste consiste nos seguintes grupos principais de atividades:

- Planejamento do teste;
- Monitoramento e controle do teste;
- Análise do teste;
- Modelagem do teste;
- Implementação do teste;
- Execução do teste;
- Conclusão do teste.



Atividades e tarefas de teste

Embora muitos desses grupos de atividades possam parecer logicamente sequenciais, eles são frequentemente implementados iterativamente.

Exemplo:

O desenvolvimento ágil envolve pequenas iterações de projeto, construção e teste de software que acontecem de forma contínua, suportadas pelo planejamento contínuo.



Atividades e tarefas de teste

Um processo de teste consiste nos seguintes grupos principais de atividades:

- Planejamento do teste;
- Monitoramento e controle do teste;
- Análise do teste;
- Modelagem do teste;
- Implementação do teste;
- Execução do teste;
- Conclusão do teste.



Planejamento do teste

O planejamento do teste envolve as atividades que definem os propósitos e a abordagem do teste para atender aos objetivos do teste dentro das restrições impostas pelo contexto. Os planos de teste podem ser revisitados com base no feedback das atividades de monitoramento e controle.

Exemplo: especificar técnicas e tarefas de teste adequadas e formular um cronograma de teste para cumprir um prazo.



Atividades e tarefas de teste

Um processo de teste consiste nos seguintes grupos principais de atividades:

- Planejamento do teste;
- Monitoramento e controle do teste;
- Análise do teste;
- Modelagem do teste;
- Implementação do teste;
- Execução do teste;
- Conclusão do teste.



Monitoramento e controle do teste

Monitoramento de teste:

Envolve a comparação contínua do progresso real com o plano de teste usando qualquer métrica de monitoramento definida no plano de teste.



Monitoramento e controle do teste

Controle do teste:

Engloba a tomada de ações necessárias para atender aos objetivos do plano de teste (que pode ser atualizado ao longo do tempo).



Monitoramento e controle do teste

Controle do teste:

Engloba a tomada de ações necessárias para atender aos objetivos do plano de teste (que pode ser atualizado ao longo do tempo).



Monitoramento e controle do teste

O progresso do teste em relação ao plano é comunicado aos *stakeholders* nos relatórios de progresso do teste, incluindo os desvios do plano e as informações para apoiar qualquer decisão de interromper o teste.

Os **stakeholders** (partes interessadas, em português) são as pessoas e as organizações que podem ser afetadas por um projeto ou empresa, de forma direta ou indireta, positiva ou negativamente



Atividades e tarefas de teste

Um processo de teste consiste nos seguintes grupos principais de atividades:

- Planejamento do teste;
- Monitoramento e controle do teste;
- Análise do teste;
- Modelagem do teste;
- Implementação do teste;
- Execução do teste;
- Conclusão do teste.



Análise do teste

Durante a análise do teste, a base de teste é analisada para identificar recursos testáveis e definir as condições de teste associadas. Em outras palavras, a análise do teste determina “o que testar” em termos dos critérios de cobertura mensuráveis.



Análise do teste

A análise do teste inclui as seguintes atividades principais:

- Analisar a base de teste apropriada ao nível de teste que está sendo utilizado;
- Avaliar a base de teste e os itens de teste para identificar os vários tipos de defeitos;
- Identificar os recursos e os conjuntos de recursos a serem testados;



Análise do teste

A análise do teste inclui as seguintes atividades principais:

- Analisar a base de teste apropriada ao nível de teste que está sendo utilizado;
- Avaliar a base de teste e os itens de teste para identificar os vários tipos de defeitos;
- Identificar os recursos e os conjuntos de recursos a serem testados;



Análise do teste

A análise do teste inclui as seguintes atividades principais:

- Definir e priorizar as condições de teste para cada recurso com base na análise da base de teste e considerando as características funcionais, não-funcionais e estruturais, outros fatores comerciais e técnicos e níveis de riscos;

Requisito Funcional: expressa uma ação que deve ser realizada através do sistema, ou seja, um requisito funcional é “o que sistema **DEVE** fazer”

Requisito não funcional: por sua vez pode ser definido como “de qual maneira o sistema deve fazer”.



Análise do teste

As atividades de análise do teste não apenas verificam se os requisitos são consistentes, expressos adequadamente e completos, mas também validam se os requisitos capturam adequadamente as necessidades do cliente, do usuário e *stakeholders*.



Atividades e tarefas de teste

Um processo de teste consiste nos seguintes grupos principais de atividades:

- Planejamento do teste;
- Monitoramento e controle do teste;
- Análise do teste;
- Modelagem do teste;
- Implementação do teste;
- Execução do teste;
- Conclusão do teste.



Modelagem do teste

Durante a modelagem de teste, as condições de teste são elaboradas em casos de teste de alto nível, em conjuntos de casos de teste de alto nível e outros *testwares*.

Testware é uma ferramenta para automação de testes de caixa preta, que visa auxiliar na definição de casos de teste e sua execução.

Testwares são programas criados para testarem as funções de outros programas.



Modelagem do teste

Análise vs Modelagem:

A análise do teste responde à pergunta “o que testar?”, enquanto a modelagem de teste responde à pergunta “como testar?”



Modelagem do teste

A modelagem de teste inclui as seguintes atividades principais:

- Projetar e priorizar casos de teste e conjuntos de casos de teste;
- Identificar os dados de teste necessários para comportar as condições de teste e os casos de teste;
- Projetar o ambiente de teste e identificar qualquer infraestrutura e ferramenta necessária;
- Capturar a rastreabilidade bidirecional entre a base de teste, as condições de teste, os casos de teste e os procedimentos de teste.



Atividades e tarefas de teste

Um processo de teste consiste nos seguintes grupos principais de atividades:

- Planejamento do teste;
- Monitoramento e controle do teste;
- Análise do teste;
- Modelagem do teste;
- Implementação do teste;
- Execução do teste;
- Conclusão do teste.



Implementação do teste

Durante a implementação do teste, o *testware* necessário para a execução do teste é criado ou concluído, incluindo o sequenciamento dos casos de teste nos procedimentos de teste. Portanto, a modelagem de teste responde à pergunta "como testar?", enquanto a implementação do teste responde à pergunta "agora temos tudo para executar os testes?"

Testware é uma ferramenta para automação de testes de caixa preta, que visa auxiliar na definição de casos de teste e sua execução.

Testwares são programas criados para testarem as funções de outros programas.



Implementação do teste

A implementação do teste inclui principalmente as seguintes atividades:

- Desenvolver e priorizar os procedimentos de teste e, potencialmente, criar os *scripts* de teste automatizados;
- Criar as suítes de teste a partir dos procedimentos de teste e (se houver) os *scripts* de teste automatizados;
- Organizar os conjuntos de testes dentro de um cronograma de maneira que resulte em maior eficiência a execução dos testes;



Implementação do teste

A implementação do teste inclui principalmente as seguintes atividades:

- Construir o ambiente de teste (incluindo, potencialmente, estrutura de teste, virtualização de serviços, simuladores e outros itens de infraestrutura), e verificando se tudo o que é necessário foi configurado corretamente;
- Preparar os dados de teste e garantir que eles sejam carregados corretamente no ambiente de teste;



Implementação do teste

As tarefas do projeto de teste e a implementação do teste são frequentemente combinadas.

Nos testes exploratórios e em outros tipos de testes baseados na experiência, a modelagem e a implementação do teste podem ocorrer e serem documentadas como parte da execução do teste.



Atividades e tarefas de teste

Um processo de teste consiste nos seguintes grupos principais de atividades:

- Planejamento do teste;
- Monitoramento e controle do teste;
- Análise do teste;
- Modelagem do teste;
- Implementação do teste;
- Execução do teste;
- Conclusão do teste.



Execução do teste

Durante a execução do teste, os conjuntos de testes são executados de acordo com a programação da execução do teste



Execução do teste

A execução do teste inclui principalmente as seguintes atividades:

- Gravar os identificadores e versões do(s) item(ns) de teste ou do objeto de teste, da(s) ferramenta(s) de teste e testware;
- Executar os testes manualmente ou usando ferramentas de execução do teste;
- Comparar os resultados reais com os resultados esperados;



Execução do teste

A execução do teste inclui principalmente as seguintes atividades:

- Analisar as anomalias para estabelecer suas prováveis causas;
- Comunicar os defeitos com base nas falhas observadas;
- Registrar o resultado da execução do teste;
- Repetir as atividades de teste como resultado de uma ação tomada por uma anomalia, ou como parte do planejado para o teste;



Execução do teste

A execução do teste inclui principalmente as seguintes atividades:

- Verificar e atualizar a rastreabilidade bidirecional entre a base de teste, as condições de teste, os casos de teste, os procedimentos de teste e os resultados de teste.



Atividades e tarefas de teste

Um processo de teste consiste nos seguintes grupos principais de atividades:

- Planejamento do teste;
- Monitoramento e controle do teste;
- Análise do teste;
- Modelagem do teste;
- Implementação do teste;
- Execução do teste;
- Conclusão do teste.



Conclusão do teste

As atividades de conclusão do teste coletam os dados das atividades de teste já concluídas para consolidar a experiência, o *testware* e qualquer outra informação relevante.



Conclusão do teste

As atividades de conclusão do teste ocorrem nos marcos do projeto, como quando um sistema de *software* é lançado, um projeto de teste é concluído (ou cancelado), uma iteração de projeto ágil é concluída (p. ex., como parte de uma reunião retrospectiva), um nível de teste é concluído ou uma liberação de manutenção foi concluída.



Conclusão do teste

A conclusão do teste inclui principalmente as seguintes atividades:

- Verificar se todos os relatórios de defeitos estão fechados, inserindo as solicitações de mudança ou itens de lista não processada do produto para quaisquer defeitos que não foram resolvidos no final da execução do teste;
- Criar um relatório de resumo de teste para ser comunicado aos *stakeholders*;
- Finalizar e arquivar o ambiente de teste, os dados de teste, a infraestrutura de teste e outros *testwares* para posterior reutilização;



Conclusão do teste

A conclusão do teste inclui principalmente as seguintes atividades:

- Entregar o testware para as equipes de manutenção, outras equipes de projeto ou stakeholders que poderiam se beneficiar de seu uso;
- Analisar as lições aprendidas das atividades de teste concluídas para determinar as alterações necessárias para futuras iterações, releases e projetos;
- Usar as informações coletadas para melhorar a maturidade do processo de teste.




Produtos de trabalho do teste

Os produtos de teste são criados como parte do processo de teste. Assim como há uma variação significativa na maneira como as organizações implementam o processo de teste, há também uma variação significativa nos tipos de produtos de trabalho criados durante esse processo, nas formas como esses produtos são organizados e gerenciados e nos nomes usados.




Produtos de trabalho do teste

O padrão [ISO29119-3] também pode servir como diretriz para produtos de trabalho de teste.




Produtos de trabalho do planejamento do teste

Os produtos de trabalho de planejamento do teste geralmente incluem um ou mais planos de teste. O plano de teste inclui informações sobre a base de teste, com as quais os outros produtos de teste serão relacionados através das informações de rastreabilidade, bem como os critérios de saída (ou definidos como “feito”).




Produtos de trabalho de monitoramento e controle do teste

Os produtos de trabalho de monitoramento e controle do teste geralmente incluem vários tipos de relatórios, incluindo relatórios de progresso do teste (produzidos em uma base contínua ou regular) e relatórios de resumo do teste (produzidos em vários marcos de conclusão)



Produtos de trabalho de monitoramento e controle do teste

Todos os relatórios de teste devem fornecer detalhes relevantes do público sobre o progresso do teste a partir da data do relatório, incluindo o resumo dos resultados da execução do teste assim que eles estiverem disponíveis



Produtos de trabalho de monitoramento e controle do teste

Por fim os produtos de trabalho de monitoramento e controle do teste também devem abordar as preocupações de gerenciamento de projetos, como conclusão das tarefas, a alocação e uso de recursos e o esforço



Produtos de trabalho da análise do teste


Os produtos de trabalho de análise do teste incluem condições de teste definidas e priorizadas, preferencialmente onde cada uma das quais é bidirecionalmente rastreável para o(s) elemento(s) específico(s) da base de teste que a cobre.



Produtos de trabalho da modelagem do teste

A modelagem do teste resulta em casos de teste e conjuntos de casos de teste para exercer as condições de teste definidas na análise do teste.


A modelagem do teste também resulta no projeto ou na identificação dos dados necessários de teste, na modelagem do ambiente de teste, e na identificação de infraestrutura e ferramentas, embora a extensão na qual esses resultados sejam documentados varie significativamente.



Produtos de trabalho da implementação do teste

Os produtos de trabalho de implementação do teste incluem:

- Os procedimentos de teste e seu sequenciamento;
- As suítes de teste;
- Um cronograma de execução do teste.

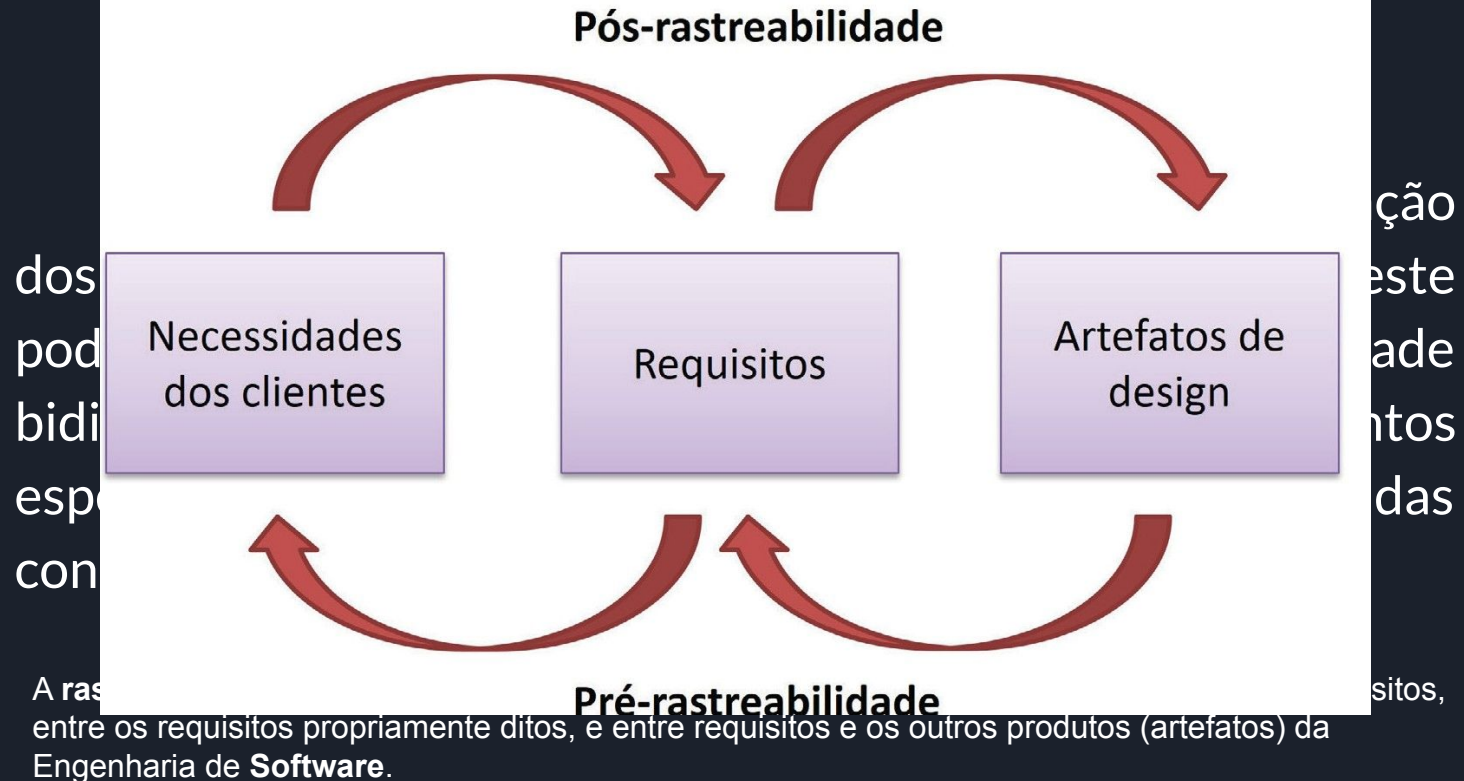



Produtos de trabalho da implementação do teste

Uma vez concluída a implementação do teste, a obtenção dos critérios de cobertura estabelecidos no plano de teste pode ser demonstrada por meio da rastreabilidade bidirecional entre os procedimentos e os elementos específicos da base de teste, através dos casos de teste e das condições de teste.

A **rastreabilidade** de requisitos consiste na identificação de relações entre as fontes dos requisitos, entre os requisitos propriamente ditos, e entre requisitos e os outros produtos (artefatos) da Engenharia de **Software**.

Produtos de trabalho da implementação do teste





Produtos de trabalho da implementação do teste

A implementação do teste também pode resultar na criação e verificação dos dados de teste e do ambiente de teste.

Os dados de teste servem para atribuir valores concretos às entradas e aos resultados esperados da execução dos casos de teste.



Produtos de trabalho da execução do teste

Os produtos de trabalho da execução do teste incluem:

- A documentação do status dos casos de teste individuais ou procedimentos de teste (p. ex., pronto para executar, passar, falhar, bloquear, ignorar deliberadamente etc.);
- Os relatórios de defeitos;
- A documentação sobre quais os itens de teste, o(s) objeto(s) de teste, as ferramentas de teste e o testware estavam envolvidos no teste.



Produtos de trabalho da execução do teste

Uma vez concluída a execução do teste, o status de cada elemento da base de teste pode ser determinado e relatado via rastreabilidade bidirecional com o(s) procedimento(s) de teste associado(s).



Produtos de trabalho da execução do teste

Ex: podemos dizer quais requisitos passaram em todos os testes planejados, quais requisitos falharam nos testes ou que possuem defeitos associados a eles e quais requisitos têm testes planejados ainda aguardando para serem executados




Produtos de trabalho da execução do teste

Isso permite a verificação de que os critérios de cobertura foram atendidos e permite o relato de resultados de testes em termos que sejam compreensíveis para os stakeholders.




Produtos de trabalho de conclusão do teste

Os produtos de trabalho de conclusão do teste incluem os relatórios de resumo de teste, os itens de ação para melhoria de projetos subsequentes ou iterações (p. ex., após um projeto de retrospectiva ágil), as solicitações de mudança ou os itens finalizados de backlog de produto e testware.



Rastreabilidade entre a base de teste e os produtos de trabalho de teste


Os produtos de trabalho de teste e os nomes desses produtos de trabalho variam significativamente. Independentemente dessas variações, a fim de implementar o monitoramento e controle efetivo do teste, é importante estabelecer e manter a rastreabilidade durante todo o processo de teste entre cada elemento da base de teste e os vários produtos de teste associados a esse elemento, conforme descrito acima



Rastreabilidade entre a base de teste e os produtos de trabalho de teste

Além da avaliação da cobertura de teste, a boa rastreabilidade suporta:

- Analisar o impacto das mudanças;
- Tornar o teste auditável;
- Atender aos critérios de governança de TI;
- Melhorar a compreensibilidade dos relatórios de progresso do teste e dos relatórios de resumo do teste para incluir o status dos elementos da base de teste;



Rastreabilidade entre a base de teste e os produtos de trabalho de teste

Além da avaliação da cobertura de teste, a boa rastreabilidade suporta:

- Relacionar os aspectos técnicos do teste com os *stakeholders* em termos que eles possam entender;
- Fornecer informações para avaliar a qualidade do produto, a capacidade do processo e o progresso do projeto em relação às metas de negócios.



Níveis de teste

Os níveis de teste são grupos de atividades de teste que são organizados e gerenciados juntos. Cada nível de teste é uma instância do processo de teste, executadas em relação ao software em um determinado nível de desenvolvimento, desde as unidades individuais ou componentes até os sistemas completos ou, quando aplicável, em sistemas de sistemas.



Níveis de teste

Os níveis de teste estão relacionados com outras atividades dentro do ciclo de vida de desenvolvimento de software:

- Teste de componentes;
- Teste de integração;
- Teste do sistema;
- Teste de aceite.



Níveis de teste

Os níveis de teste são caracterizados pelos seguintes atributos:

- Objetivos específicos;
- Base de teste, referenciada para derivar casos de teste;
- Objeto de teste (ou seja, o que está sendo testado);
- Defeitos e falhas típicas;
- Abordagens e responsabilidades específicas.



Níveis de teste

Para cada nível de teste, é necessário um ambiente de teste adequado. No teste de aceite, por exemplo, um ambiente de teste com características mais próximas do ambiente de produção é ideal, enquanto no teste de componentes os desenvolvedores normalmente usam seu próprio ambiente de desenvolvimento.



Teste de componente

O teste de componente (também conhecido como teste de unidade ou módulo) se concentra em componentes que são testáveis separadamente.



Teste de componente

Objetivos do teste de componente:

- Reduzir o risco;
- Verificar os comportamentos funcionais e não funcional do componente ao projetado e especificado;
- Construir a confiança na qualidade do componente;
- Encontrar defeitos no componente;
- Evitar que os defeitos espalhem para níveis mais altos de teste.



Teste de componente

Em alguns casos, especialmente em modelos de desenvolvimento incremental e iterativo (p. ex., Ágil), em que as alterações de código estão em andamento, os testes de regressão de componentes automatizados desempenham um papel fundamental na criação da confiança de que as alterações não impactaram os componentes existentes.



Teste de componente

Base de teste:

Exemplos de produtos de trabalho que podem ser usados como base de teste para testes de componentes incluem:

- Projeto detalhado;
- Código;
- Modelo de dados;
- Especificações de componentes.



Teste de componente

Objetos de teste:

- Componentes, unidades ou módulos;
- Estruturas de código e dados;
- Classes;
- Módulos de banco de dados.



Teste de componente

Defeitos típicos e falhas:

- Funcionalidade incorreta (p. ex., não conformidade descrita nas especificações do projeto);
- Problemas no fluxo de dados;
- Código e lógica incorretos.



Teste de componente

Defeitos típicos e falhas:

Os defeitos geralmente são corrigidos assim que são encontrados, geralmente sem gerenciamento formal dos defeitos. No entanto, quando os desenvolvedores relatam defeitos, isso fornece informações importantes para análise de causa raiz e melhoria de processo



Teste de componente

Abordagens e responsabilidades específicas:

O teste de componente geralmente é executado pelo desenvolvedor que escreveu o código, mas obrigatoriamente requer acesso ao código que está sendo testado.

Os desenvolvedores podem alternar o desenvolvimento de componentes com a localização e correção de defeitos.

Os desenvolvedores geralmente escrevem e executam testes depois de codificarem um componente.



Teste de componente

Abordagens e responsabilidades específicas:

No entanto, especialmente no desenvolvimento ágil, escrever os casos de teste de componentes automatizados pode preceder a gravação do código do aplicativo.



Teste de Integração

O teste de integração se concentra nas interações entre componentes ou sistemas.

Os objetivos do teste de integração:

- Reduzir risco;
- Verificar se os comportamentos funcionais e não-funcionais das interfaces estão projetados e especificados;
- Construir confiança na qualidade das interfaces;



Teste de Integração

Os objetivos do teste de integração:

- Encontrar defeitos (que podem estar nas próprias interfaces ou nos componentes ou sistemas);
- Evitar que os defeitos espalhem para níveis mais altos de teste.



Teste de Integração

Assim como no teste de componentes, em alguns casos, o teste de regressão de integração automatizada garante que as alterações não interromperam as interfaces, os componentes ou sistemas existentes.



Teste de Integração

Existem dois níveis diferentes de teste de integração:

- O teste de integração de componentes foca na integração e interface entre componente integrados.
- O teste de integração do sistema concentra-se nas interações e interfaces entre sistemas, pacotes e micro serviços.



Teste de Integração

Base de teste:

- Software e modelagem do sistema;
- Diagramas de sequência;
- Especificações de interface e protocolo de comunicação;
- Casos de uso;
- Arquitetura no nível de componente ou sistema;
- Fluxos de trabalho;
- Definições de interface externa.



Teste de Integração

Objetos de teste

- Subsistemas;
- Bancos de dados;
- Infraestrutura;
- Interfaces;
- APIs;
- Micro serviços.



Teste de Integração

Defeitos típicos e falhas:

Defeitos e falhas típicos para testes de integração de componentes:

- Dados incorretos, dados ausentes ou codificação de dados incorreta;
- Sequenciamento ou temporização incorretos de chamadas de interface;
- Incompatibilidade de interface;
- Falhas na comunicação entre componentes;
- Falha de comunicação não manipulada ou tratada de forma errada entre componentes;
- Suposições incorretas sobre o significado, as unidades ou limites dos dados que estão sendo transmitidos entre os componentes.



Teste de Integração

Defeitos típicos e falhas:

Defeitos e falhas típicos para testes de integração de sistemas incluem:

- Estruturas de mensagens inconsistentes entre sistemas;
- Dados incorretos, dados ausentes ou codificação de dados incorreta;
- Incompatibilidade de interface;
- Falhas na comunicação entre sistemas;
- Falha de comunicação não manipulada ou tratada de forma errada entre sistemas;
- Suposições incorretas sobre o significado, unidades ou limites dos dados que estão sendo transmitidos entre sistemas;
- Falha no cumprimento dos regulamentos de segurança obrigatórios



Teste de Integração

Abordagens e responsabilidades específicas:

Os testes de integração de componentes e os testes de integração de sistemas devem se concentrar na própria integração. Por exemplo, se integrar o módulo A com o módulo B, os testes devem focar na comunicação entre ambos e não na funcionalidade dos módulos individualmente, como deveria ter sido coberto durante o teste de componentes.



Teste de Integração

Abordagens e responsabilidades específicas:

O teste de integração de componentes geralmente é de responsabilidade dos desenvolvedores, e o teste de integração do sistema geralmente é de responsabilidade dos testadores. No formato ideal, os testadores que realizam o teste de integração do sistema devem entender da arquitetura do sistema e devem ter influenciado no planejamento da integração



Teste de Sistema

O teste de sistema se concentra no comportamento e nas capacidades de todo um sistema ou produto, geralmente considerando as execuções das tarefas de ponta a ponta do sistema e os comportamentos não-funcionais exibidos ao executar tais tarefas.



Teste de Sistema

Os objetivos do teste do sistema incluem:

- Reduzir o risco;
- Verificar se os comportamentos funcionais e não-funcionais do sistema estão como projetados e especificados;
- Validar se o sistema está completo e funcionará como esperado;
- Criar confiança na qualidade do sistema como um todo;
- Encontrar defeitos;
- Evitar que os defeitos espalhem para níveis mais altos de teste ou produção.



Teste de Sistema

Base de teste:

- Especificações de requisitos de sistema e software (funcionais e não-funcionais);
- Relatórios de análise de risco;
- Casos de uso;
- Épicos e histórias de usuários;
- Modelos de comportamento do sistema;
- Diagramas de estado;
- Sistema e manuais do usuário;



Teste de Sistema

Objetos de teste

- Aplicações;
- Sistemas de hardware e software;
- Sistemas operacionais;
- Sistema sob teste (SUT);
- Configuração do sistema e dados de configuração.

Sistema sob **Teste** (System Under Test, **SUT**): sistema que está sendo testado. É um nome genérico, usado também em outros tipos de **testes**, não necessariamente de unidades. Às vezes, usa-se também o termo código de produção, ou seja, código que vai ser executado pelos clientes do sistema.



Teste de Sistema

Defeitos típicos e falhas

- Cálculos incorretos;
- Comportamento funcional ou não funcional do sistema incorreto ou inesperado;
- Controle e/ou fluxos de dados dentro do sistema incorretos;
- Falha na execução correta e completa de tarefas funcionais de ponta a ponta;
- Falha do sistema em funcionar adequadamente no(s) ambiente(s) de produção;
- Falha do sistema para funcionar conforme descrito nos manuais do sistema e do usuário.



Teste de Sistema

Abordagens e responsabilidades específicas:

O teste de sistema deve focar no comportamento geral, funcional ou não, de ponta a ponta do sistema como um todo.

Testadores independentes geralmente realizam testes no sistema.



Teste de aceite

O teste de aceite, como o teste do sistema, geralmente se concentra no comportamento e na capacidade de todo um sistema ou produto.



Teste de aceite

Os objetivos do teste de aceite incluem:

- Estabelecer confiança na qualidade do sistema como um todo;
- Validar que o sistema está completo e funcionará como esperado;
- Verificar se os comportamentos funcionais e não-funcionais do sistema são os especificado



Teste de aceite

Formas comuns de testes de aceite:

- Teste de aceite do usuário;
- Teste de aceite operacional;
- Teste de aceite contratual e regulatório;
- Alfa teste e Beta.



Teste de aceite

Teste de aceite do usuário (UAT):

O teste de aceite do sistema pelos usuários é tipicamente focado em validar a adequação do uso do sistema pelos usuários pretendidos em um ambiente operacional real ou simulado.



Teste de aceite

Teste de aceite operacional (OAT)

O teste de aceite do sistema pelas operações ou pela equipe de administração de sistemas geralmente é realizado em um ambiente de produção (simulado).



Teste de aceite

Teste de aceite contratual e regulatório:

O teste de aceite contratual é realizado com base nos critérios de aceite de um contrato para desenvolver softwares específicos. Os critérios de aceite devem ser definidos quando as partes concordam com o contrato.



Teste de aceite

Alfa teste e beta teste:

Os “alfa” e “beta” teste são normalmente usados por desenvolvedores de software comercial de prateleira (COTS) que desejam obter feedback de usuários, clientes ou operadores em potencial ou existentes antes que o produto de software seja colocado no mercado.



Teste de aceite

Base de teste:

- Processos de negócios;
- Requisitos do usuário ou de negócios;
- Regulamentos, contratos legais e normas;
- Casos de uso;
- Requisitos de sistema;
- Documentação do sistema ou usuário;
- Procedimentos de instalação;
- Relatórios de análise de risco.



Teste de aceite

Objetos de teste típicos

- Sistema sob teste;
- Configuração do sistema e dados de configuração;
- Processos de negócios para um sistema totalmente integrado;
- Sistemas de recuperação (para continuidade de negócios e testes de recuperação de desastres);
- Processos operacionais e de manutenção;
- Formulários;
- Relatórios;
- Dados de produção existentes e convertidos.



Teste de aceite

Defeitos típicos e falhas:

- Fluxos de trabalho do sistema não atendem aos requisitos do negócio ou do usuário;
- Regras de negócios não são implementadas corretamente;
- O sistema não satisfaz os requisitos contratuais ou regulatórios;
- Falhas não-funcionais, como vulnerabilidades de segurança, eficiência de performance inadequada sob altas cargas ou operação inadequada em uma plataforma suportada



Teste de aceite

Abordagens e responsabilidades específicas:

O teste de aceite é geralmente responsabilidade dos clientes, usuários de negócios, proprietários de produtos ou operadores de um sistema, além de stakeholders que também podem estar envolvidos.

O teste de aceite é frequentemente considerado como o último nível de teste em um ciclo de vida de desenvolvimento sequencial.



Tipo de teste

Um tipo de teste é um grupo de atividades de teste destinado a testar características específicas de um sistema de software, ou parte de um sistema, com base em objetivos de teste específicos.



Teste funcional

O teste funcional de um sistema envolve testes que avaliam as funções que o sistema deve executar.



Teste funcional

A eficácia dos testes funcionais pode ser medida através da cobertura funcional. A cobertura funcional é a medida em que algum tipo de elemento funcional foi exercido por testes e é expresso como uma porcentagem do(s) tipo(s) de elemento a ser coberto.



Teste não funcional

Os testes não-funcionais de um sistema avaliam as características de sistemas e de *softwares*, como usabilidade, eficiência de performance ou segurança.

O teste não-funcional deve ser realizado em todos os níveis de teste, e feito o mais cedo possível. A descoberta tardia de defeitos não-funcionais pode ser extremamente perigosa para o sucesso de um projeto.



Teste não funcional

A eficácia dos testes não-funcionais pode ser medida por meio de cobertura não funcional. A cobertura não funcional é a medida em que algum tipo de elemento não funcional foi exercido por testes e é expressa como uma porcentagem do tipo de elemento a ser coberto.



Teste caixa-branca

O teste caixa-branca é derivado de testes com base na estrutura interna ou na implementação do sistema. A estrutura interna pode incluir código, arquitetura, fluxos de trabalho e fluxos de dados dentro do sistema



Teste caixa-branca

A eficácia dos testes caixa-branca pode ser medida através da cobertura estrutural. A cobertura estrutural é a extensão em que algum tipo de elemento estrutural foi testado e é expresso como uma porcentagem do tipo de elemento a ser coberto



Teste relacionado a mudança

Quando são feitas alterações em um sistema, seja para corrigir um defeito ou por causa de uma funcionalidade nova ou variável, deve-se testar para confirmar se as alterações corrigiram o defeito ou implementaram a funcionalidade corretamente e não causaram consequências adversas imprevistas.



Teste relacionado a mudança

Especialmente em ciclo de vida de desenvolvimento iterativo e incremental (p. ex., Ágil), novos recursos, alterações nos recursos existentes e recompilação de código resultam em alterações frequentes no código, o que também requer testes relacionados a estas alterações.



Teste de manutenção

O teste de manutenção se concentra em testar as alterações no sistema, bem como em testar as partes inalteradas que podem ter sido afetadas pelas alterações. A manutenção pode envolver lançamentos planejados e liberações não planejadas (hot fixes).



Exercícios

1. Cite 3 atividade de um grupo de teste, junto a uma breve explicação.
2. Explique o que é rastreabilidade de requisitos.
3. Liste os níveis de teste de software, explique 2 deles.