Aula - Teste de Software

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campinas

Aula - Teste de Software

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campinas

Aula - Teste de Software

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campinas

-Teste Caixa Preta
-Teste Caixa Branca
-Teste de Unidade
-Teste de Integração
-Teste de Validação: teste Alfa e teste Beta
-Teste de Sistema: teste de Recuperação, teste de Segurança e teste de Estresse.

Teste de Software

- O desenvolvimento de software utilizando as metodologias, técnicas e ferramentas da Engenharia de Software **não oferece a total garantia de qualidade do produto obtido**, apesar de melhorá-la significativamente.

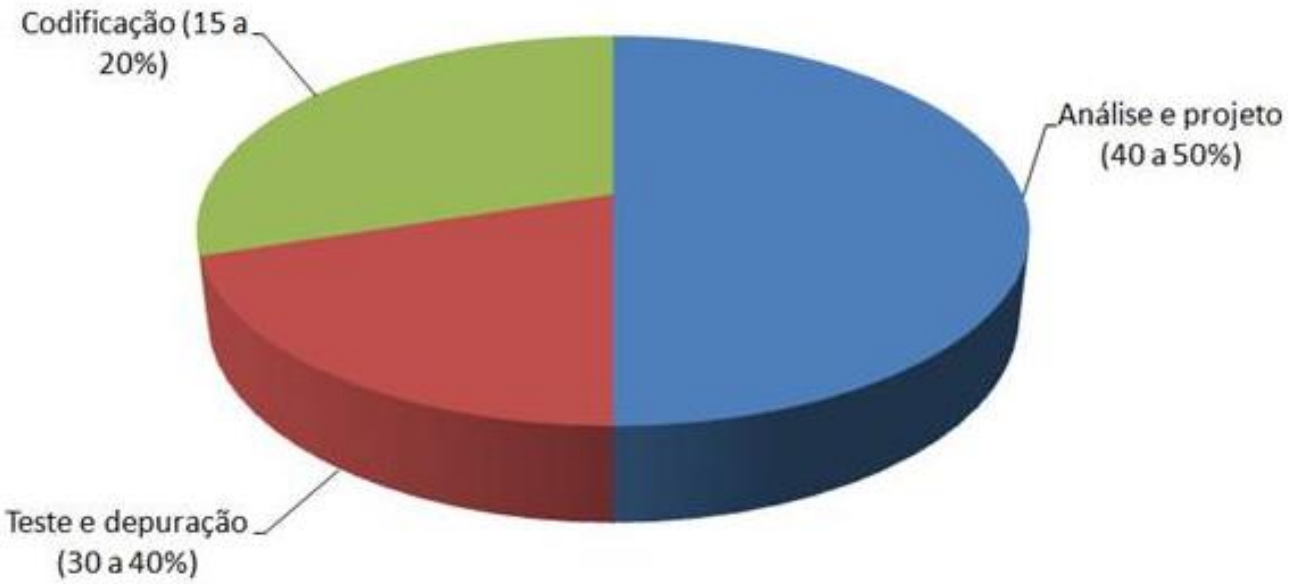
- Por esta razão, uma etapa fundamental na obtenção de um alto nível de qualidade do software a ser produzido é aquela onde são realizados os **procedimentos de teste**, uma vez que esta é a última etapa de revisão da especificação, do projeto e da codificação.

Teste de Software

- A realização, de forma cuidadosa e criteriosa, dos procedimentos associados ao teste de um software assume uma importância cada vez maior dado o impacto sobre o funcionamento (e o custo) que este componente tem assumido nos últimos anos.

Teste de Software

- Por esta razão, o esforço despendido para realizar a etapa de teste pode chegar a **40% do esforço total empregado no desenvolvimento do software,** sob o ponto de Pressman**.**



Teste de Software

- No caso de programas que serão utilizados em **sistemas críticos** (aqueles sistemas dos quais dependem vidas humanas, como controle de vôo e a supervisão de reatores nucleares), a atividade de teste pode custar de 3 a 5 vezes o valor gasto nas demais atividades de desenvolvimento do software.

Teste de Software

- Os objetivos do teste de software podem ser expressos, de forma mais clara, pela observação das três regras definidas por **Myers**:

1. A atividade de teste é o processo de executar um programa com a intenção de descobrir um erro;
2. Um bom caso de teste é aquele que apresenta uma probabilidade de revelar um erro ainda não descoberto.
3. Um teste bem sucedido é aquele que revela um erro ainda não descoberto.

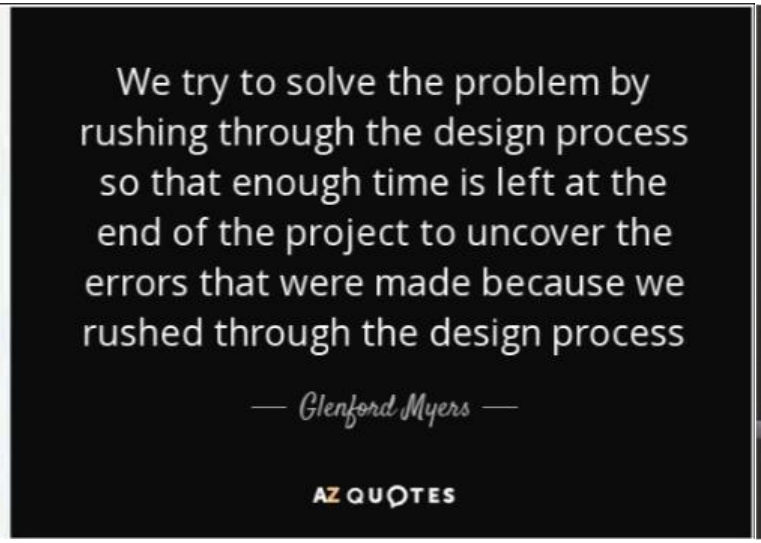
Teste de Software

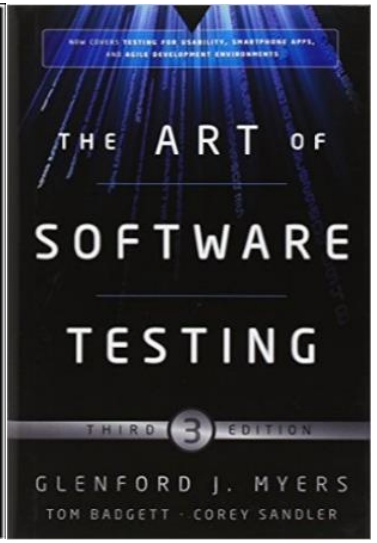
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

- As três regras expressam o objetivo primordial do teste que é o de **encontrar erro**, contrariando a falsa ideia de que uma **atividade de teste bem sucedida é aquela em que nenhum erro foi encontrado.**
- A etapa de teste deve ser conduzida de modo que o maior número de erros possível seja encontrado com um menor dispêndio de tempo e esforço.

Regra 10 de Myers

Em 1979, Glenford Myers em seu livro ‘The Art of Software Testing’ (A arte de teste de software), já apresentava o conceito no qual **quanto mais cedo descobrimos e corrigimos o erro, menor é o seu custo para o projeto.**
Esse custo em correção de **BUGS** cresce 10 vezes para cada estágio em que o projeto do software avança.





Regra 10 de Myers

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campinas

Nos requisitos	Na especificação	No código	Durante a execução do teste	Durante o teste de aceitação	Com o software em produção
- Alterar documento de requisitos.	- Alterar documento de requisitos. - Alterar a especificação.	- Alterar documento de requisitos. - Alterar a especificação. - Alterar o Código.	- Alterar documento de requisitos. - Alterar a especificação. - Alterar o Código. - Alterar o teste, reteste e teste de regressão.	- Alterar documento de requisitos. - Alterar a especificação. - Alterar o Código. - Alterar o teste, reteste e teste de regressão. - Refazer teste de aceitação.	- Alterar documento de requisitos. - Alterar a especificação. - Alterar o Código. - Alterar o teste, reteste e teste de regressão. - Refazer teste de aceitação. - Avisar usuários da falha e da solução de contorno. - O banco de dados do cliente pode necessitar de correções. - Montar um novo build do software pode levar horas.

Aumento do trabalho (Re-trabalho)

Princípios de Teste de Software

Os princípios básicos do teste de qualquer produto resultante de uma tarefa de engenharia são:

- conhecida a função a ser desempenhada pelo produto, testes são executados para para demonstrar que cada função é completamente operacional.

Este primeiro princípio deu origem a uma importante abordagem de teste, conhecida como o **teste de caixa preta** (black box);

Princípios de Teste de Software

Os princípios básicos do teste de qualquer produto resultante de uma tarefa de engenharia são:

- com base no conhecimento do funcionamento interno do produto, realiza-se testes para assegurar de que todas as peças destes estão completamente ajustadas e realizando sua função.

Á abordagem originada por este segundo princípio, foi dado o nome de **teste de caixa branca** (white box).

Teste Caixa Preta

- Quando o procedimento de teste está relacionado ao produto de software, o **teste de caixa preta refere-se a todo teste que implica na verificação do funcionamento do software através de suas interfaces**, o que, geralmente, permite verificar a operacionalidade de todas as suas funções.
- É importante observar que, no teste de caixa preta, a forma **como o software está organizado internamente não tem real importância**, mesmo que isto possa ter algum impacto na operação de alguma função observada em sua interface.

Teste Caixa Branca

- Um **teste de caixa branca num produto de software está relacionado a um exame minucioso de sua estrutura interna e detalhes procedimentais.**
- Os caminhos lógicos definidos no software são exaustivamente testados, pondo à prova conjuntos bem definidos de condições ou laços.
- Durante o teste, o “status” do programa pode ser examinado diversas vezes para eventual comparação com condições de estado esperadas para aquela situação.

Teste Caixa Branca

- Apesar da importância do teste de caixa branca, não se deve guardar a falsa ideia de que a realização de testes de caixa branca num produto de software vai oferecer a garantia de 100% de correção deste ao seu final.
- Isto porque, mesmo no caso de programas de pequeno e médio porte, a diversidade de caminhos lógicos pode atingir um número bastante elevado, representando um grande obstáculo para o sucesso completo desta atividade.

Tipos de Teste

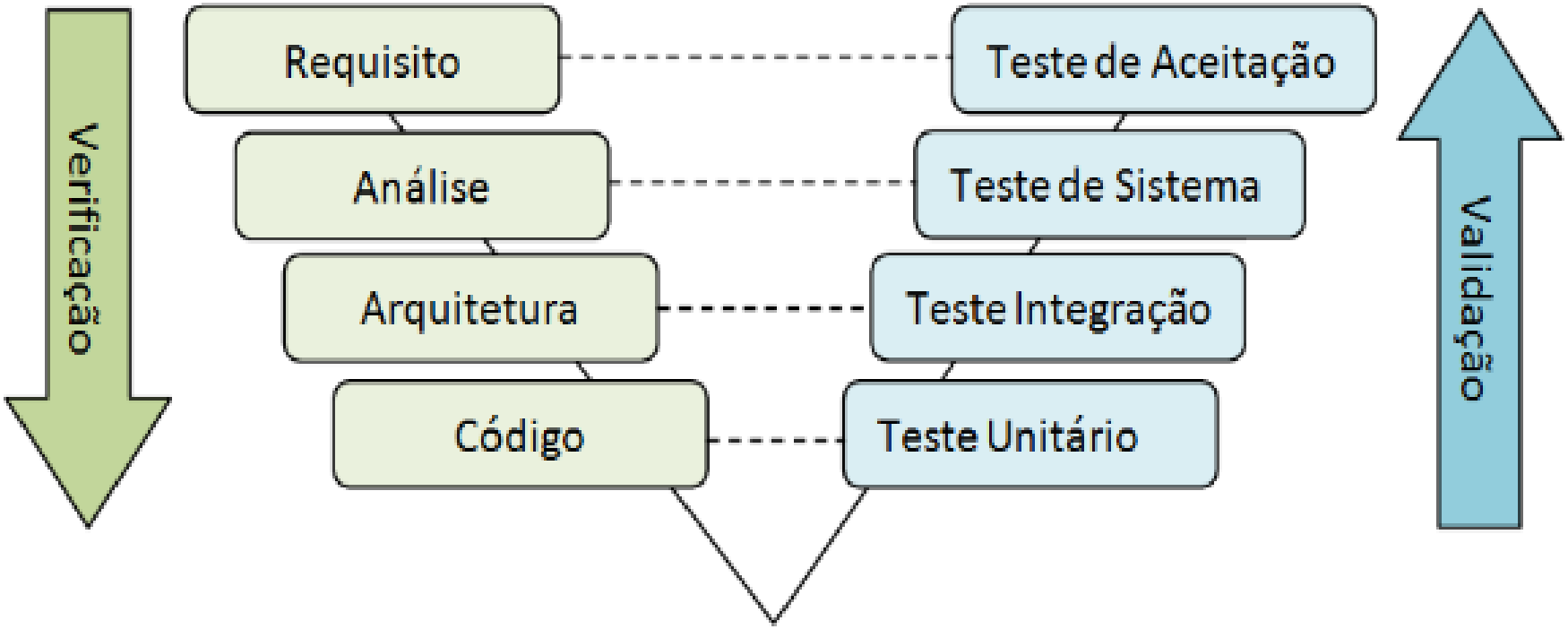
- Outra forma de **subdividir** o teste de software é **quanto ao seu objetivo na busca por erros do programa** são**:**

1. O Teste de Unidade
2. O Teste de Integração
3. O Teste de Validação
4. O Teste de Sistema

Tipos de Teste

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campinas

Modelo “V” (Fases do Desenvolvimento X Fases dos Testes)



Tipos de Teste

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campinas

Modelo “V” (Fases do Desenvolvimento X Fases dos Testes)

Verificação:

Fizemos o software corretamente?

Tem o objetivo de avaliar se o que foi planejado realmente foi realizado. Ou seja, **se os requisitos e funcionalidades documentados foram implementados,** além disso a verificação também pode ser realizada para avaliar se os requisitos estão sendo documentados como deveriam e ainda prever falhas ou inconsistências entre requisitos.

Validação:

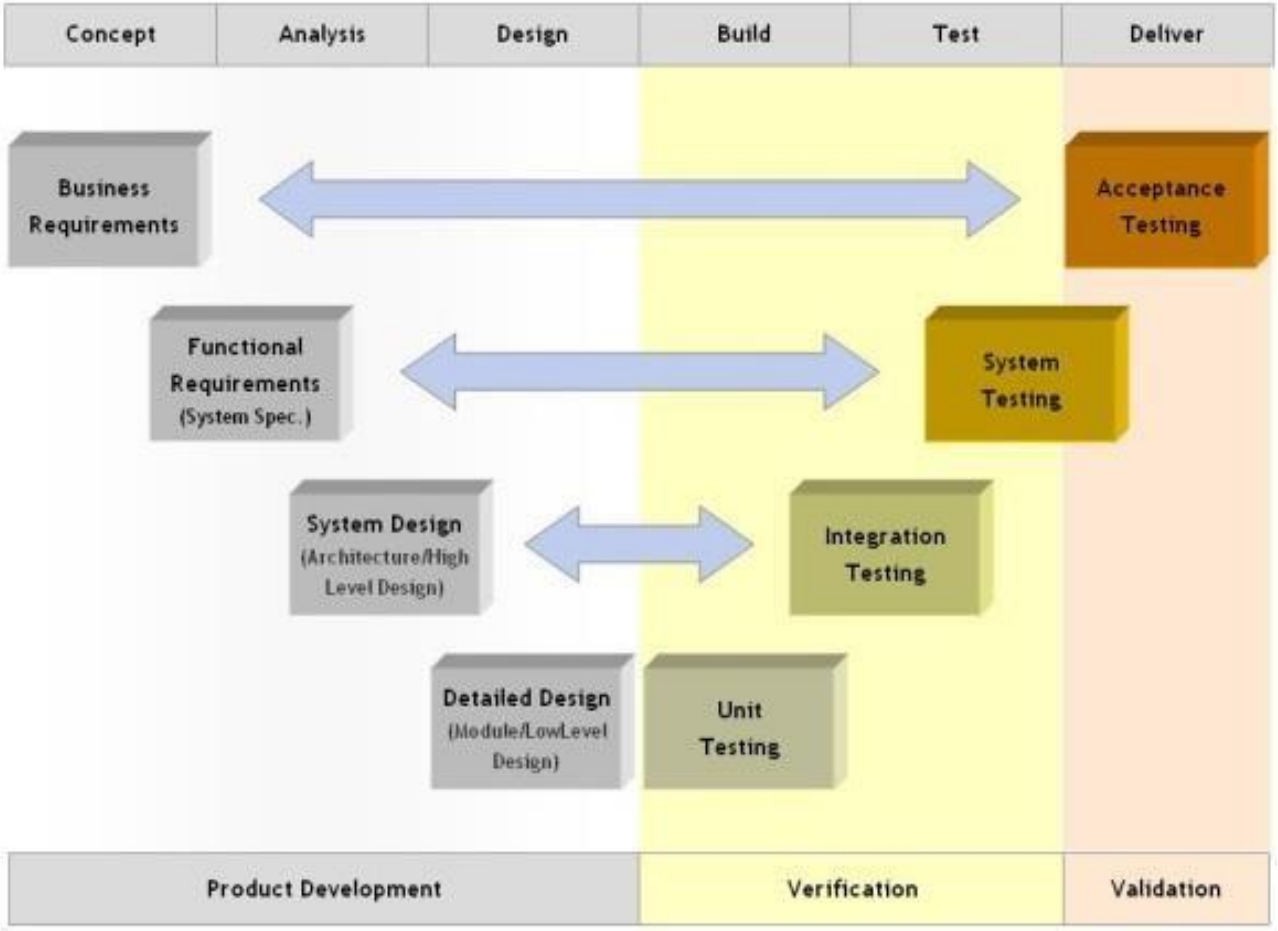
Fizemos o software correto?

Tem o objetivo de avaliar se o que foi entregue atende as expectativas do cliente. Ou seja, **se os requisitos, independente do que foi planejado, estão sendo implementados para atender a regra de negócio do cliente, se o sistema é realmente aquilo que o cliente quer e está pagando para ter.** A validação final do sistema é realizada pelo próprio cliente ou usuário.

O **teste de software** é considerado uma **técnica dinâmica de verificação e validação**, pois o software é executado com dados de teste e seu comportamento é analisado.

Tipos de Teste

- O **Modelo V** é uma variação do modelo cascata, que demonstra como as atividades de testes estão relacionadas com analise e projeto.

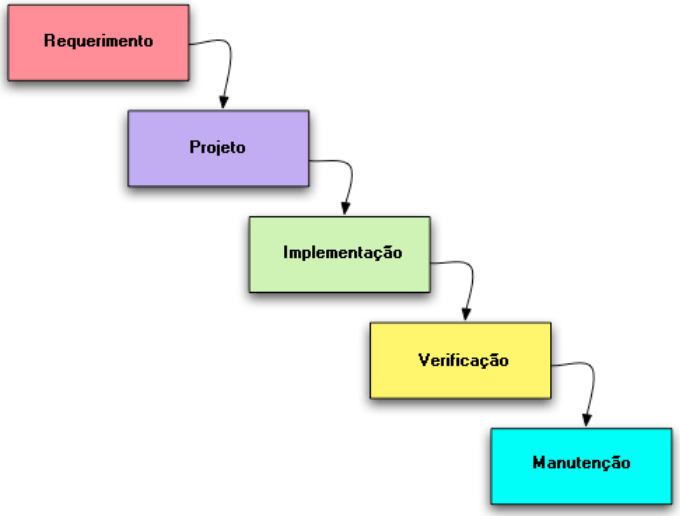


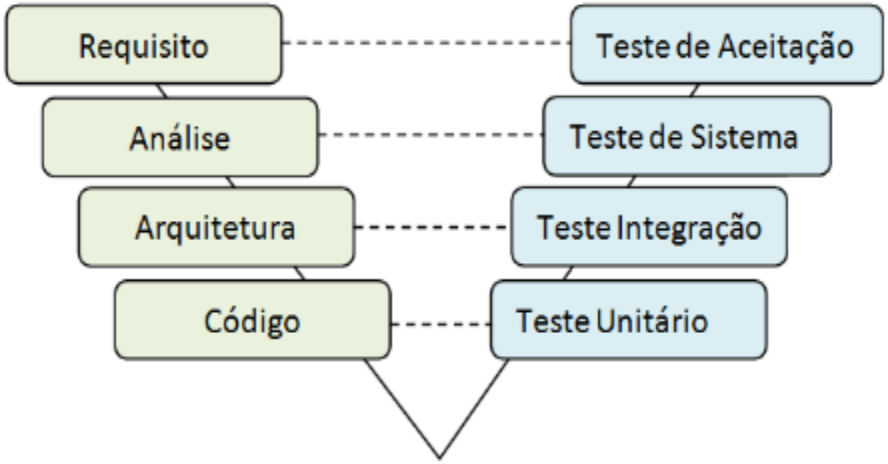
Modelo V

- A conexão entre os lados esquerdo e direito do modelo V implica que, caso sejam encontrados problemas durante a verificação e a validação, o lado esquerdo do modelo 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.
- Em outras palavras o **modelo V torna mais explicitas algumas iterações e repetições do trabalho, ocultas no modelo cascata.**

Modelo V

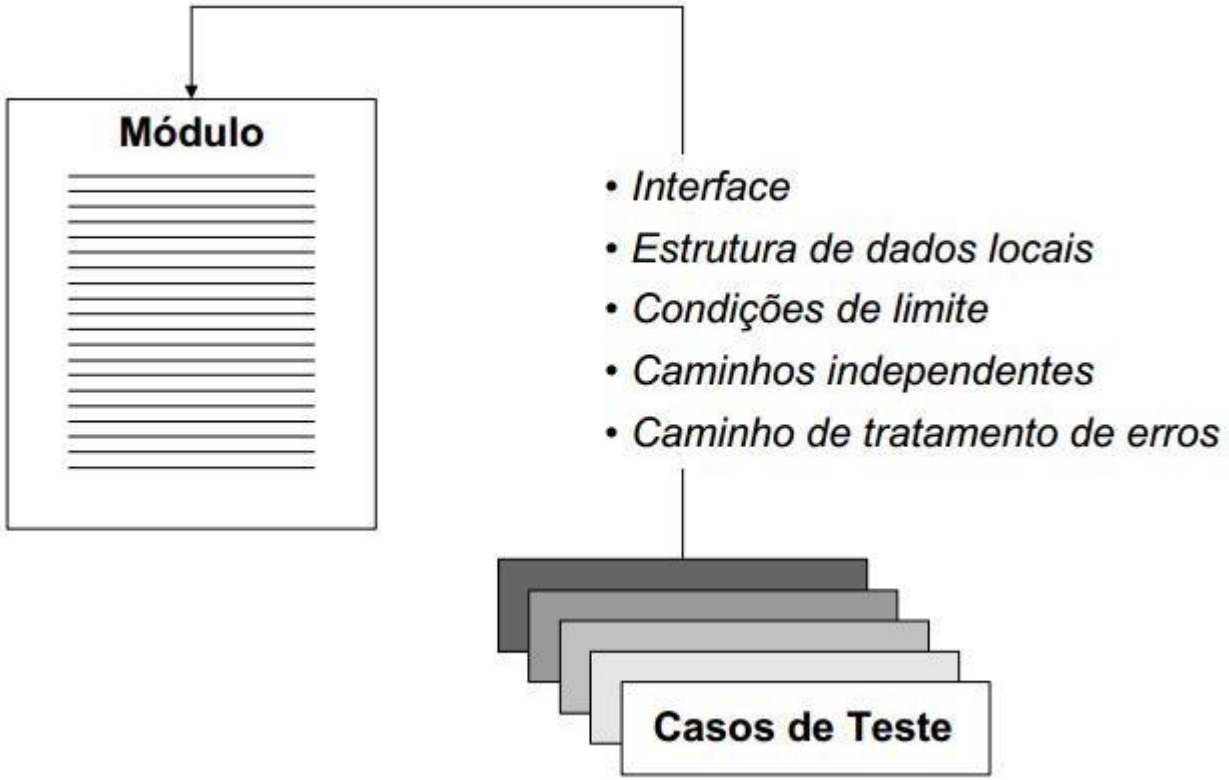
Enquanto **o enfoque do modelo cascata esta nos documentos e nos artefatos, o enfoque do V esta na atividade e na correção.**





- **Teste de Unidade**
- O teste de unidade objetiva a **verificação de erros existentes nas unidades de projeto** do mesmo, à qual daremos o nome de **módulo.**
- Nesta modalidade de teste, é importante **utilizar as informações contidas no documento de projeto detalhado do software**, as quais servirão de guia para sua aplicação.
- O teste de unidade é, de certa forma, **uma técnica de teste de caixa branca**, podendo ser realizado em paralelo sobre diferentes módulos.

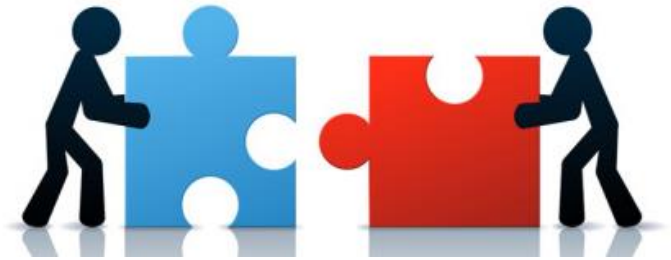
- **Teste de Unidade**
- A figura ilustra a forma como são desenvolvidos os testes de unidade.



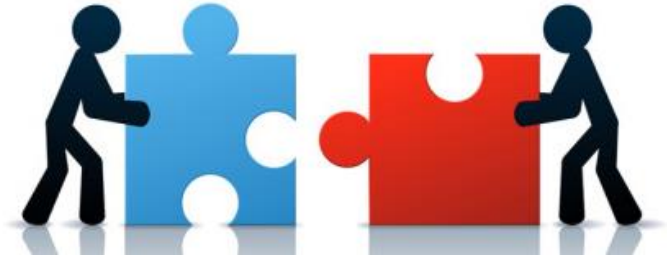
Tipos de Teste

- **Teste de Unidade**

1. A interface, em busca de erros de passagem de dados para dentro e para fora do módulo;
2. As estruturas de dados locais, para se prover a garantia de que todos os dados armazenados localmente mantêm sua integridade ao longo da execução do módulo;
3. As condições limite, que permitem verificar que o módulo executa respeitando os valores máximos e mínimos estabelecidos para seu processamento;
4. Os caminhos independentes (ou caminhos básicos) da estrutura de controle são analisados para se ter garantia de que todas as instruções do módulo foram executadas pelo menos uma vez;
5. Os caminhos de tratamento de erros (se existirem), serão testados para observar a robustez do módulo.



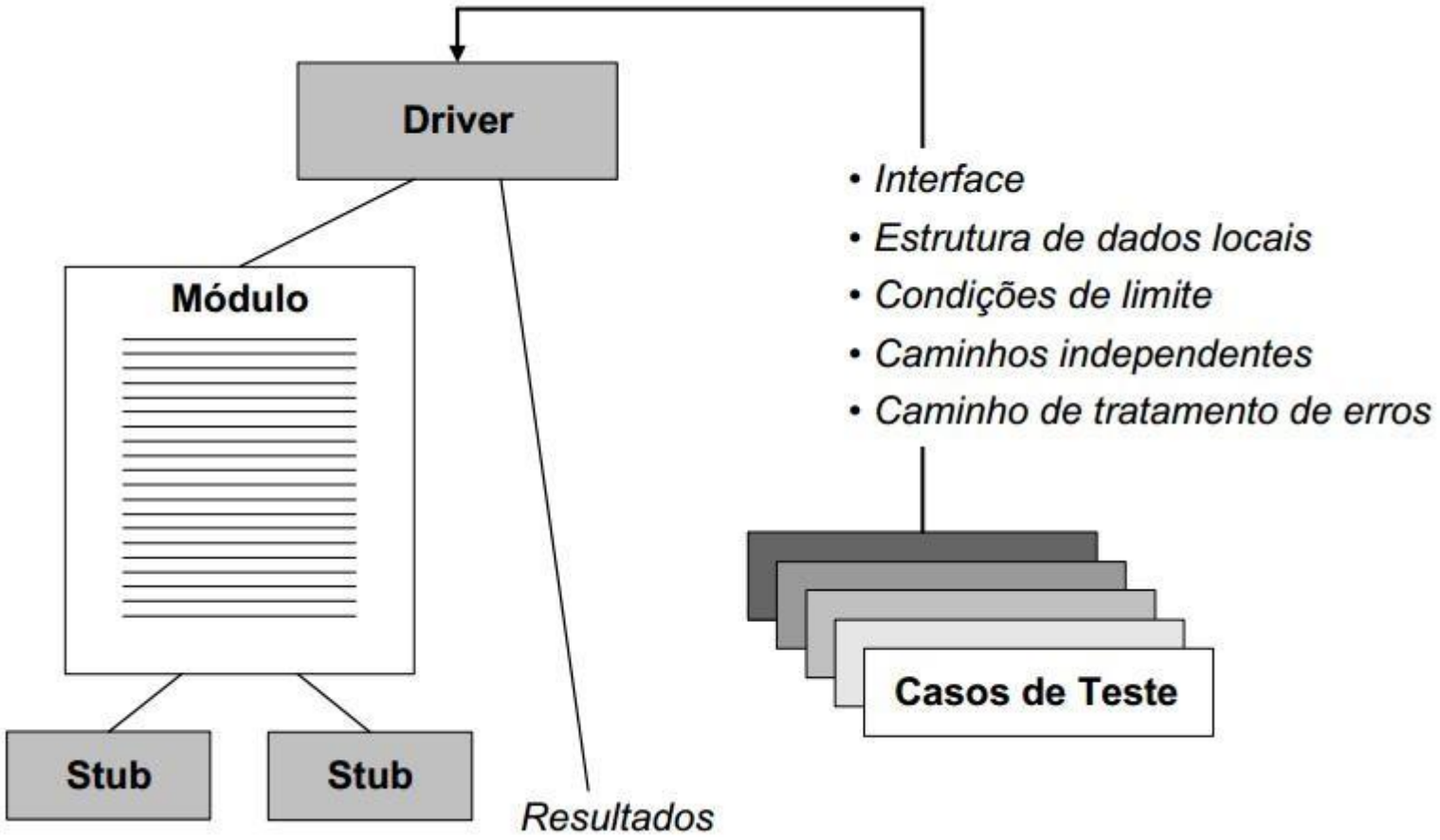
- **Teste de Integração**
- O Teste de Integração busca **erros surgidos quando fazemos a integração das diferentes unidades e componentes do software.**
- É importante lembrar que, o fato de se ter analisado os módulos do software de forma exaustiva (através de procedimentos de teste de unidade), **não há nenhuma garantia** de que estes, uma vez colocados em conjunto para funcionar, não apresentarão anomalias de comportamento.



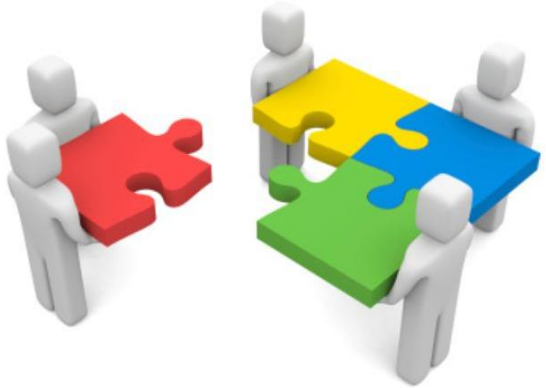
- **Teste de Integração**
- Uma das maiores causas de erros encontrados durante o teste de integração são os chamados **erros de interface**, devido principalmente, às incompatibilidades de interface entre módulos que deverão trabalhar de forma cooperativa.

Tipos de Teste

- **Teste de Integração**



- **Teste de Validação**
- Ao final do teste de integração, o software é finalmente estruturado na forma de um pacote ou sistema.
- A forma mais simples de definição do teste de validação é a **verificação de que o software como um todo cumpre corretamente a função para a qual ele foi especificado.**



Tipos de Teste

- **Teste de Validação**
- É importante lembrar que, no **documento de casos de uso do software,** são **definidos os chamados critérios de validação,** que servirão de guia para o julgamento de aprovação ou não do software nesta etapa de testes.

Documentação do Caso de uso: Realizar deposito.

Nome do Caso de Uso	Realizar Depósito
Caso de Uso Geral	
Ator Principal	Cliente
Atores Secundários	Funcionário
Resumo	Descreve os passos necessários para um cliente depositar algum valor em uma determinada conta
Pré-Condições	
Pós-Condições	
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Informar o número da conta	
	2. Verificar se a conta existe
3. Fornecer o valor a ser depositado	
	4. Executar caso de uso Registrar Movimento
Restrições/Validações	1. A conta precisa existir e estar ativa

Tipos de Teste

- **Teste de Validação**
- Além do esforço na validação de um software, na maioria das vezes é extremamente difícil para o desenvolvedor ou programador prever as diferentes maneiras como o software será utilizado.
- Quando um software é construído sob demanda, uma bateria de **testes de aceitação** é conduzida**, envolvendo o cliente.**

Tipos de Teste

- **Teste de Validação**
- No caso de um software de prateleira, (aquele software que é construído para um segmento do mercado de software e que é destinado à utilização por um grande conjunto grande de usuários), os testadores de software fazem uso de um processo denominado **teste Alfa** e **teste Beta**.

- **Teste de Validação: Teste Alfa**
- Em casos especiais de processos de desenvolvimento de software **–sistemas operacionais, sistemas gerenciadores de bancos de dados e outros softwares distribuídos em escala nacional e internacional** – os testes requerem fases também especiais antes do produto ser disponibilizado a todos os usuários.
- O período entre o término do desenvolvimento e a entrega é conhecido como **fase alfa** e os testes executados nesse período, como **testes alfa**.

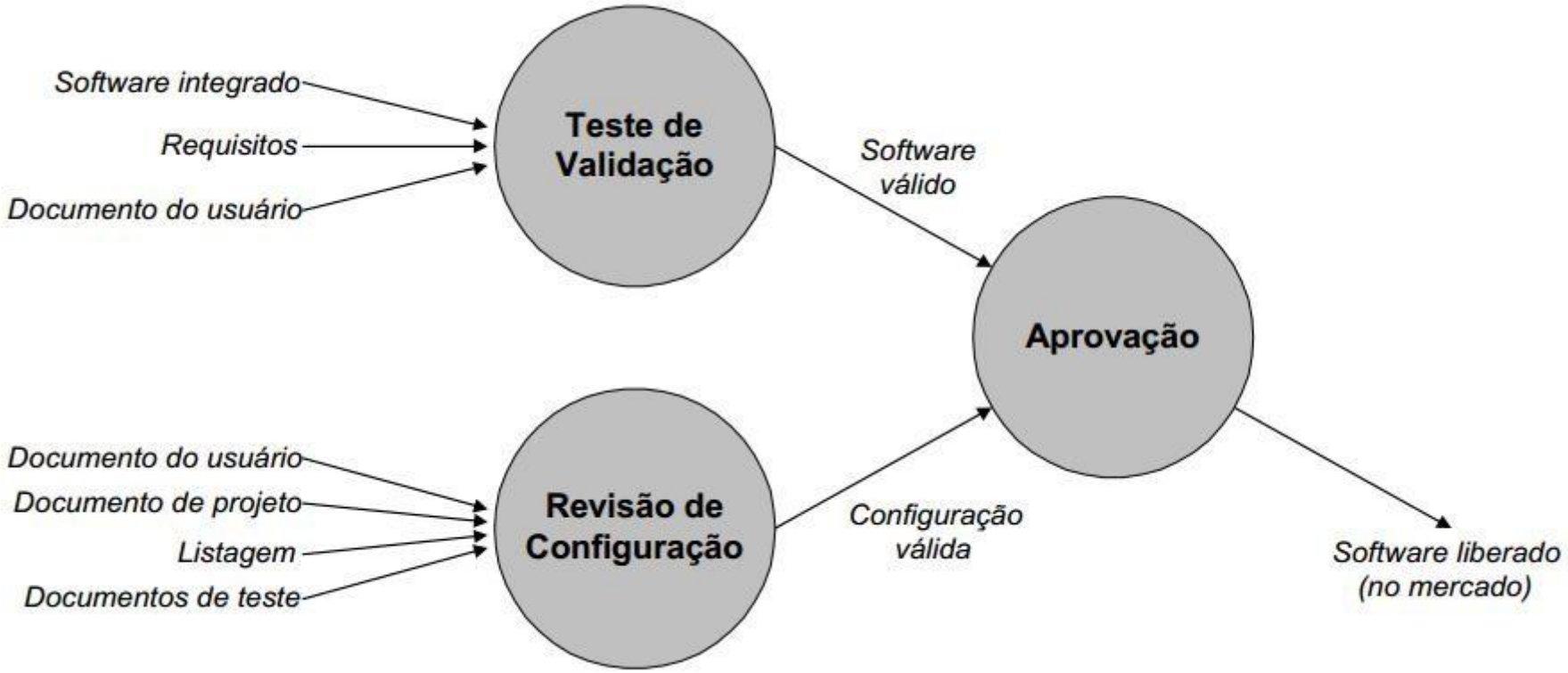
- **Teste de Validação: Teste Alfa**
- Os testes alfa são realizados com base no **envolvimento de um cliente**, **ou numa amostra de clientes**, sendo realizado nas instalações do desenvolvedor do software.
- O software é utilizado num ambiente natural, sendo que o desenvolvedor observa o comportamento do cliente e vai registrando as anomalias detectadas durante a utilização.

- **Teste de Validação: Teste Beta**
- Voltado para **softwares cuja distribuição atingirá grande número de usuários específicos de uma ou várias empresas compradoras**.
- O teste é conduzido em uma ou mais instalações do cliente, pelo usuário final do software. O **desenvolvedor geralmente não está presente**.
- Consequentemente, o **teste beta** é uma aplicação do software num ambiente que não pode ser controlado pelo desenvolvedor.

- **Teste de Validação: Teste Beta**
- O cliente registra todos os problemas (reais ou imaginários) que são encontrados durante o teste beta e os relata ao desenvolvedor em intervalos regulares.
- Com o resultado dos problemas relatados durante os testes beta, os engenheiros de software fazem modificações e depois se preparam para liberar o produto de software para toda a base de clientes.

Tipos de Teste

- **Teste de Validação**



- **Teste de Sistema**
- Neste tipo de teste, **são verificados os aspectos de funcionamento do software, integrado aos demais elementos do sistema**, como por exemplo o hardware e outros elementos.
- Dependendo do destino do software, é somente neste momento do teste, em que algunas partes do funcionamento do software podem ser efetivamente testados (particularmente, aqueles aspectos do funcionamento que dependem da interação dos demais elementos do sistema).

Tipos de Teste

- **Teste de Sistema**
- O teste de sistema inclui diversas modalidades de teste, cujo objetivo é testar o sistema computacional como um todo.
- Embora cada teste tenha uma finalidade diferente, o objetivo global acaba sendo atingido, uma vez que estes abrangem todos os elementos constituintes do sistema, verificando se estes foram adequadamente integrados.

Tipos de Teste

- **Teste de Sistema: Teste de Recuperação**
- Neste tipo de teste, o objetivo é **observar a capacidade do sistema para recuperar-se da ocorrência de falhas, num tempo previamente determinado.**
- Em alguns casos (cada vez mais frequentes), exige-se que o sistema seja tolerante a falhas, ou seja, que seja previsto retomar seu funcionamento mesmo no caso de ocorrência de algumas falhas.
- Neste tipo de teste,falhas são provocadas artificialmente (por uma técnica denominada injeção de falhas), para analisar a capacidade do sistema do ponto de vista da recuperação.

- **Teste de Sistema: Teste de Segurança**

- O teste de segurança visa **garantir que o sistema se comporte adequadamente diante as mais diversas tentativas ilegais de acesso.**
- Por exemplo, num sistema de informação acadêmica, um aluno deve ter autorização de acesso para a visualização das notas obtidas nas disciplinas que cursou (histórico escolar), mas não deve ter a possibilidade de alterar seus valores.
- No teste de segurança, o analista deve desempenhar um papel semelhante ao de um hacker, tentando contornar todos os mecanismos de segurança implementados no mesmo.

Tipos de Teste

- **Teste de Sistema: teste de Estresse**
- O teste de estresse, como o nome indica, **consiste em verificar como o sistema vai se comportar em situações limite.**
- Este limite pode ser verificado sob diferentes pontos de vista, dependendo das características do software.
- Alguns aspectos que podem ser observados neste contexto são, por exemplo:

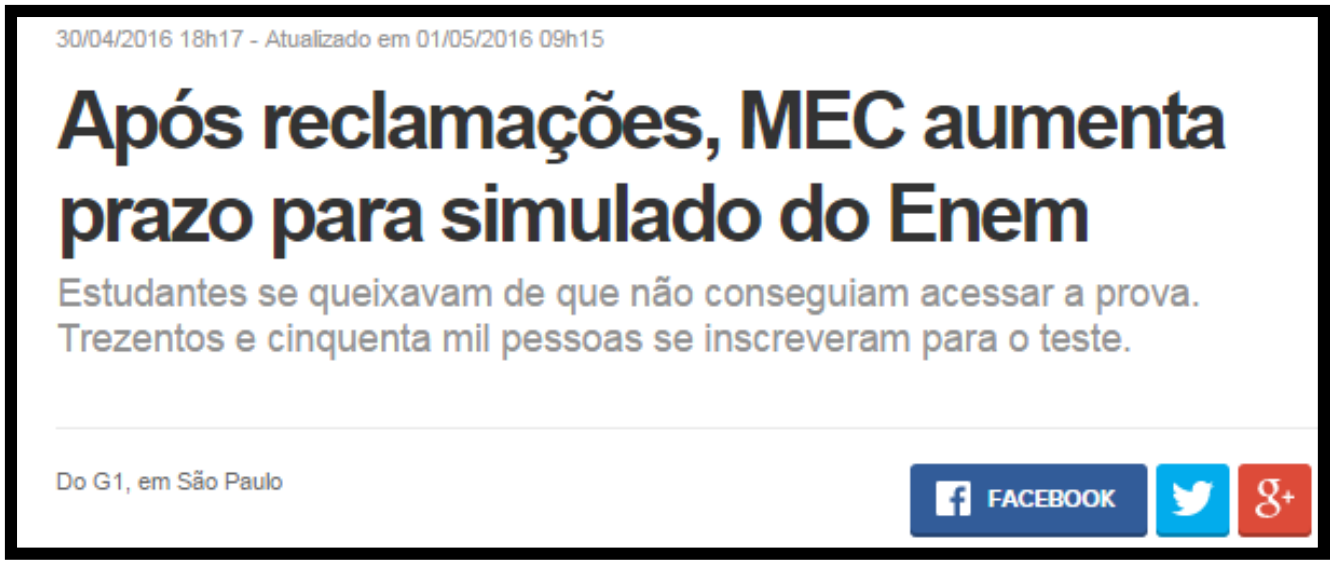
a) limite em termos de quantidade de usuários conectados a um determinado sistema servidor;

b) quantidade de utilização de memória e processador

Tipos de Teste

Teste de Estresse

Quantos usuários simultâneos o sistema pode atender sem capotar?



30/04/2016 18h17 - Atualizado em 01/05/2016 09h15

Após reclamações, MEC aumenta prazo para simulado do Enem

Estudantes se queixavam de que não conseguiam acessar a prova. Trezentos e cinquenta mil pessoas se inscreveram para o teste.

Do G1, em São Paulo

FACEBOOK

Guia do Estudante PLAY Curso ENEM ASSINE AGORA

Enem

Enem 2016 registrou 9,2 milhões de inscritos

Número é superior a 2015, com 8,4 milhões; inscrição pode ser paga até dia 25

Últimas

Enem

Dicas do que fazer dias antes da prova do Enem

1 nov 2018, 11h41

Tipos de Teste

- **Teste de Sistema: teste de Desempenho**
- Este tipo de teste consiste em **verificar se os requisitos de desempenho estão sendo atendidos para o sistema como um todo.**
- Neste aspecto, o não atendimento a um requisito de tempo pode afetar de forma irreversível a função do sistema e, no caso de alguns sistemas (os chamados sistemas críticos), a não realização desta função ou o não atendimento a estes requisitos temporais pode resultar em prejuízos catastróficos (risco a vidas humanas ou grandes prejuízos financeiros).

Projeto de Teste de Software

- A realização, com sucesso, da etapa de teste de um software deve ter, como ponto de partida, uma atividade de projeto dos casos de teste deste software.
- Projetar casos de teste para um software pode ser uma atividade tão complexa quanto a de projeto do próprio software, mas ela é necessária como única forma de conduzir, de forma eficiente e eficaz, o processo de teste.

Casos de Teste

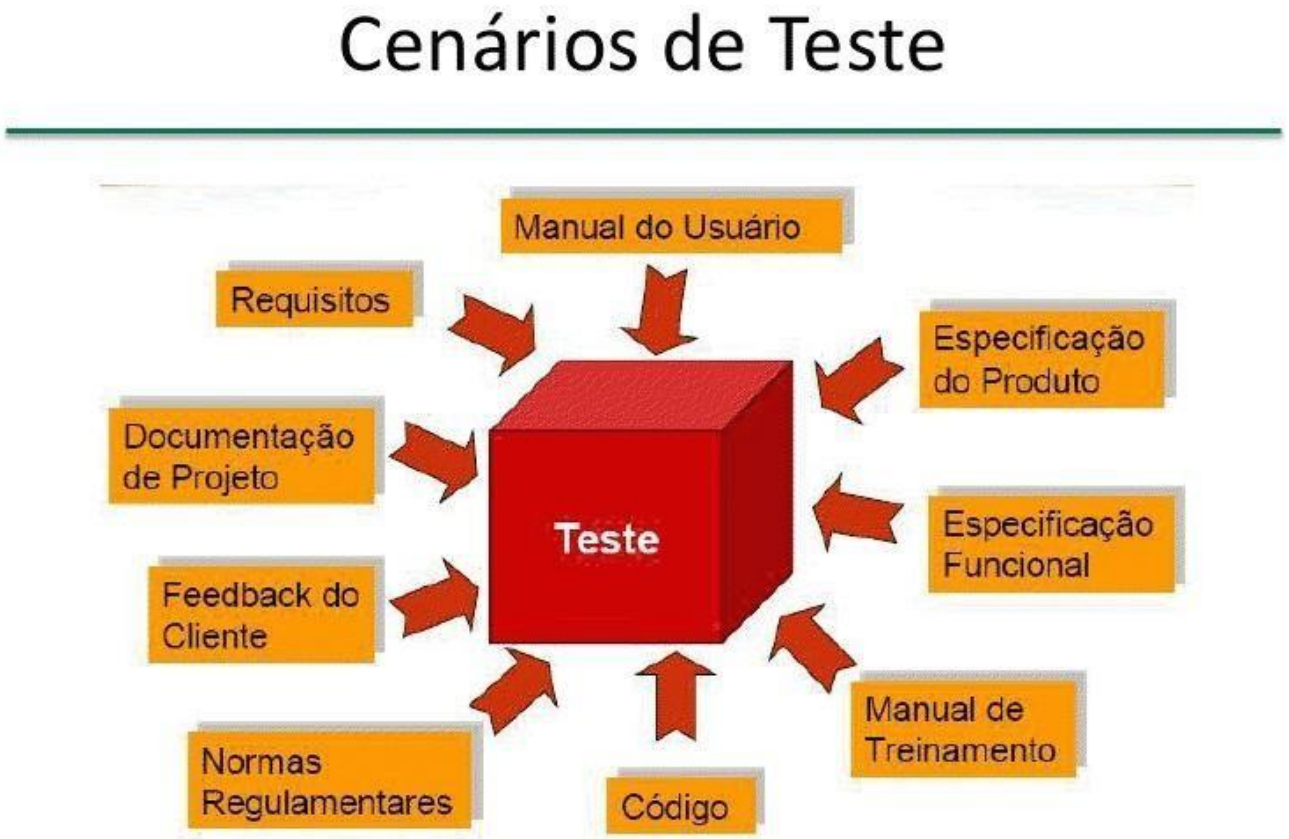
- Em engenharia de software, caso de teste é um conjunto de condições usadas para teste de software.
- Ele pode ser elaborado para identificar defeitos na estrutura interna do software por meio de situações que exercitem adequadamente todas as estruturas utilizadas na codificação.

Casos de Teste

- Para elaborar os casos de teste a partir do requisitos especificados deve-se considerar se considerar os seguintes itens:

1. Identificar todos os cenários contidos nas especificação existente;
2. Para cada cenário, identificar um ou mais casos de teste;
3. Para cada caso de teste, identificar condições de execução;
4. Adicionar os dados para as condições nos casos de teste.

Casos de Teste



Casos de Teste

Um bom começo seria **escrever os casos de testes de cada uma das funcionalidades ou requisito do software.**

Nada proporciona satisfação maior ao usuário final, com relação ao software, do que uma visão clara de suas expectativas, para que sejam verificadas e validadas.

Casos de Teste

Os casos de teste refletem os requisitos que devem ser verificados.
A verificação desses requisitos pode ser feita de forma diferente e por testadores distintos.
Como talvez você não consiga verificar todos os requisitos, nem seja responsável por isso, é essencial para o sucesso do projeto **selecionar os requisitos mais apropriados e importantes para o teste.**
Os requisitos escolhidos para verificação **representarão um equilíbrio entre o custo, o risco e a necessidade de verificá-los**.

Projetar Testes

O **projeto dos testes** contempla a criação dos casos de teste e demais artefatos necessários às atividades de execução dos testes conforme definido no **Plano de Teste**.

Projetar Testes

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campinas

Plano de Teste.

É o documento responsável por apresentar o planejamento para execução dos testes do software em desenvolvimento, incluindo a abrangência, abordagem, recursos e cronograma das atividades de teste.

Casos de Teste

É necessário desenvolver casos de teste para cada cenário de caso de uso.

Os cenários de caso de uso são identificados através da descrição dos caminhos que percorrer o fluxo básico e os fluxos alternativos, do início ao fim, através do caso de uso.

Casos de Teste

No diagrama a seguir, por exemplo, os vários caminhos através de um caso de uso, que refletem o fluxo básico e os fluxos alternativos, são representados com as setas.

O fluxo básico, representado pela linha preta e reta é o caminho mais simples através do caso de uso. Cada fluxo alternativo começa no fluxo básico e, depois, de acordo com uma condição específica, é executado.

Os fluxos alternativos podem retornar ao fluxo básico (fluxos alternativos 1 e 3), podem originar-se de outro fluxo alternativo (fluxo alternativo 2), ou podem terminar o caso de uso sem retornar a um fluxo (fluxos alternativos 2 e 4).

Exemplos de Fluxos de Eventos em um caso de uso

Cenários de Casos de usos

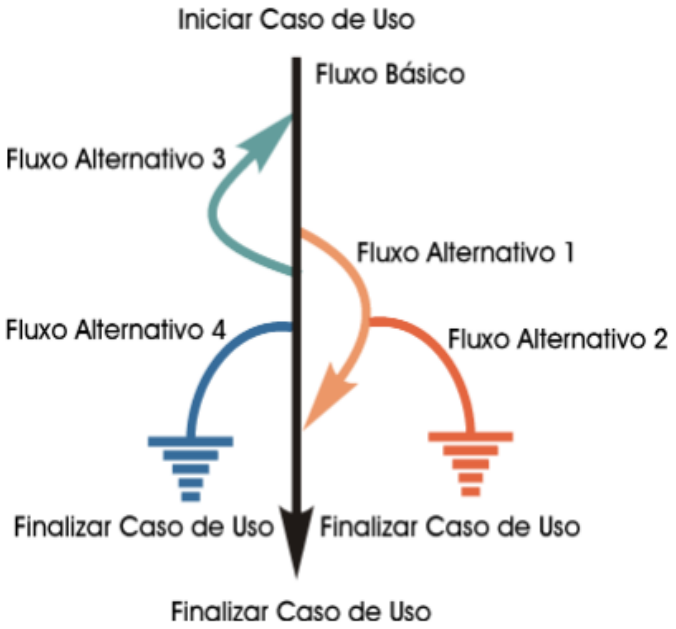
Após cada caminho possível através do caso de uso mostrado no diagrama, é possível identificar os diversos cenários de caso de uso.
Começando pelo fluxo básico e depois combinando esse fluxo com os fluxos alternativos, é possível identificar os seguintes cenários de caso de uso:

Exemplos de Fluxos de Eventos em um caso de uso

Casos de Teste

Cenários de Casos de usos

Cenário 1	Fluxo Básico			
Cenário 2	Fluxo Básico	Fluxo alternativo 1		
Cenário 3	Fluxo Básico	Fluxo alternativo 1	Fluxo alternativo 2	
Cenário 4	Fluxo Básico	Fluxo alternativo 3		
Cenário 5	Fluxo Básico	Fluxo alternativo 3	Fluxo alternativo 1	
Cenário 6	Fluxo Básico	Fluxo alternativo 3	Fluxo alternativo 1	Fluxo alternativo 2
Cenário 7	Fluxo Básico	Fluxo alternativo 4		
Cenário 8	Fluxo Básico	Fluxo alternativo 3	Fluxo alternativo 4	



Exemplos de Fluxos de Eventos em um caso de uso

OBSERVAÇÃO:
Por praticidade, os Cenários 5, 6 e 8 apenas descrevem uma única execução do loop indicado pelo Fluxo Alternativo 3.

Casos de Teste

É possível obter os casos de teste para cada cenário através da **identificação da condição específica que causará a execução desse cenário de caso de uso específico**.

Exemplos de Fluxos de Eventos em um caso de uso

Casos de Teste

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campinas

Exemplo de criação de Casos de teste baseado em Casos de uso.

[UC07]	
Nome:	Cadastrar usuário
Atores:	Vendedor
Prioridade:	Essencial
Requisitos associados:	• [RF02] Cadastro de usuários
Entradas:	• Nome, CPF, Data de Nascimento • Login (exceto para funcionário) • Senha (exceto para funcionário) • Crédito, data inicial, final, permissão para comprar em dinheiro, dias da semana (apenas funcionário)
Pré-condições:	• O vendedor deve estar logado no sistema • O usuário não deverá estar cadastrado no sistema
Pós-condições:	• O usuário deve estar cadastrado no banco de dados
Fluxos de eventos	
Fluxo Normal:	1. O usuário seleciona a opção "Cadastrar Usuário". [Fluxo Excepcional 1] 2. O usuário seleciona se o cadastro a ser feito é de funcionário, gerente ou vendedor. 3. O usuário fornece os dados da pessoa a ser cadastrada. 4. O sistema verifica se todas as informações obrigatórias foram fornecidas. [Fluxo Excepcional 2] 5. O sistema valida as informações fornecidas. 6. O sistema mostra uma tela para confirmação do cadastro. 7. O usuário confirma o cadastro. 8. Os dados são armazenados na base de dados.
Fluxo Excepcional 1:	1. A opção "Voltar" é selecionada. 2. Todos os dados informados são descartados. 3. A tela inicial do sistema é mostrada.
Fluxo Excepcional 2:	1. Uma ou várias informações obrigatórias não são informadas ou o formato da informação não é o adequado. 2. O sistema mostra uma tela informando que falta uma ou mais informações obrigatórias. 3. O sistema retorna para a tela "Cadastrar Pessoa" com os dados informados nos seus respectivos lugares.

Casos de Teste

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campinas

Exemplo de criação de Casos de teste baseado em Casos de uso.

ID	CT_001	Tipo do teste	Funcional
Objetivo	Verificar se a funcionalidade de cadastrar usuário está executando corretamente.		
Requisito	[RF02] Cadastro de usuários		
Pré-condição	1. O vendedor deve estar logado no sistema 2. O usuário não deverá estar cadastrado no sistema		
Nº. do Passo	**Passos**	**Resultados Esperados**	
1	Selecione a opção "Cadastrar Usuário".	A tela de cadastro é apresentada.	
2	Forneça os dados do usuário (nome = nome1, CPF = 032.165.987-52, data de nascimento = 23/03/1985, tipo de usuário = gerente) e selecione o botão "Continuar".	A tela para cadastrar login e senha é apresentada.	
3	Forneça os valores de login e senha (login = login1 e senha = senha1) e selecione o botão "Cadastrar".	O sistema retorna uma mensagem informando que o usuário foi cadastrado com sucesso.	
4	Verifique se o usuário está cadastrado no banco de dados com todas as informações fornecidas.	O usuário foi cadastrado no banco de dados com sucesso.	

Engenharia de software

- **Bibliografia**