

Teste de Sistemas



Módulo 1 – O que é teste de software



Fundamentos do Teste de Software

AGENDA

- Por que é necessário testar
- **O que é teste de software**
- Princípios gerais do teste
- Fundamentos do processo de teste
- A psicologia do teste
- Informações complementares

O que é teste de software

O que é teste de software?

Quando questionadas, as pessoas geralmente pensam que o teste somente consiste na execução do software em busca de defeitos (exemplo: executar o software para verificar seu funcionamento). Na realidade, a execução do teste é tão somente uma parte do teste, mas não representa todas as atividades.



Veja abaixo algumas definições para teste:

"Testar é analisar um programa com a intenção de descobrir erros e defeitos."

(Glenford Myers)

Teste (test) é "o processo que consiste em todas as atividades do ciclo de vida, tanto estáticas quanto dinâmicas, voltadas para o planejamento, preparação e avaliação de produtos de software e produtos de trabalho relacionados a fim de determinar se elas satisfazem os requisitos especificados e demonstrar que estão aptas para sua finalidade e para a detecção de defeitos."

(Glossário de termos do ISTQB)

O que é teste de software

Abordagem tradicional dos testes

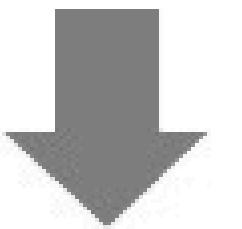
A abordagem tradicional dos testes consiste em mostrar que o sistema:

- Faz o que deve fazer
- Não faz o que não deve fazer



META: Mostrar que o sistema funciona

CRITÉRIO DE SUCESSO: Sistema em funcionamento correto



RESULTADO: Sistema com defeitos

O que é teste de software

A melhor abordagem para os testes

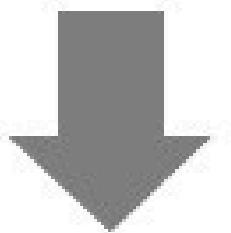
A melhor abordagem é seguir o conceito definido por Myers, o qual estabelece que devemos mostrar que o sistema não funciona:

- Faz o que não deve fazer
- Não faz o que deve fazer



META: Encontrar defeitos

CRITÉRIO DE SUCESSO: Sistema com falhas



RESULTADO: Sistema com menos defeitos

O que é teste de software

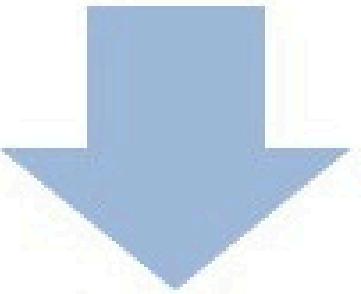
O paradoxo do teste

A proposta do teste como vimos é encontrar defeitos. Porém, ao encontrarmos defeitos a confiança do software é “destruída”. E, obviamente, não se pode ter confiança em algo que não está funcionando. Então, podemos identificar duas propostas do teste:

1. Proposta do teste: “destruir” a confiança do software.
2. Proposta do teste: estabelecer a confiança.

**Temos um
paradoxo**

Conclusão



**A melhor maneira de construir a confiança
é tentar destruí-la.**

O que é teste de software

Atividades do teste

Se pudéssemos delimitar exatamente as fronteiras das atividades do teste, teríamos dois grandes conjuntos de atividades:

Teste de software

Teste Dinâmico

(Dynamic Testing)

- Inclui atividades de planejamento, execução e controle do teste e necessita que o software seja executado.
- É o mais utilizado pelas empresas de desenvolvimento de softwares.
- O custo das correções dos defeitos tende a ser mais alto.

Teste Estático

(Static Testing)

- Inclui atividades de revisão, inspeção e análise estática do código.
- Revisão e inspeção podem ser realizadas em qualquer documentação do projeto e também no código-fonte durante a etapa de construção do software.

O que é teste de software

Atividades do teste

Visão geral das atividade do teste dinâmico:

Planejamento e controle	Os testes devem ser planejados e geridos como projetos. No planejamento, os objetivos, escopo, cronograma, recursos, estimativas, critérios de conclusão são definidos.
Seleção das condições de teste (itens de teste)	Define o que testar (cenários e condições).
Modelagem dos casos de teste	Casos de teste são criados para cada condição de teste.
Execução do teste	Executa os casos de teste.
Verificação dos resultados	Identifica o que deu certo e errado.
Avaliação dos critérios de conclusão	Avalia se os objetivos do teste foram alcançados.
Relatar sobre o processo de teste e o sistema sob teste	Relata o ocorrido.
Atividades de encerramento	Encerramento do projeto de teste e emissão de relatórios gerenciais.

Testes também incluem revisões de documentos (incluindo código-fonte) e análise estática (teste estático).

O que é teste de software

Objetivos do teste

- Testes podem possuir objetivos diferentes:
 - Encontrar defeitos
 - Prevenir defeitos
 - Ganhar confiança sobre o nível de qualidade e prover informações
- No processo de teste, diferentes pontos de vista levam a diferentes objetivos:

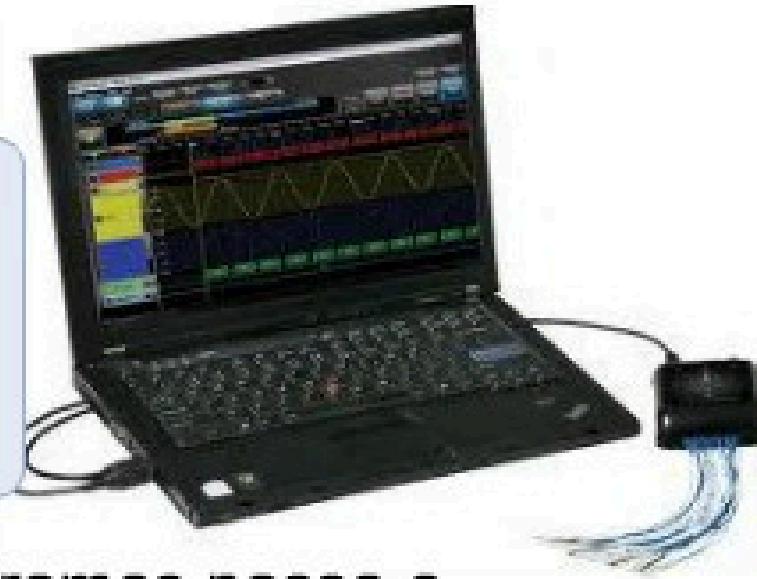


Teste durante o desenvolvimento (teste de unitário, integração e de sistemas)	Causar o maior número de falhas possíveis, de modo que os defeitos no software possam ser identificados e resolvidos.
Teste de aceite (ou aceitação)	Confirmar se o sistema está funcionando conforme o esperado para dar confiabilidade de que o software atingiu os requisitos.
Teste de manutenção	Verificar se não foram inseridos erros durante o desenvolvimento de mudanças.
Durante o teste operacional	Avaliar as características do software, como confiabilidade ou disponibilidade

O que é teste de software

Teste x Depuração (debugging)

Depuração (debugging): é uma ferramenta utilizada por programadores para reproduzir falhas, investigar o estado dos programas e encontrar o defeito correspondente.



- A depuração possibilita aos programadores executar os programas passo a passo, parar um programa em qualquer que seja a sentença, estabelecer e examinar variáveis de programas.
- Os testes são os responsáveis por encontrar as falhas, já a depuração é a atividade responsável por encontrar o defeito que está causando a falha e eliminá-lo.
- Existem falhas que podem exigir um bom tempo de depuração até que o programador encontre o defeito que a esteja causando.
- As responsabilidades de cada atividade são distintas:
 - Os desenvolvedores são os responsáveis por depurar o software
 - Os testes são executados principalmente pelos analistas de testes, mas podem ser executados por programadores também



Fundamentos do Teste de Software

AGENDA

- Por que é necessário testar
- O que é teste de software
- **Princípios gerais do teste**
- Fundamentos do processo de teste
- A psicologia do teste
- Informações complementares

Princípios gerais do teste de software

Os sete princípios-chave:

1 Teste demonstra a presença de defeitos

(Testing shows presence of defects)

2 Teste exaustivo é impossível

(Exhaustive testing is impossible)

3 Teste antecipado

(Early testing)

4 Agrupamento de defeitos

(Defect clustering)

5 Paradoxo do pesticida

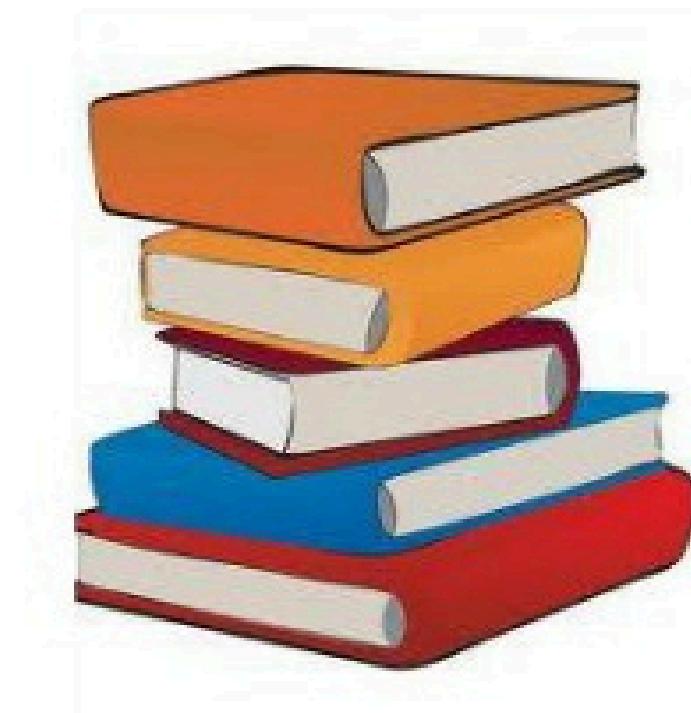
(Pesticide paradox)

6 Teste depende do contexto

(Testing is context dependent)

7 A ilusão da ausência de erros

(Absence-of-errors fallacy)



Estes 7 princípios
estão descritos em
detalhes na página
14 do [syllabus do
BSTQ](#).



Fundamentos do Teste de Software

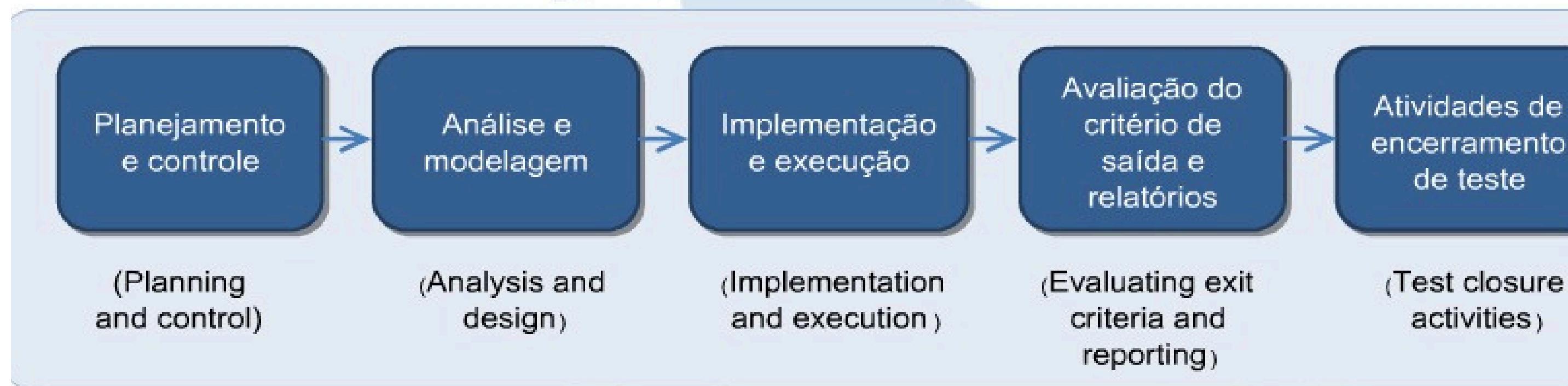
AGENDA

- Por que é necessário testar
- O que é teste de software
- Princípios gerais do teste
- **Fundamentos do processo de teste**
- A psicologia do teste
- Informações complementares

Fundamentos do processo de teste

Atividades do processo de teste

O teste é um processo executado paralelamente com o desenvolvimento de software, que consiste nas cinco seguintes etapas:



- Cada uma destas etapas pode ser executada uma ou várias vezes para atingir os objetivos esperados.
- Estas etapas podem se sobrepor ou serem executadas paralelamente.
 - Exemplo: o controle do teste pode ser realizado paralelamente a todas as etapas do processo de teste
- Este processo pode ser utilizado para as atividades do teste dinâmico e estático.

Planejamento do teste (Test planning)

- Durante esta etapa, devemos entender quais são as metas e objetivos do projeto e do cliente, quais são os riscos do projeto e, principalmente, qual é o escopo do projeto (quais funcionalidades do software deveremos testar e quais atividades deveremos realizar para testar tais funcionalidades).
- É nesta etapa que criamos o plano de teste (test plan):
 - “Documento que descreve o escopo, a abordagem (estratégia), os recursos e o cronograma das atividades de testes.”
 - Documenta qualquer exceção para a abordagem de teste. Por exemplo, somente uma técnica de modelagem de caso de teste foi usada para esta área funcional porque ela é menos crítica.
 - Considera todos os recursos (humanos ou não) necessários para a correta execução dos testes.
 - Contém o cronograma do projeto de teste.



Objetivos principais:

- Verificar a missão
- Definir os objetivos do teste
- Especificar as atividades de teste necessárias para atingir a missão e os objetivos

**MODELO DE PLANO DE TESTES NA PASTA
TESTE DE SOFTWARE
NA REDE SENAI**

Planejamento do teste (continuação)

Principais atividades desta etapa:

- Identificar os objetivos do teste
- Determinar o escopo
- Determinar a abordagem do teste (técnicas, cobertura, identificar e interagir com as equipes envolvidas no teste, testware)
- Determinar os recursos requeridos do teste
- Implementar a política de teste e/ou a estratégia de teste
- Agendar atividades de análise e modelagem de teste
- Agendar implementação, execução e avaliação do teste
- Determinar os critérios de finalização (saída)



Cronograma do teste

Fundamentos do processo de teste

Controle do teste (Test control)

- É a constante atividade de comparar o progresso atual contra o que foi planejado, reportando o status e os desvios do plano.
 - Envolve a tomada de ações necessárias (ações corretivas) para alcançar a missão e objetivos do projeto
 - O teste deve ser monitorado durante todo o projeto
- O planejamento do teste leva em consideração o retorno de informações das atividades de monitoração e controle.

Principais atividades desta etapa são:

- Medir e analisar os resultados
(o teste deve ser executado de acordo com o planejado).
- Monitorar e documentar progresso, cobertura do teste e critérios de saída.
- Iniciar ações corretivas.
- Tomar decisões.



O planejamento e controle do teste são atividades de gestão e normalmente são de responsabilidade do gerente do projeto de teste. Em projetos menores, o gerente do projeto de desenvolvimento pode ser o responsável por estas atividades.

Fundamentos do processo de teste

Análise e modelagem do teste

- Após o planejamento das atividades é hora de produzir as condições de teste.

Condições de teste: “item ou evento de um componente ou sistema que pode ser verificado por um ou mais casos de teste. Por exemplo: função, transação, característica, atributo de qualidade ou elemento estrutural.”

- As condições de teste indicam “o quê” será testado e não “como”.
- A aplicação deve ser analisada em detalhes, tanto na perspectiva dos usuários como na perspectiva técnica.
- As condições de teste têm a missão de transformar os objetivos do teste em algo tangível (palpável).
- O principal objetivo da etapa de análise e modelagem do teste é exercitar as condições de uso do software de uma maneira eficiente, alcançando o máximo de cobertura com o mínimo de casos de teste.

Modelagem de teste: “especifica as condições de teste (cobertura de itens) para um item de teste (componente ou funcionalidade do software que deve ser testado), detalha a abordagem de teste e identifica os casos de teste de alto nível associados.”

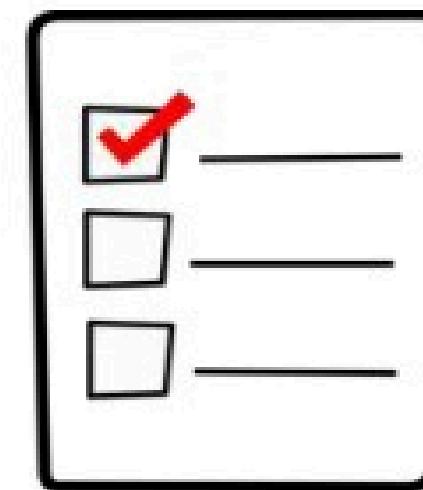
- As etapas de análise e modelagem de teste e implementação e execução do teste estão intimamente ligadas. Uma diz o que testar e a outro como.

Fundamentos do processo de teste

Análise e modelagem do teste

Principais atividades desta etapa são:

- Revisar a base de teste (todos os documentos a partir dos quais os requisitos de um determinado componente ou sistema podem ser inferidos; documentação na qual os casos de teste estão baseados).
 - Durante esta revisão procure por requisitos mal definidos, requisitos incompletos, requisitos ambíguos, etc.
- Identificar condições de teste e requisitos de dados de teste.
- Priorizar as condições de teste.
- Avaliar a testabilidade dos requisitos e do sistema.
 - Verifique se é possível testar os requisitos (exemplo de requisito testável: imprimir 100 notas fiscais em até 30 segundos)
- Modelar o ambiente de teste (set-up) e identificar qualquer requisito de infraestrutura e ferramentas.



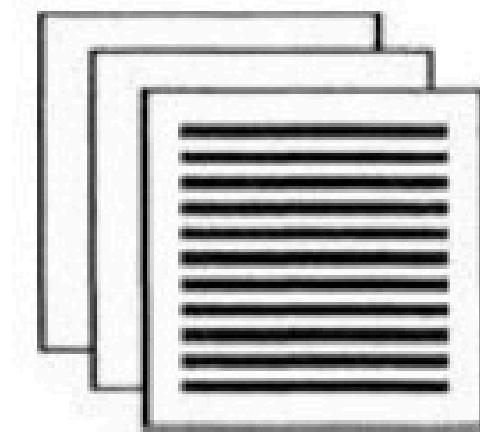
Devemos realizar esta etapa para ganharmos tempo na execução do teste e garantirmos que estamos testando aquilo que é realmente mais importante e mais crítico para o negócio



Fundamentos do processo de teste

Implementação e execução

- Durante esta etapa, transformaremos as condições de teste em casos de teste e configuraremos o ambiente para a execução e, posteriormente, executaremos os testes.
- Casos de teste especificam em detalhes como iremos testar uma determinada condição de teste.
 - Exemplo: para testar o cadastro de cliente, o caso de teste deverá conter todos os passos necessários para o cadastramento do cliente com o devido resultado esperado para cada ação ou conjunto de ações
 - Para cada condição de teste pode haver mais de um caso de teste.
 - Geralmente são 2 casos de teste: um positivo e um negativo
 - Um caso de teste sempre deve conter qual o resultado esperado após a sua execução
 - Uma vez que os casos de teste estiverem criados, podemos partir para a execução dos mesmos
 - Comece pelas condições de teste priorizadas



Implementação e execução

Principais atividades da implementação são:

- Criar os casos de teste
- Priorizar os casos de teste
- Criar dados de teste
- Escrever os procedimentos de teste
- Escrever os scripts automatizados de teste (se for o caso)
- Criar as suites de teste a partir dos casos de teste para uma eficiente execução
- Verificar se o ambiente de teste foi configurado corretamente



Fundamentos do processo de teste

Implementação e execução

Cada caso de teste é especificado contendo:

- Seu objetivo
- O estado inicial do software (pré-condições)
- Entradas (ou procedimentos)
- Resultados esperados

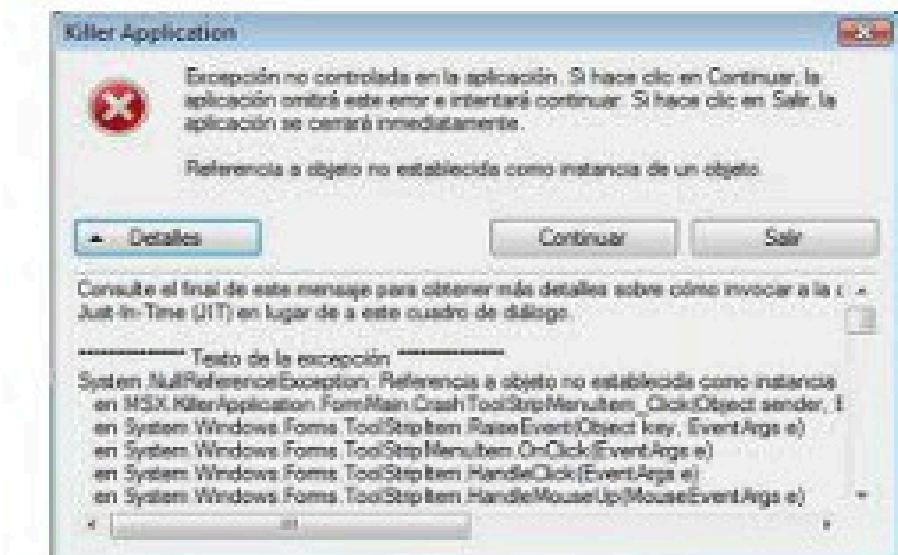
**MODELO DE DOCUMENTAÇÃO PARA CASOS DE
TESTE ENCONTRA-SE NA REDE INTERNA SENAI**

Fundamentos do processo de teste

Implementação e execução

Principais atividades da execução são:

- Executar os casos de teste de acordo com a sequência planejada
- Registrar as saídas da execução do teste
- Checar os resultados
- Analisar incidentes para estabelecer a sua causa raiz
- Repetir, quanto necessário, atividades de teste como resultado de ações tomadas para cada defeito
- O nível alcançado de cobertura do teste deve ser registrado para medir os critérios de saída



Fundamentos do processo de teste

Critérios de finalização (Exit criteria)

- São aplicados em todos os níveis do teste para determinar quando se deve parar o teste. Critério de saída de uma etapa é critério de entrada para outra (exemplo: o critério de finalização ou saída do teste unitário é um critério de entrada para o nível de teste integrado).
- Ajudam identificar se os objetivos do teste foram atingidos e se podemos encerrar as atividades.

Critérios de saída são “um conjunto de condições genéricas e específicas, acordadas pelos stakeholders/interessados, que permite que um processo seja oficialmente considerado completado.”

- Critérios de saída podem ser definidos em termos de:
 - Eficácia (por exemplo: cobertura ou requisitos)
 - Restrições de tempo ou orçamento
 - Porcentagem de teste executado sem gerar incidentes
 - Número de defeitos remanescentes
- Critérios de cobertura são definidos em termos de itens que são exercitados pelos casos de teste.

Fundamentos do processo de teste

Exemplos de critérios de finalização (ou de saída)

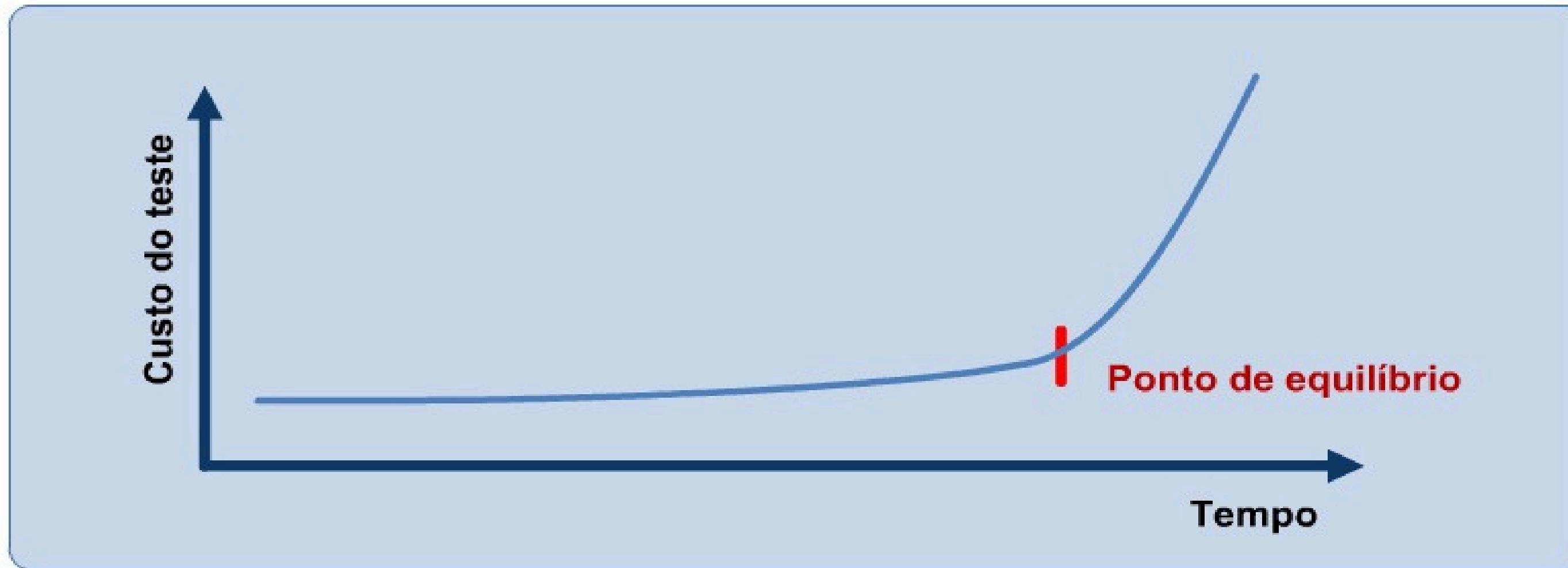
Outros critérios de finalização que você pode utilizar em seus projetos:

1. Todos os casos de teste executados
2. Todos os casos de teste aprovados
3. Sem incidentes para resolver
4. Sem incidentes sérios para resolver
5. Cobertura pré-definida alcançada:
 - Cobertura de código
 - Cobertura de funcionalidades
6. Confiabilidade (MTBF) requerida

Fundamentos do processo de teste

Quando parar de testar?

Veja no gráfico abaixo a curva que representa o custo do teste:



Um consenso entre os profissionais é que o momento certo para se encerrar os testes é onde a curva do custo do teste se intersecta com a curva do custo da falha para o negócio, gerando o que chamamos de ponto de equilíbrio. O ponto de equilíbrio normalmente acontece quando já identificamos cerca de 94% dos defeitos existentes no software. Para realizar estas medições nós precisamos de certa maturidade tanto nos processos de desenvolvimento como nos processos de teste.

Fundamentos do processo de teste

Avaliação dos critérios de saída e relatórios

A **Verificação de finalização** (**Checking for test completion**) envolve olhar para os critérios de finalização de teste previamente especificados no plano de teste para ver se foram atingidos. Se não foram atingidos, alguns certos testes podem ser repetidos e, em alguns casos, pode ser apropriado especificar alguns novos casos de teste para atingir um determinado objetivo – por exemplo: voltar para a especificação de desenhar mais testes a serem executados.

Fluxo das atividades do teste:



Fundamentos do processo de teste

Atividades de encerramento de teste

- Esta etapa trata das atividades necessárias para coletar dados do projeto de teste, visando extrair números para medir o desempenho das atividades, consolidar experiências, realizar reuniões de lições aprendidas e atualizar e armazenar o *testware*.
→ O *testware* é toda a documentação gerada pelo processo de teste, como por exemplo planos de teste, condições de teste, casos de teste, base de dados utilizada, etc. Todos esses documentos devem ser armazenados de tal forma que possam ser recuperados facilmente, caso seja necessário, para serem reutilizados em novos projetos que alterem requisitos do projeto original.

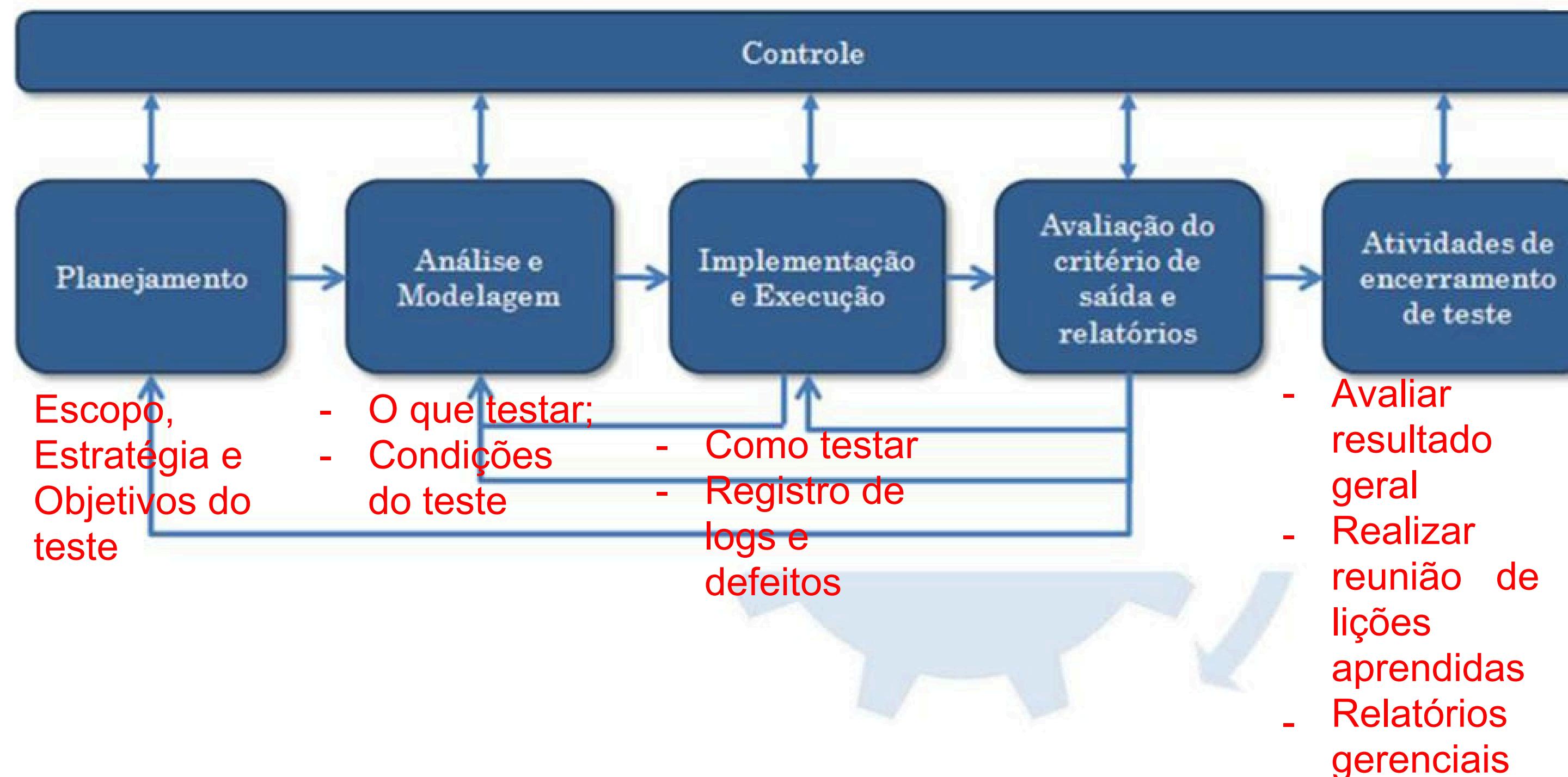
Principais atividades do encerramento são :

- Checar se o que foi planejado foi entregue (gera um relatório final).
- Analisar os defeitos encontrados (identifica causas raízes e elabora planos de ação).
- Finalizar e armazenar o *testware* para reutilização.
- Realizar reunião de lições aprendidas.



Fundamentos do processo de teste

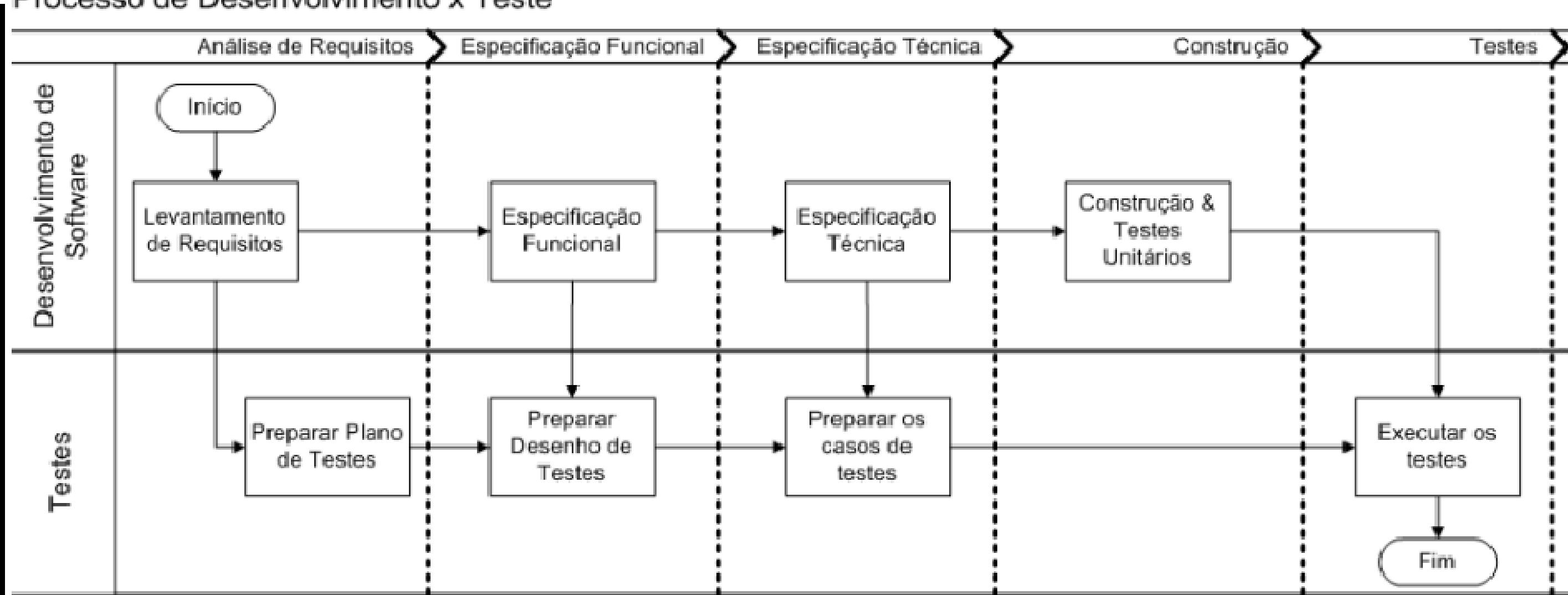
As cinco etapas/fases do processo de teste – Interações



Processo de desenvolvimento x Teste

O fluxo apresentado abaixo representa algumas atividades do desenvolvimento e de teste e seu relacionamento.

Processo de Desenvolvimento x Teste



Importante: o teste não é uma fase do ciclo de desenvolvimento de software, é parte de todas as fases.



Fundamentos do Teste de Software

AGENDA

- Por que é necessário testar
- O que é teste de software
- Princípios gerais do teste
- Fundamentos do processo de teste
- **A psicologia do teste**
- Informações complementares

A psicologia do teste

Psicologia do teste

- Antes:
 - Mostrar que o sistema atingiu seus requisitos
 - O objetivo primário de encontrar defeitos
- Agora:
 - Objetivo primário de encontrar defeitos
- Encontrar defeitos pode ser visto como uma crítica ao desenvolvedor.
- Mas o principal intuito é encontrar defeitos construtivos:
 - Poupando tempo
 - Reduzindo riscos
 - Reduzindo custos
 - Melhorando habilidades
 - Aumentando a produtividade



A psicologia do teste

Relacionamento entre desenvolvedores e testadores

- Os testes podem ser percebidos como um processo destrutivo pela equipe de desenvolvimento.
→ O processo de desenvolvimento é naturalmente criativo e construtivo
- Programadores e testadores veem o desenvolvimento de sistemas de perspectivas diferentes:



Programador

- Procura construir softwares novos baseados nas exigências dos usuários e, realmente, quer que eles funcionem.
- Trabalhará longas horas.
- Geralmente, altamente motivado e muito determinado a fazer um bom trabalho.

Testador

- Está mais preocupado em garantir que o usuário realmente tenha o sistema que quer, que seja confiável e não faça nada que não deve.
- Trabalhará longas horas procurando por defeitos no software.



- É muito importante que todos trabalhem juntos e que haja respeito mútuo.
- Colaboração é a abordagem certa – meta comum.
- Comunicar os defeitos objetivamente, não subjetivamente.

Características de bons analistas de teste

Um analista de teste necessita:

- Boas habilidades de comunicação
- Boas habilidades de observação
- Habilidades para lidar com pessoas
- Ser curioso
- Ser paciente
- Ser confiável
- Ser meticoloso
- Ser um investigador nato
- Ter muita atenção aos detalhes
- Ser criativo para identificar prováveis locais de falha



A psicologia do teste

Teste independente

Podemos testar nosso próprio trabalho? .

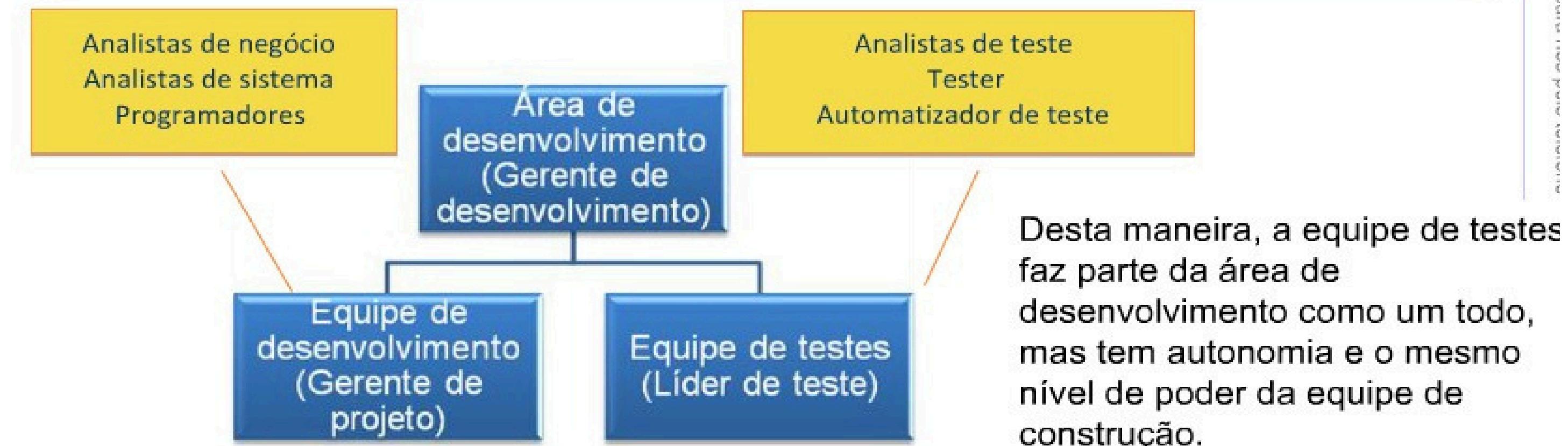
- Quando escrevemos um artigo ou um texto e repassamos pela primeira vez para alguém ler, é quase inevitável que erros sejam descobertos e, às vezes, estavam na nossa cara, mas nós não conseguíamos ver. Algumas razões para isto:
 - Fazemos suposições
 - Somos emocionalmente ligados ao produto
 - Estamos familiarizados com o produto e não vemos os defeitos óbvios
 - Vemos exatamente o que queremos ver
 - Temos um interesse velado em passar o trabalho como correto e não encontrar defeitos
 - Pesquisas apontam que encontramos apenas de 30% a 50% dos nossos próprios erros
- O pensamento certo poderia habilitar desenvolvedores para testar o código.
- As duas equipes possuem perspectivas diferentes: a equipe de desenvolvimento quer que o software funcione e a equipe de testes quer encontrar defeitos.
- O ideal é que tenhamos uma equipe separada da equipe de desenvolvimento para realização dos testes e que essa equipe tenha autonomia para realizar um trabalho isento, sem pressões e interferências do desenvolvimento.

A psicologia do teste

Teste independente

Os níveis de independência do teste podem ser menores ou maiores de acordo com a equipe utilizada:

- Menor**
- **Baixo**: testes são realizados pelo programador que escreveu o software.
 - **Médio**: testes são realizados por um programador diferente.
 - **Alto**: testes são realizados por uma pessoa/equipe não faz parte da equipe de desenvolvimento (equipe de testes ou fábrica de testes).
- Maior**
- **Utopia**: testes são gerados por ferramentas sem intervenção humana.





Fundamentos do Teste de Software

AGENDA

- Por que é necessário testar
- O que é teste de software
- Princípios gerais do teste
- Processo fundamental de teste
- A psicologia do teste
- **Informações complementares**

Informações complementares

Principais envolvidos nas atividades de teste



Gestor da qualidade

Responsável pelas ações de controle de qualidade dos produtos.



Líder do projeto de teste

Responsável por liderar e planejar o projeto de teste.



Arquiteto de teste

Responsável por montar e disponibilizar o ambiente de teste.



Analista de teste

Responsável pela modelagem e elaboração dos casos de teste e scripts de teste.



Testador (Tester)

Responsável pela execução dos casos de teste e pela documentação dos resultados.



Usuários

Responsáveis pela avaliação e aceitação do sistema.

Informações complementares

Principais organizações envolvidas

A disseminação dos processos de teste é relativamente nova no Brasil. Algumas organizações e institutos criados estão tendo a missão de profissionalizar a atividade no país.



CTFL – Certified Tester Foundation Level



CTFL – Certified Tester Foundation Level



CSTE – Certified Software Tester



CBTS – Certificação Brasileira de Teste de Software

Observação: clique sobre as logos para obter mais informações sobre as instituições e suas certificações.

Informações complementares

Normas e padrões

Algumas normas e padrões que são amplamente utilizados pelo comunidade de testes de software:

BS 7925-1

- Define um glossário de termos sobre teste de software

BS 7925-2

- Define os padrões para o processo de teste de software

IEEE 829

- Define normas e padrões para o processo de teste e qualidade do produto

ISO 9126-1

- Define as características da qualidade do software

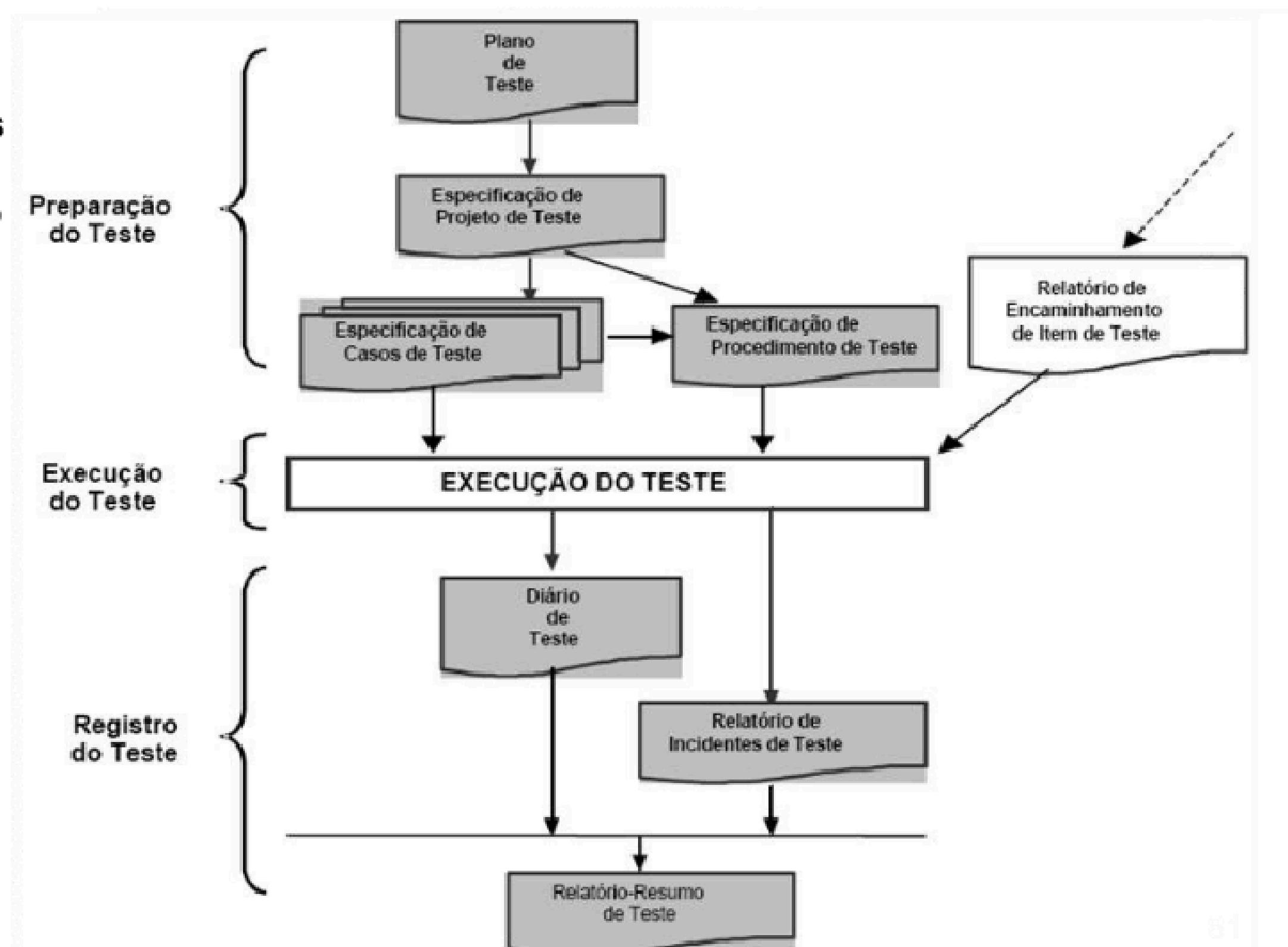
Informações complementares

A norma IEEE 829

A Norma IEEE 829 descreve um conjunto de documentos para as atividades de teste de software. Este conjunto de documentos é produzido, segundo a norma, durante três etapas:

- **Preparação do teste**
- **Execução do teste**
- **Registro do teste**

A figura abaixo apresenta os documentos que são gerados em cada uma das fases e o relacionamento entre eles.



Documentos da norma IEEE 829

	Plano de Teste	Define o planejamento para execução do teste, incluindo abrangência, abordagem, recursos e cronograma das atividades de teste. Identifica os itens e as funcionalidades a serem testados, as tarefas a serem realizadas e os riscos associados com a atividade de teste.
Especificações	Especificação de projeto de teste	Refina a abordagem apresentada no plano de teste e identifica as funcionalidades e características a serem testadas pelo projeto e por seus testes associados. Este documento também identifica os casos e os procedimentos de teste, se existirem, e apresenta os critérios de aprovação.
	Especificação de caso de teste	Define os casos de teste, incluindo dados de entrada, resultados esperados, ações e condições gerais para a execução do teste.
	Especificação de procedimento de teste	Especifica os passos para executar um conjunto de casos de teste.
Relatórios	Diário de teste	Apresenta registros cronológicos dos detalhes relevantes relacionados à execução dos testes.
	Relatório de incidente de teste	Documenta qualquer evento que ocorra durante a atividade de teste e que requeira análise posterior.
	Relatório-resumo de teste	Apresenta de forma resumida os resultados das atividades de teste associadas a uma ou mais especificações de projeto de teste e provê avaliações baseadas nesses resultados.
	Relatório de encaminhamento de item de teste	Identifica os itens encaminhados para teste no caso de equipes distintas serem responsáveis pelas tarefas de desenvolvimento e de teste.

Resumo e sugestões

De uma maneira geral, agora você deve:

- Reconhecer e usar a terminologia básica de teste
- Entender por que testes são necessários
- Definir erro, falha e defeito
- Explicar por que erros ocorrem
- Explicar o processo de teste de software e suas etapas
- Reconhecer as principais atividades de cada etapa do processo de teste
- Entender o que são teste dinâmico e teste estático
- Entender a diferença de formas de pensar entre programadores e testadores
- Entender por que você não pode testar seu próprio trabalho
- Reconhecer as principais normas e padrões aplicados ao teste de software

