

Formação em Testes Manuais e Automatizados

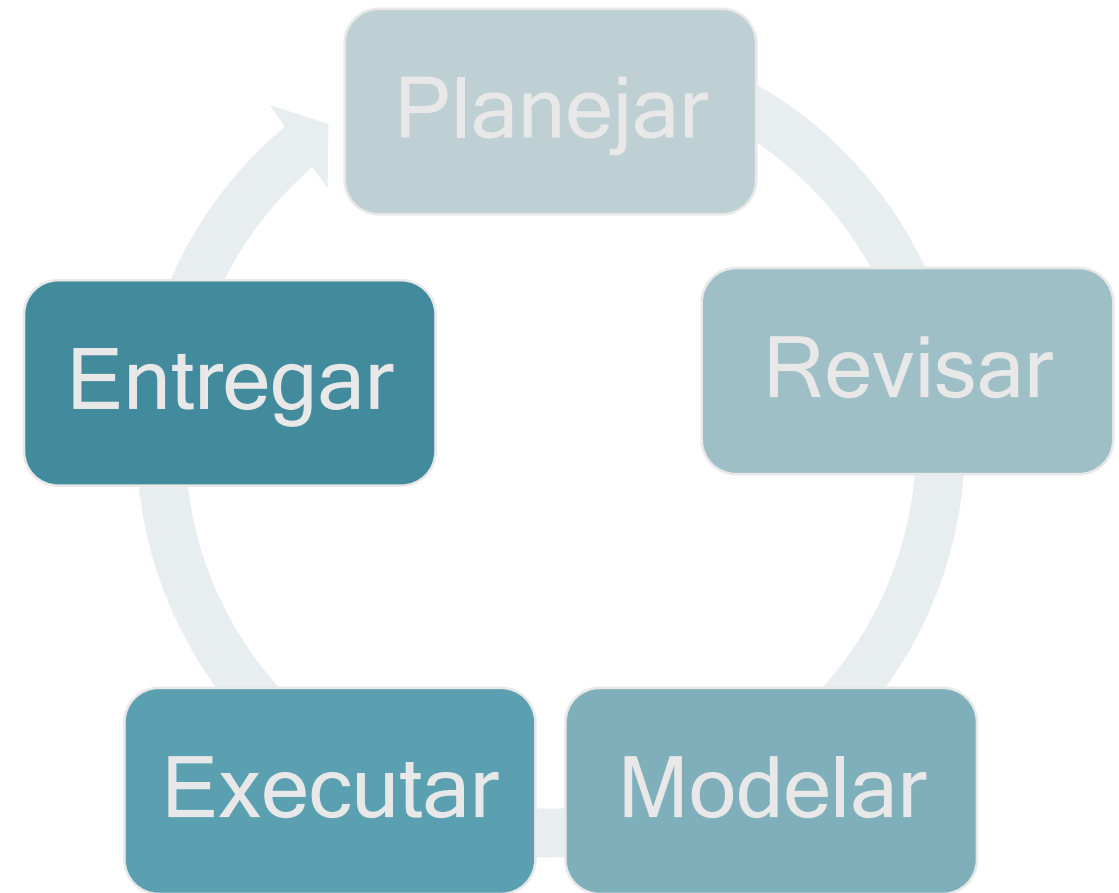
Minsait e Uniesp-PB

PROCESSO DE TESTES

6

6. Processo de Testes

Um Processo de Teste de Software tem como objetivo estruturar as etapas, as atividades, os artefatos, os papéis e as responsabilidades do teste, permitindo organização e controle de todo o ciclo do teste, minimizando os riscos e agregando valor ao software.



6.1 Planejamento dos Testes - Fatores Influenciadores

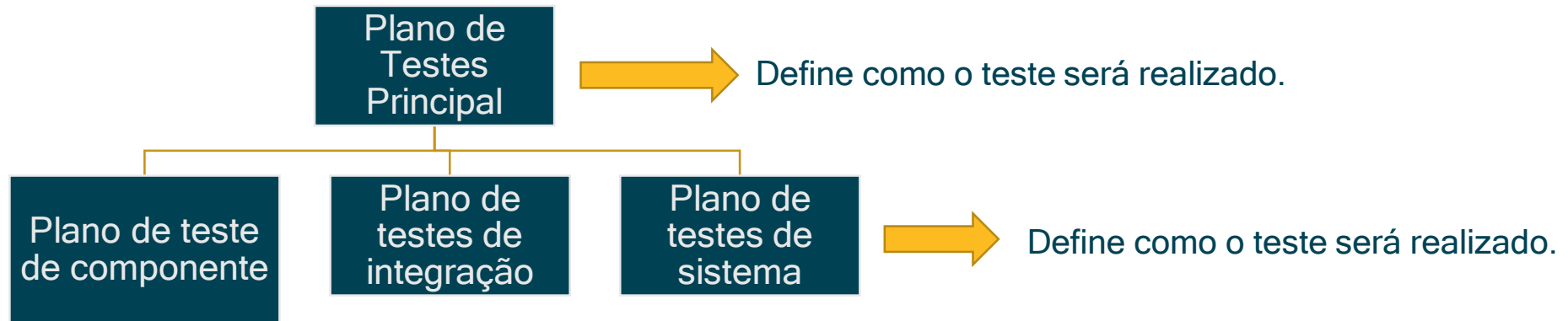


6.1 Planejamento dos Testes

PROJECT TEST PLAN		AUTHOR: A TESTER	
INTRODUCTION	IN SCOPE	OUT OF SCOPE	
ISSUES	RESOURCES	ENVIRONMENTS - TOOLS	
ASSUMPTIONS	TIMESCALES		

O Planejamento dos Testes:

- É uma atividade contínua e é executado durante todo o ciclo de vida do produto;
- Conforme o andamento do projeto e do planejamento, mais informações ficam disponíveis e mais detalhes podem ser incluídos no plano de testes (ou seja, o plano de testes é um artefato vivo);
- O planejamento pode ser documentado em um plano de testes principal e caso haja testes que ocorram em níveis diferentes (unitário/componente, integração, sistema, etc) poderão ter planos de testes separados para cada um destes níveis.



6.1 Planejamento dos Testes

O Planejamento dos Testes permite:

- A definição do escopo (ou seja, a identificação dos itens e funcionalidades que deverão ser testados);
- Quem serão os responsáveis?
- Quais serão os riscos?
- Quais serão as premissas?
- Quais serão os dados que deverão ser utilizados?
- Quais serão os ambientes?
- Quais serão os marcos ou cronogramas para as principais atividades de testes e etc.

PROJECT TEST PLAN		AUTHOR: A TESTER	
INTRODUCTION		IN SCOPE	OUT OF SCOPE
RISKS	ASSUMPTIONS	RESOURCE	ENVIRONMENTS + TOOLS
		TIMESCALES	

6.2 Revisão por Pares (Testes Estáticos)

PROJECT TEST PLAN		AUTHOR: A TESTER	
INTRODUCTION	IN SCOPE	OUT OF SCOPE	
ISSUES	RESOURCES	ENVIRONMENTS - TOOLS	
ASSUMPTIONS	TIMESCALES		

Os testes estáticos podem ser realizados por meio de:

- **Revisões** (formais ou informais): Através da avaliação manual dos produtos de trabalho, realizados por um ou mais indivíduos;
- **Análise Estática:** Avaliação do código ou outros produtos de trabalho orientados por ferramentas.

A fase de revisão por pares, dentro desta proposta de processo de testes, é realizada através de um teste estático, do tipo **Revisão**.

6.2 Revisão por Pares (Testes Estáticos)

PROJECT TEST PLAN		AUTHOR: A TESTER	
INTRODUCTION	IN SCOPE	OUT OF SCOPE	
RISKS	RESOURCES	ENVIRONMENTS - TOOLS	
ASSUMPTIONS	TIMESCALES		

Temos como exemplos de produtos de trabalho para testes estáticos (revisões ou análise estática):

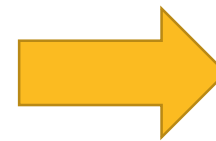
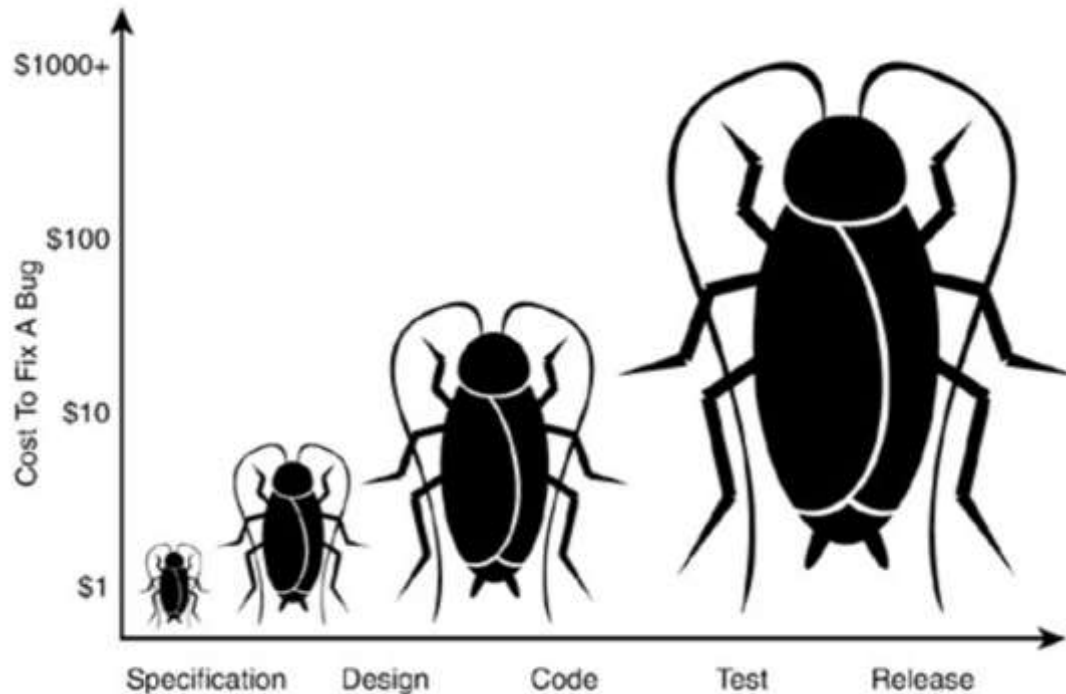
- Especificações (incluindo requisitos de negócio, requisitos funcionais e requisitos de segurança);
- Épicos, histórias de usuários e critérios de aceite;
- Especificações de arquitetura e design;
- Código;
- Testware (incluindo planos de teste, casos de teste, procedimentos de teste e scripts de teste automatizado);
- Guias de usuários;
- Páginas Web
- Contratos, planos de projeto, cronogramas e orçamentos.

PROJECT TEST PLAN		AUTHOR: A TESTER	
INTRODUCTION	IN SCOPE	OUT OF SCOPE	
ISSUES	RESOURCES	ENVIRONMENTS - TOOLS	
ASSUMPTIONS	TIMESCALES		

6.2 Revisão por Pares (Testes Estáticos)

O teste estático permite a detecção antecipada dos defeitos antes da realização dos testes dinâmicos (quando aplicado no início do ciclo de desenvolvimento).

Relembrando a regra 10 de Myers, os defeitos detectados precocementecostumam ser muito mais baratos de serem removidos.



Então, usar técnicas de testes estáticos para encontrar e corrigir defeitos rapidamente é quase sempre muito mais barato do que usar testes dinâmicos

6.2 Revisão por Pares (Testes Estáticos)

PROJECT TEST PLAN		AUTHOR: A TESTER	
INTRODUCTION	IN SCOPE	OUT OF SCOPE	
ISSUES	RESOURCES	ENVIRONMENTS - TOOLS	
ASSUMPTIONS	TIMESCALES		

Os testes estáticos e dinâmicos se complementam, encontrando diferentes tipos de defeitos:

Teste estático	Teste dinâmico
Encontra defeitos diretamente em produtos de trabalho	Identifica falhas causadas por defeitos quando o software é executado.
Pode ser utilizado para melhorar a consistência e a qualidade interna dos produtos de trabalho	Geralmente se concentra em comportamentos externamente visíveis.

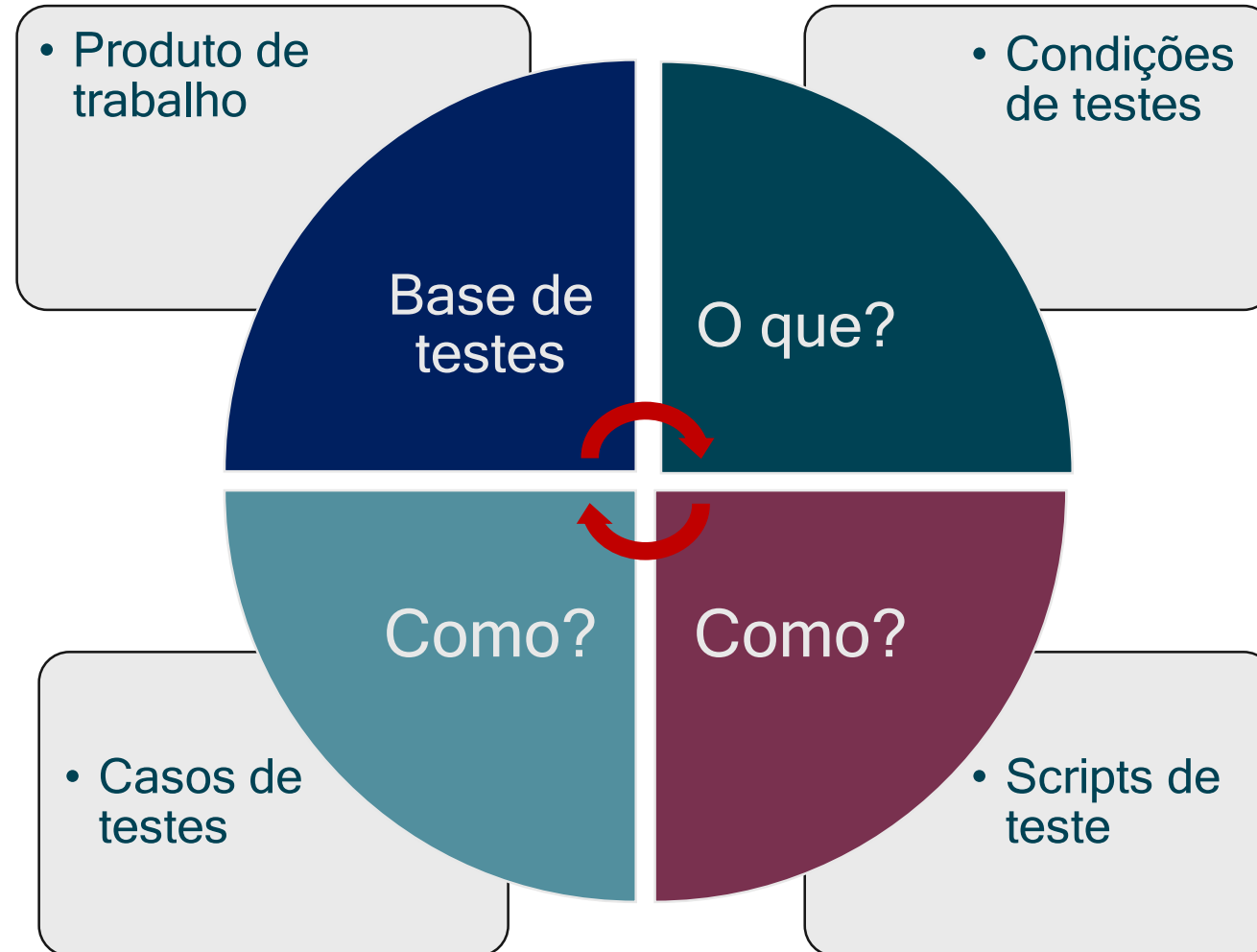
6.2 Revisão por Pares (Testes Estáticos)

PROJECT TEST PLAN		AUTHOR: A TESTER	
INTRODUCTION	IN SCOPE	OUT OF SCOPE	
ASSUMPTIONS	RESOURCES	ENVIRONMENTS - TOOLS	
	TIMESCALES		

Os defeitos mais comuns que são mais fáceis e baratos de encontrar e corrigir por meios de testes estáticos incluem:

Tipo de Defeito	Descrição
Defeitos em requisitos	Inconsistências, ambiguidades, contradições, omissões, imprecisões e redundância.
Defeitos de design	Algoritmos ineficientes ou estruturas de banco de dados , alto acoplamento, baixa coesão.
Defeitos de codificação	Variáveis com valores indefinidos, variáveis que são declaradas mas nunca utilizadas, código inacessível, código duplicado.
Desvio dos padrões	Falta de aderência dos padrões de codificação
Vulnerabilidades de segurança	Suscetibilidade a estouro de buffer
Lacunas ou imprecisões na rastreabilidade ou cobertura de testes	Ausencia de testes para um critério de aceite, falta de qualidade no mapeamento dos cenários.

6.3 Modelagem de Testes - Identificando as condições e desenhando os casos de testes

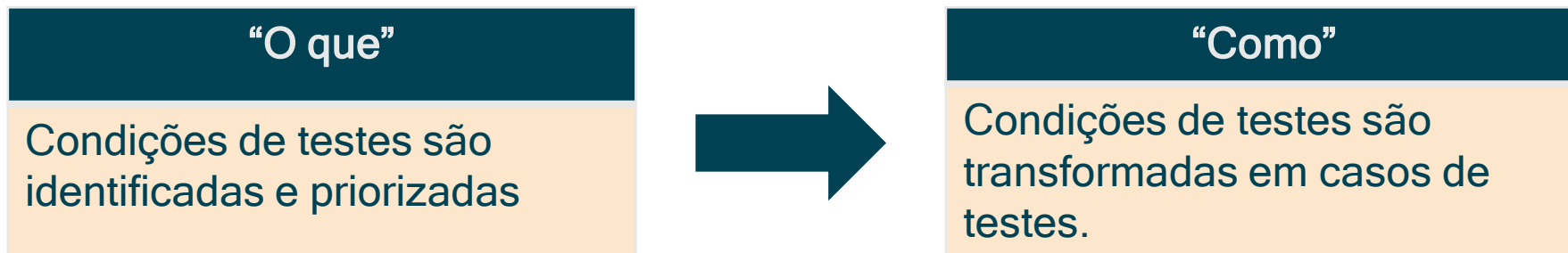


6.3 Modelagem de Testes - Condições de Teste

Condição de teste pode ser definido como um aspecto da base de teste que é deve ser levado em consideração para possa ser atingidos os objetivos específicos do teste.

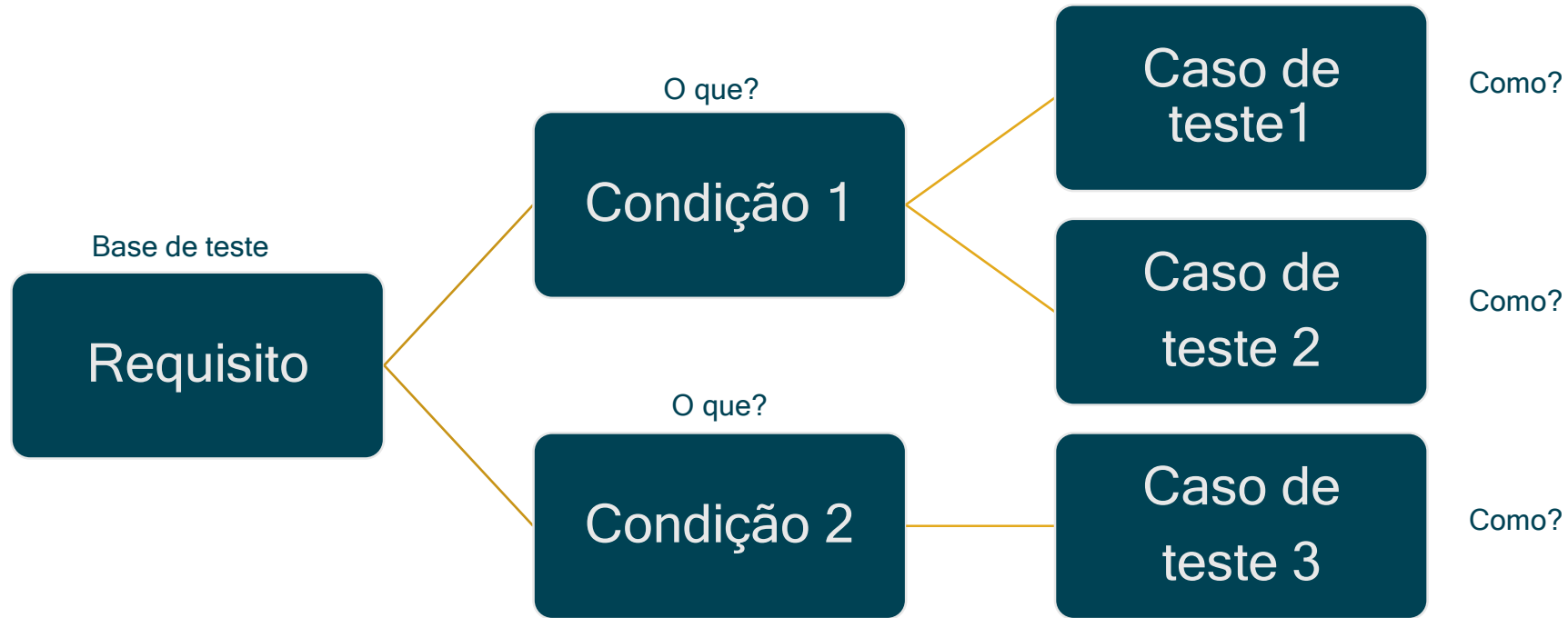
A preparação do teste pode ser colocado da seguinte forma:

1. Definir “o que” será testado;
2. Definir “como” será testado.



6.3 Modelagem de Testes - Condições de Teste

Uma **Condição de teste** é um requisito que deve ser satisfeito por um ou mais casos de teste.



Nem todas as condições de testes são tão importantes. Por isso é aconselhável que um risco ou uma prioridade seja dada a cada condição de testes (quando realizamos desta forma, estamos utilizando a abordagem baseada em riscos).

6.3 Modelagem de Testes - O que é caso de teste?

Caso de teste é um conjunto de pré-condições, entradas, ações, resultados esperados e pós condições, desenvolvidos com base nas condições de testes.

São eles que definem as entradas a serem informadas pelo testador (manualmente ou com apoio ferramental) e os resultados esperados a partir desta ação (ou seja, é um procedimento de teste que define os passos necessários para que um usuário do sistema possa realizar a execução do teste de maneira que ele consiga avaliar se o resultado atual é sucesso ou falha).

A Especificação/Modelagem das suítes e dos casos de testes permitem que a gente possa mapear todas as validações que existem no escopo de entrega e assim garantir os cenários mínimos a serem validados.

6.3 Modelagem de Testes - Como Criar?

- **Título do Caso de Teste:** O título deve ser claro e objetivo, para que se permita entender a primeira vista a proposta de validação do teste;
- **Objetivo do Teste:**
 - Neste item, deve ser informado o objetivo detalhado para a realização da validação dos testes;
 - Conter apenas um objetivo central por caso de teste. Se na escrita você verificar que possui mais de um objetivo, quebre em mais de um caso de teste;
- **Precondições:** O que é necessário, essencial para que eu consiga testar esse caso de teste? Neste item, deverá ser descrito todas as informações para que os testes possam ser realizados;

6.3 Modelagem de Testes - Como criar?

- **Passos de Reprodução (Ou Steps) com Resultados Esperados:**
 - Neste item, deverá ser descrito o passo a passo do usuário e do sistema, para que se consiga validar o fluxo de teste (ou funcionalidade) e conseguir fazer a avaliação de **sucesso** ou **falha**.
 - As maiores boas práticas na escrita de um caso de teste, é em referência à este item, onde destacamos: descrição completa do acesso ao ambiente dos testes (URL, Menu, Tela/Modal, etc) e descrição completa do passo a passo.

6.3 Modelagem de Testes - Boas Práticas

1) Os casos de testes devem sempre iniciar com um **verbo infinitivo** – preencher, validar, clicar, acionar – ou **imperativo** – preencha, valide, verifique. Exemplos:

- Validar a máscara do campo “CPF”.
- Verificar se o registro foi salvo na base de dados.
- Clique no botão “Voltar”.
- Acessar o menu Administração Geral > Configurações.

6.3 Modelagem de Testes - Boas Práticas

2) Os **resultados esperados** são as respostas às ações realizadas pelo usuário. Cada ação deverá ter um resultado esperado:

#	Ação	Resultado Esperado
1	Validar a máscara do campo "CPF".	Exibida máscara 999.999.999-99.
2	Preencher o campo "CPF" com dados válidos e clicar no botão "Salvar".	Registro salvo na tabela <nome_da_tabela>.
3	Clicar no botão "Voltar".	Sistema deve retornar à página inicial.

3) **Pode X Deve**: nunca utilize frases como "o sistema *pode*", prefira a palavra **deve**, assim não se deixa dúvidas do resultado esperado para a ação.



6.4 Execução e Gerenciamento de Incidentes

A fase de execução dos testes é a fase onde realmente a “mágica acontece”.

É - finalmente - quando a qualidade poderá ser avaliada e também é quando conseguimos ter a percepção do que será entregue em Homologação/Produção - ou seja, se ela é compatível com a expectativa do cliente.

A fase de execução é o reflexo de todo o projeto, desde a fase de requisitos, passando pelo desenvolvimento e finalizando nos testes.



6.4 Execução e Gerenciamento de Incidentes

Para essa fase, o Analista de Qualidade deverá realizar a execução dos testes garantindo:

- Execução dos Testes Iniciais, baseados nos casos de testes;
- Execução dos Testes Exploratórios, baseados no conhecimento funcional ou conhecimento técnico;
- Abertura de Bugs com Classificação de Tipologia (Requirement Error, Implementation Error, etc), Criticidade/Severidade (Blocker, Critical, High, Minor e Trivial) e Prioridade (Baixa, Média, Alta e Crítica)
- Gestão de Baselines de Teste;
- Reteste de Bugs;
- Execução de Regressão Mínima após finalização dos testes.

6.4 Execução e Gerenciamento de Incidentes



Ops, encontrei um erro! O que devo fazer?





6.4 Execução e Gerenciamento de Incidentes

Abrir um bug, claro! E para isso, deveremos utilizar o seguinte template:

- Título: [BUG] - ID - Número do Requisito, História do Usuário ou Caso de Uso - Problema.
- Descrição do Objetivo;
- Pré-condições/Dados de Teste;
- Passo a passo do teste;
- Resultado Esperado;
- Resultado atual;
- Evidências (imagem, documento ou vídeo).

