

REQUISITOS DE SOFTWARE



REQUISITO

- Um requisito de sistema descreve o que é requerido para que o sistema cumpra o seu objetivo.
- Engenharia de Requisitos é o processo de descobrir, analisar, documentar e verificar as funções e restrições do sistema (*Sommerville, 2003*).

REQUISITO

- Uma **condição** ou uma **funcionalidade** necessária a um **usuário** para resolver um problema.
- Uma **condição** ou **funcionalidade** que deve ser **obtida** por um componente de sistema para satisfazer um contrato, padrão, especificação, ou outro documento formalmente definido.

(IEEE)

OBJETIVOS DA ENGENHARIA DE REQUISITOS

- Estabelecer e manter concordância com os clientes e desenvolvedores;
- Registrar e acompanhar requisitos ao longo de todo o desenvolvimento;
- Definir as fronteiras do sistema;
- Fornecer a base para o cronograma/custo de desenvolvimento do sistema.

NÍVEIS DE REQUISITOS

Requisitos do usuário → Declarações em linguagem natural e também em diagramas, sobre as funções que o sistema deve fornecer e as restrições sob os quais deve operar.

Requisitos de sistema → Um documento estruturado com descrições detalhadas dos serviços de sistemas. Escrito como um contrato entre o cliente e contratante .

REQUISITOS FUNCIONAIS

- Declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações.

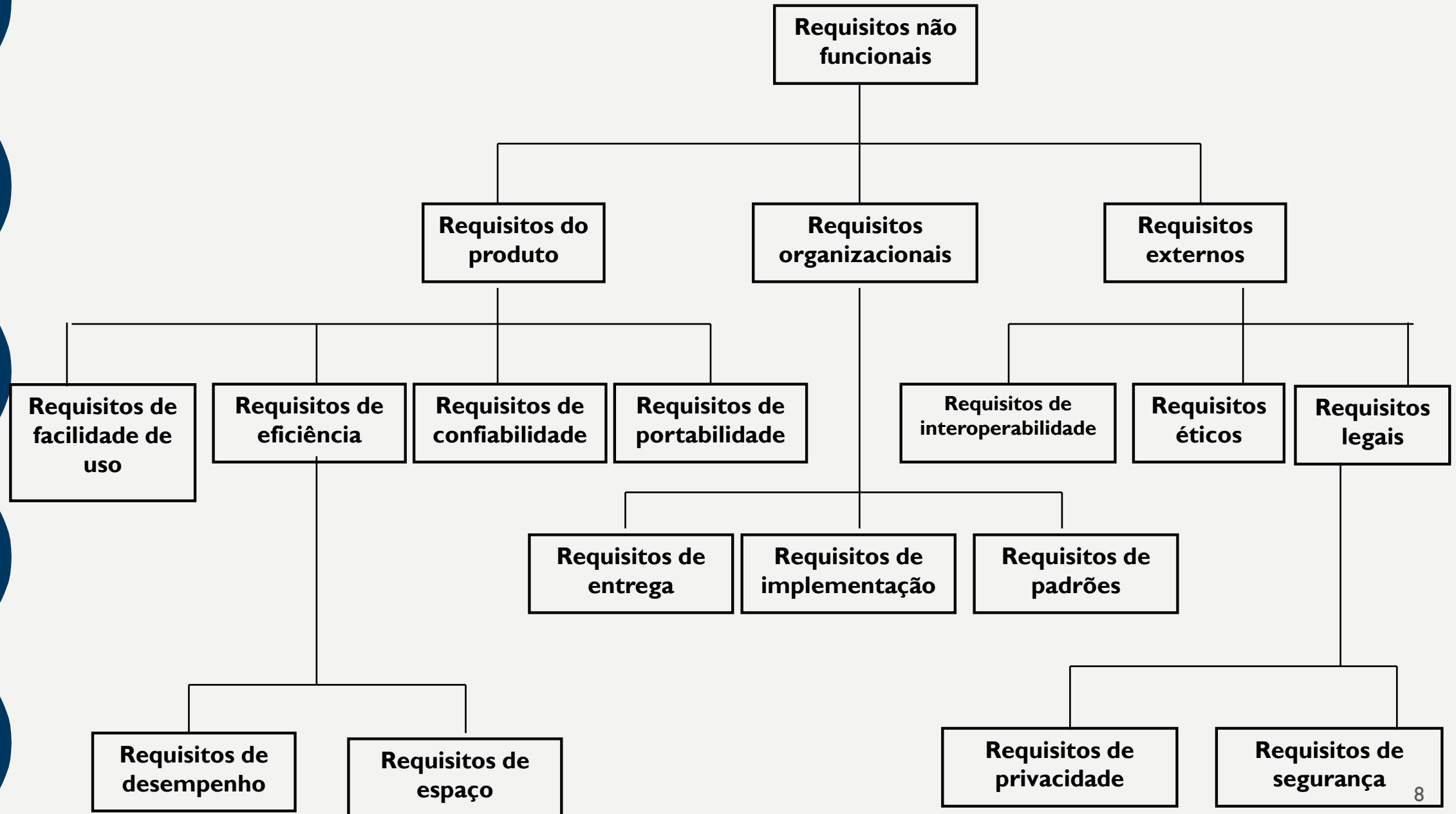
Qualidade e restrições sobre os serviços ou as funções oferecidas pelo sistema.

Podem estar relacionados propriedades de confiabilidade, tempo de resposta, Segurança.

A falha de não cumprir com um requisito não funcional de sistema pode tornar todo o sistema inútil.

REQUISITOS NÃO FUNCIONAIS

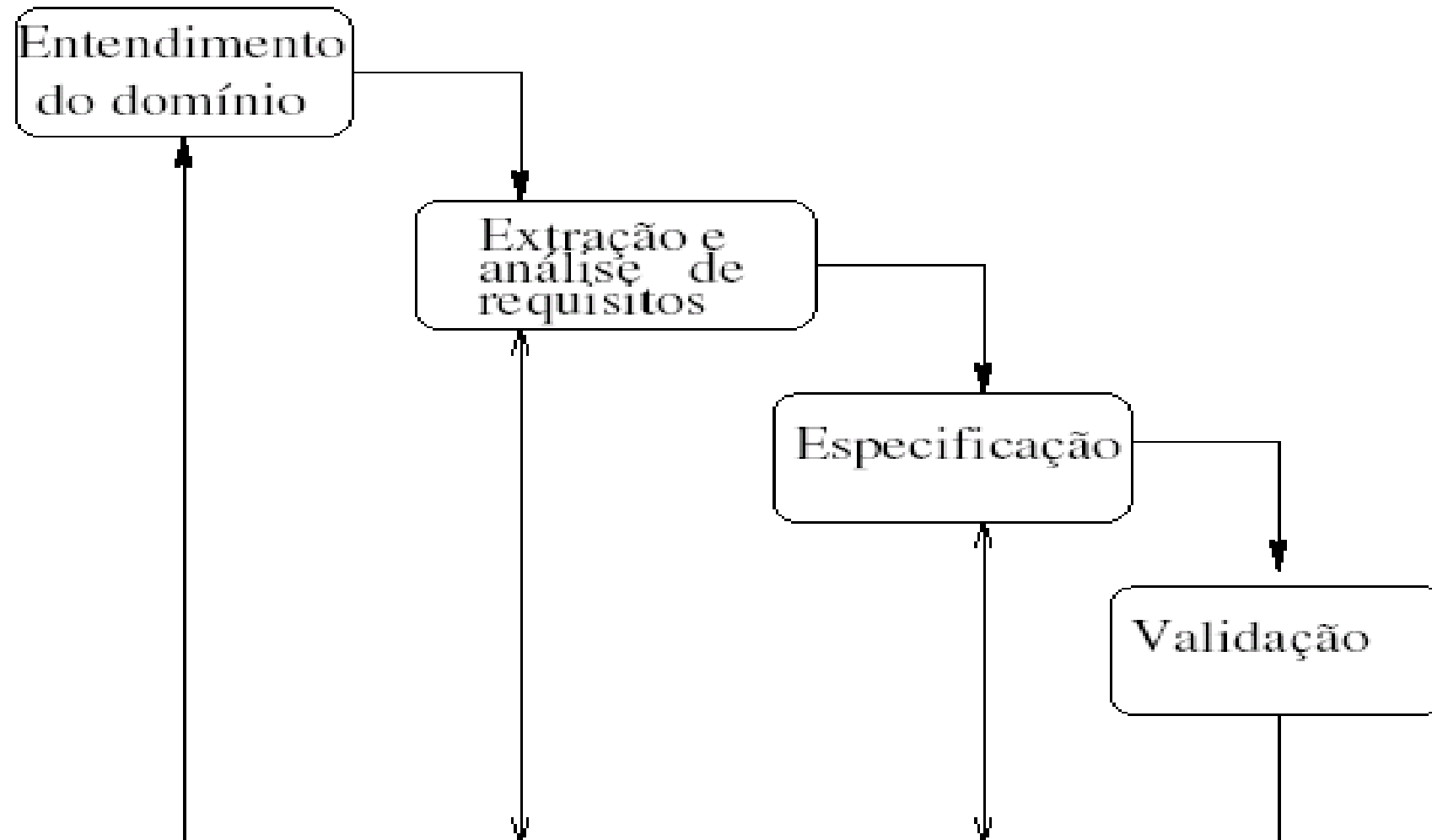
TIPOS DE REQUISITOS NÃO FUNCIONAIS



STAKEHOLDERS (PARTES INTERESSADAS)

- *Internos: Pessoas cujo interesse na organização que conterá o sistema vem de um **relacionamento direto**.*
- *Externos: Pessoas que não trabalham diretamente na organização que conterá o sistema, mas **são afetados de alguma forma pelas ações deste negócio**.*

PASSOS PARA O PROCESSO DE EXTRAÇÃO DE REQUISITOS DE SOFTWARE



TESTE DE SOFTWARE

DEFININDO TESTE DE SOFTWARE

- O teste do software é uma das fases do processo de **engenharia de software** que visa atingir um nível de **qualidade** de produto superior.
- A maioria das pessoas pensa que o teste de software serve para demonstrar o correto funcionamento de um programa...

CONTROLE DE QUALIDADE!

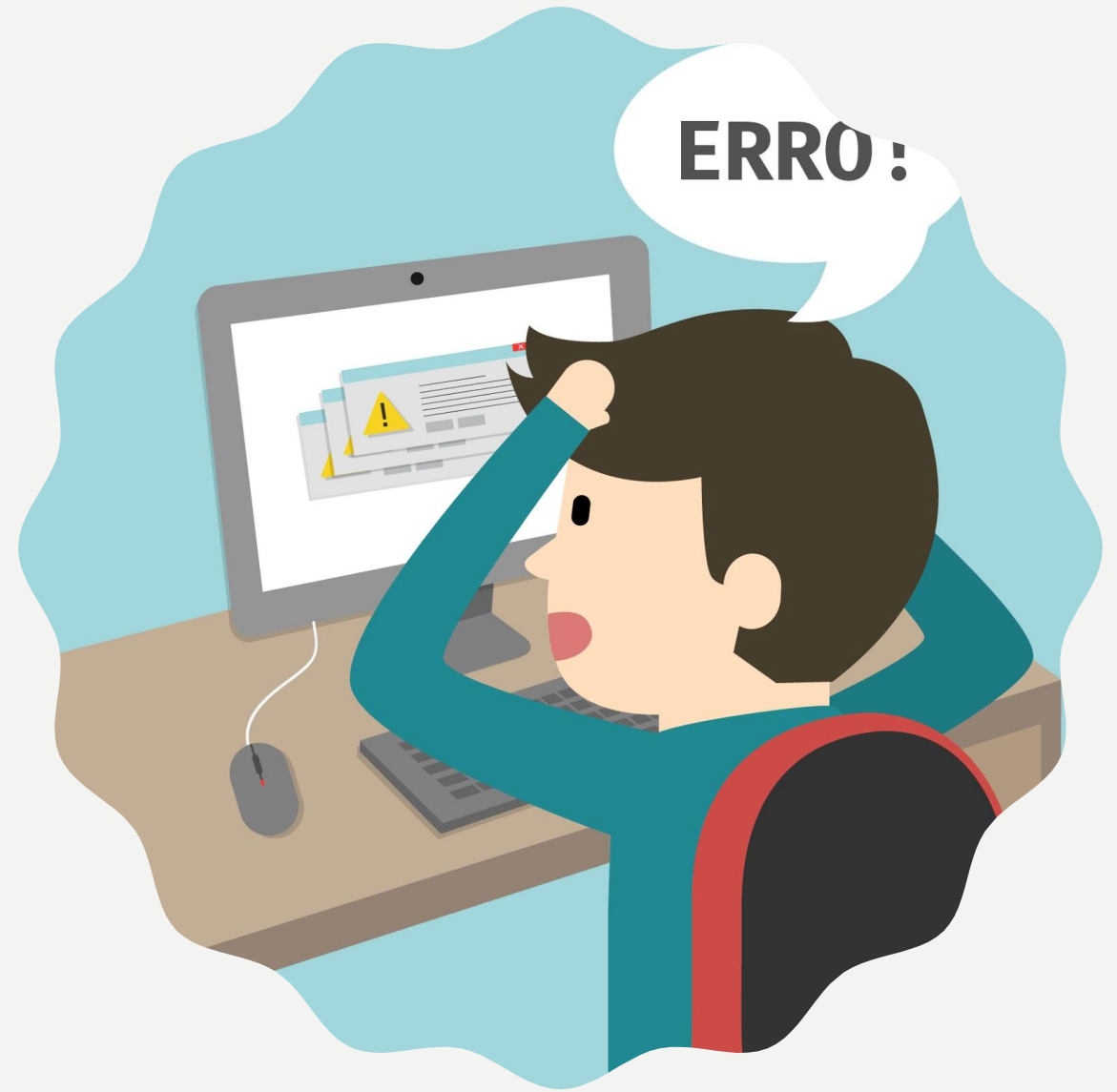
O controle de qualidade é feito através de uma série de inspeções, revisões e testes, usados através do ciclo de desenvolvimento para garantir que cada trabalho produzido está de acordo com sua especificação/requerimento.

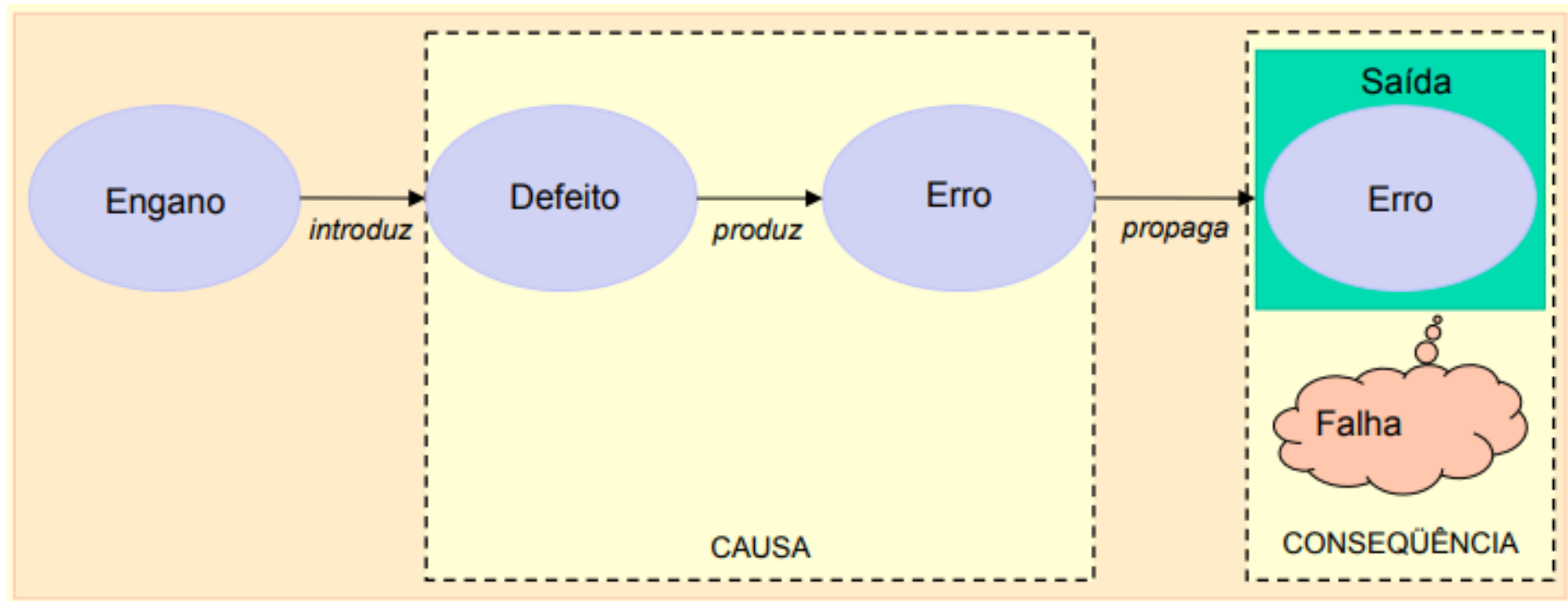
Engano: Digitar algo errado no código, como troca ponto e vírgula por dois pontos no final do comando.

Defeito: deficiência mecânica ou algorítmica que, se ativada, pode levar a uma falha - Instrução ou comando incorreto.

Erro: item de informação ou estado de execução inconsistente.

Falha: evento notável em que o sistema viola suas especificações .





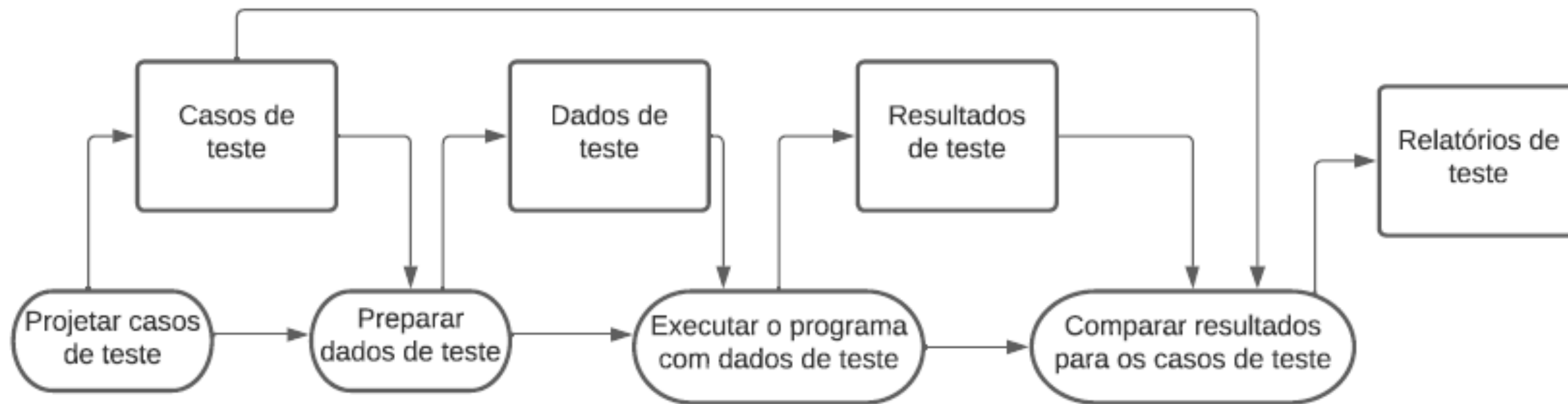
DEFEITOS NO PROCESSO DE DESENV.

- São gerados na comunicação e na transformação de informações.
- Continuam presentes nos diversos produtos de software produzidos e liberados (10 defeitos a cada 1000 linhas de código).
- A maioria encontra-se em partes do código raramente executadas .
- **Principal causa:** tradução incorreta de informações .
- Quanto antes a presença do defeito for revelada, menor o custo de correção do defeito e maior a probabilidade de corrigi-lo corretamente.

PORQUE OCORREM FALHAS?

- A especificação pode estar errada ou incompleta;
- Talvez a base de dados esteja organizada de forma que não seja permitido distinguir os tipos de usuário;
- Pode ser que haja um erro no algoritmo de controle dos usuários;





ETAPAS DO TESTE

CATEGORIAS DE TESTE DE SOFTWARE

- Teste de Unidade
- Teste de Componente
- Teste de Integração
- Teste de Sistema
- Teste de Aceitação

NÍVEIS DE TESTE DE SOFTWARE

Teste Unitário	Teste de integração	Teste de Sistema	Teste de Aceitação
			
Testar componentes ou unidades de forma isolada.	Testar a integração das unidades que foram testadas isoladamente.	Provar que o sistema não funciona. Testar requisitos funcionais e não-funcionais.	Teste de aceitação ou homologação. Obter a homologação do cliente.
Especificações Técnicas e Código Fonte		Requisitos de negócio, funcionais e não-funcionais	
Programador	Programador/Arquiteto	Analista de Teste	Cliente

TESTE DE UNIDADE/UNITÁRIO

- É um tipo de atividade que visa testar pequenas partes ou unidades do sistema;
- O universo desse tipo de teste são os métodos dos objetos, atributos e eventos que alteram a mudança de estado;
- Encontrar falhas de funcionamento dentro de uma pequena parte do sistema, independentemente do todo.

TESTE DE COMPONENTE (INTEGRAÇÃO)

- Propósito de testar um componente como um todo e não apenas as suas funções ou métodos;
- Mesmo assim, o teste continua a ser executado sem considerar a interação com outras partes do sistema , ou seja, leva-se apenas em consideração o componente a ser testado e nenhuma outra entidade do sistema.
- Ex: Teste com a chamada de outros métodos.
 - Problemas com passagem de parâmetro.

TESTE DE COMPONENTE (INTEGRAÇÃO)

- Os tipos de falhas encontradas são de envio e recebimento de dados, Ex: um objeto A pode estar esperando o retorno de um valor x ao executar um método do objeto B, porém este objeto B pode retornar um valor y, ou seja, diferente do esperado.

TESTE DE SISTEMA

- Tem por objetivo varrer o sistema em busca de falhas através da utilização do mesmo, como se fosse um usuário final.
- Os testes são executados nos mesmos ambientes , com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia de manipulação do sistema.

TESTE DE ACEITAÇÃO

- Realizados por um restrito grupo de usuários finais do sistema.
- Simulam operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado.
- Usado para determinar se um sistema satisfaz ou não seus critérios de aceitação.

TESTE DE REGRESSÃO

- Essa abordagem tem a função de garantir que os módulos ou componentes da aplicação que não foram modificadas ainda funcionam corretamente após o programador modificar alguma outra parte da aplicação.

TESTE DE REGRESSÃO

- Essa abordagem tem a função de garantir que os módulos ou componentes da aplicação que não foram modificadas ainda funcionam corretamente após o programador modificar alguma outra parte da aplicação.

BUG

- BUG é uma palavra genérica para representar qualquer classe de defeito.
- O termo foi introduzido quando os primeiros computadores, que eram feitos de válvulas, apresentavam algum tipo de problema inexplicável.

ALGUMAS CAUSAS DE BUG

- Falta de comunicação entre os membros da equipe.
- A complexidade do software.
- Erros de programação.
- Mudança de requerimentos no meio do projeto.
- Requerimentos ambíguos.
- Pressão da gerência sobre o prazo do projeto.

PLANEJAMENTO DE TESTES

- Planejar a capacidade;
- Padronizar entradas e saídas;
- Definir atividades e métodos;
- Estabelecer e coletar métricas;
- Verificar o processo;
- Ambiente;
- Cronograma, riscos e objetivos.

PLANOS DE TESTES

- Introdução com identificação do projeto (definições, abreviações, referências), definição de escopo e objetivos;
- Conjunto de requisitos a serem testados;
- Tipos de testes a serem realizados e ferramentas utilizadas;
- Recursos utilizados nos testes;
- Cronograma de atividades (e definição de marcos de projeto).

PLANEJAMENTO DE TESTES

- Questões como que tipo de testes serão realizados de acordo com a natureza do sistema a ser desenvolvido;
- Quais ações serão realizadas para esses testes;
- Quais programas de testes serão utilizados: como reportar erros a serem revelados, controlar o software a ser testado e como aceitar o teste;
- Quais treinamentos eles precisam para realizar essas ações;
- Quais responsabilidades eles têm dentro do processo de teste, onde esses testes serão realizados e como seu tempo será planejado são decididos.