**Introdução ao teste de Software.**

**“TESTES MOSTRAM A PRESENÇA, NÃO A AUSÊNCIA DE ERROS!”**

**Atividade de teste é necessária porque quem desenvolve software somos nós, humanos. E humanos podem cometer erros.** E erros podem ocorrer por diversos fatores: complexidade do software desenvolvido, mudanças necessárias em software já existente e que pode levar a novas falhas, falta de uso de métodos que apoiem a validação e verificação da qualidade do software, situações imprevisíveis, questões pessoais e do próprio ambiente de trabalho e que interferem no desempenho do testador, dentre outras questões.

Considerando a importância da atividade de software sob um ponto de vista mais técnico e de negócio, ***ela é requerida para revelar a presença de defeitos, também pode ser usada para aumentar a confiança e satisfação do cliente com o software, e é útil para assegurar a qualidade do produto.***

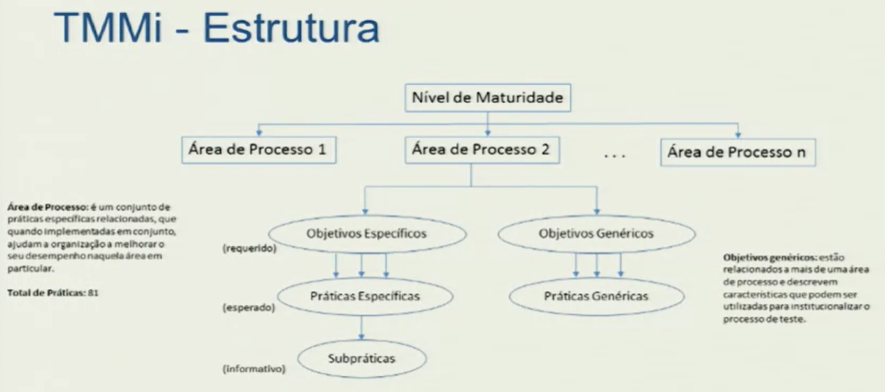
***Falhas em softwares são extremamente perigosas. Elas podem causar perdas monetárias e humanas irreversíveis, e a história está cheia de tais exemplos.***

**Então, é muito importante que você, enquanto gerente de uma empresa de software, estudante da área de tecnologia, profissional, educador e/ou pesquisador da área de software, esteja ciente da importância da atividade de Teste de Software. E, mais do que isso, reconheça o valor do profissional que atua nessa área, seja no mercado de trabalho ou na academia.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

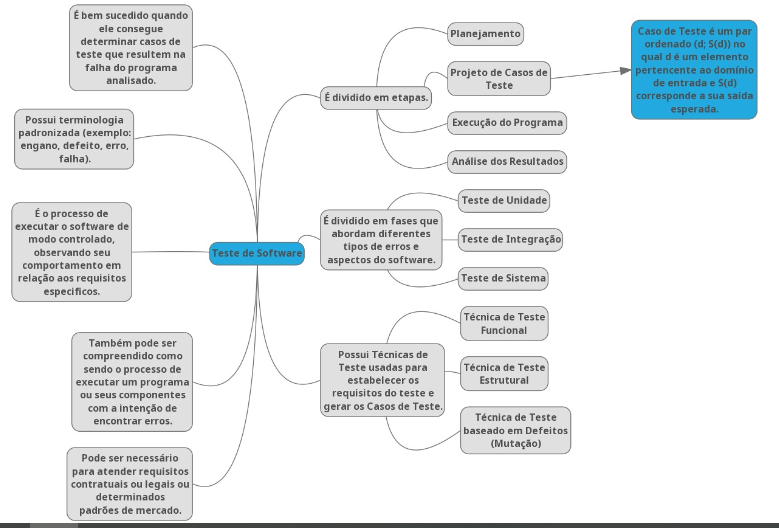
**TMMI- Test Maturity Model Integration (Integração do Modelo de Maturidade de Teste)**

Estabelece requisitos necessários para ter um processo de testes de software com qualidade.



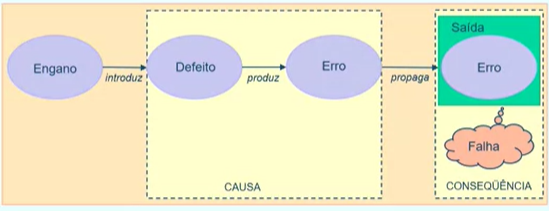
**“Resultados da pesquisa mostraram que das 81 práticas do TMMI, 33 são utilizadas e consideradas essenciais.”** (Trabalho de mestrado orientado por Profª Sandra Fabbri, Profº Fabiano Ferrari)

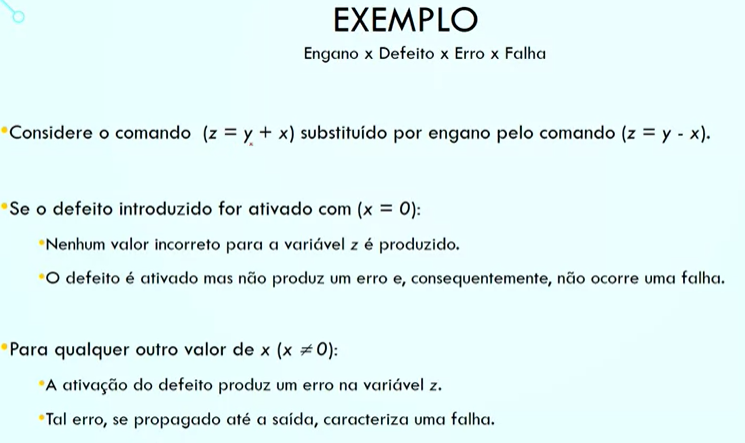
### O Mapa Conceitual a seguir apresenta uma ideia geral dos conceitos e definições envolvidas.



**Engano x Defeito x Erro x Falha (Mistake x Fault x Error x Failure)**

* Um **engano** introduz um **defeito** no software (ação humana que produz um resultado incorreto);
* O **defeito** (um passo, processo, ou definição de dados incorreta em um produto de software), quando ativado, pode **produzir** um **erro** (diferença do valor computado, observado ou medido e o valor teoricamente correto de acordo com a especificação);
* O **erro**, se **propagado** até a saída do software, constitui uma **falha** (inabilidade do sistema ou componente realizar a função requerida, considerando as questões de desempenho exigidas).





# Terminologias de Teste de Software (Conceitos Básicos)

Toda área profissional tem suas terminologias bem definidas no teste de software, isso não seria diferente.

Sendo assim, trago aqui uma lista simples com alguns termos e seus significados na área de teste.

* **Anomalia:** qualquer sintoma, comportamento ou resultado diferente do esperado ou previsto.
* **Ciclo de Teste:** Execução da massa de teste sobre o produto de software a cada versão que é liberada. Um passo pode conter um ou mais ciclos de teste.
* **Critério de Cobertura:** de maneira simplificada é o que define regras práticas para quando parar os testes. Porcentagem do software que será testada e a equipe entende como aceitável. Ex: 75% de cobertura.
* **Defeito/Bug:** passo, processo ou definição de dados incorretos. Ou ainda qualquer problema existente no software ou sistema sob teste que pode causar sua falha em atender as expectativas do usuário. Em outras palavras, um defeito é uma fonte potencial de insatisfação com o produto.
* **Engano:** ação humana que produz um defeito.  
  Erro: estado inconsistente, observado em momento de execução, de um produto ou sistema de software. Esse erro pode ou não causar uma falha.
* **Falha:** estado inesperado do produto ou sistema de software ocasionado por um erro.
* **Ferramenta de teste:** qualquer ferramenta de hardware e/ou software utilizada durante a execução de um caso de teste. Podendo ser utilizada para configurar o ambiente, criar condições de teste, ou medir os resultados dos testes. Uma ferramenta de teste geralmente é separada do caso de teste em si. Ex: Postman, Selenium, etc.
* **Feedback de usuário:** informações do ponto de vista de usuário sobre determinado produto, atividade, ou assunto. Pode ainda ser entendido como a avaliação de determinado assunto.
* **Passo de teste: (test step):**cada instrução presente em um caso de teste. Ex: abrir a página do google; digitar “vinicius pessoni”; clicar no botão pesquisar.
* **Testadores (Teters, QA- Quality Assurance, QE – Quality Engineer):** responsáveis por configurar o ambiente de teste, iniciar os testes, interpretar resultados, reproduzir anomalias manualmente, isolar defeitos por experimentação, e restaurar o ambiente ao seu estado anterior à configuração para execução do teste. Em adicional possuem a responsabilidade de dar manutenção no sistema de teste, especialmente nos casos e conjuntos de teste.
* **Teste de Unidade ou Componente:** testa a menor parte de um software, definido quando utilizado. Exemplo: função, método, classe.

**ETAPAS DA ATIVIDADE DE TESTE**

* **Planejamento: Recursos humanos, tecnológicos, estratégias, técnicas de teste e critérios.**

**Plano de teste:** Planejamento para execução do teste, funcionalidades a serem testadas, tarefas de teste; Riscos associados, cronograma, requisitos do ambiente, etc. Claro, aqui está um modelo básico de plano de teste:

**Buscar por:** Modelo de Plano de Teste (Testing Plan Model), Template de Plano de Teste (Testing Plan Template).

**--------------------------------------------------------------------------------------------------------------------**

**EXEMPLO**

**# Plano de Teste**

**## Introdução**

O objetivo deste plano de teste é definir as atividades, recursos e procedimentos necessários para garantir que o sistema / software esteja funcionando conforme o esperado e atenda aos requisitos do usuário.

**## Objetivos**

Os objetivos deste plano de teste são:

- Identificar e documentar os requisitos de teste.

- Desenvolver testes para validar o sistema / software.

- Realizar testes para garantir que o sistema / software atenda aos requisitos do usuário.

- Documentar os resultados dos testes e quaisquer problemas encontrados.

- Garantir que o sistema / software esteja pronto para ser implantado / entregue.

**## Escopo**

O escopo deste plano de teste inclui:

- Testes de integração do sistema.

- Testes de unidade.

- Testes de aceitação do usuário.

- Testes de desempenho.

- Testes de segurança.

**## Responsabilidades**

As responsabilidades das partes envolvidas no plano de teste são:

- A equipe de desenvolvimento fornecerá o software para teste.

- A equipe de teste desenvolverá e executará testes.

- O gerente de projeto supervisionará todo o processo de teste.

**## Recursos**

Os recursos necessários para este plano de teste são:

- Equipe de teste.

- Ambiente de teste.

- Ferramentas de teste.

- Conjunto de dados de teste.

**## Cronograma**

O cronograma para este plano de teste é:

- Desenvolvimento de testes: 2 semanas.

- Execução de testes: 4 semanas.

- Documentação de testes: 1 semana.

**## Requisitos de teste**

Os requisitos de teste incluem:

- Requisitos funcionais do software.

- Requisitos não funcionais do software, como desempenho e segurança.

- Requisitos de aceitação do usuário.

**## Testes**

Os testes incluem:

- Testes de integração do sistema.

- Testes de unidade.

- Testes de aceitação do usuário.

- Testes de desempenho.

- Testes de segurança.

**## Critérios de aceitação**

Os critérios de aceitação são:

- O software deve passar em todos os testes definidos.

- O software deve atender a todos os requisitos funcionais e não funcionais.

- O software deve ser aceito pelo usuário.

**## Resultados dos testes**

Os resultados dos testes serão documentados em um relatório de teste que incluirá:

- Descrição do teste realizado.

- Resultados do teste.

- Problemas encontrados durante o teste.

- Ações corretivas tomadas.

**## Aprovação**

Este plano de teste deve ser revisado e aprovado pelo gerente de projeto antes de iniciar o processo de teste.

**--------------------------------------------------------------------------------------------------------------------**

* **Projetos de Casos de teste: São elaborados os casos de teste com o quais o programa deve ser testado.**

**Saída:** Especificação de Projeto de Caso de Teste (refina o planejamento); Especificação de Caso de Teste (Dados de entrado, resultados esperados, ações para execução do teste); Especificação de Procedimento de Teste (passos para executar o conjunto de casos de teste).

**---------------------------------------------------------------------------------------------------------------**

**EXEMPLO**

**# Caso de Teste**

**## Identificação**

- ID do caso de teste:

- Nome do caso de teste:

**## Descrição**

- Descrição do objetivo do caso de teste.

**## Pré-condições**

- Condições necessárias para que o caso de teste seja executado com sucesso.

**## Passos**

- Lista de passos a serem seguidos para executar o teste.

**## Dados de entrada**

- Descrição dos dados de entrada necessários para executar o teste.

**## Resultados esperados**

- Descrição dos resultados esperados após a execução do teste.

**## Critérios de sucesso**

- Descrição dos critérios que devem ser atendidos para considerar o teste bem-sucedido.

**## Pós-condições**

- Estado do sistema após a execução do teste.

**## Ambiente de teste**

- Descrição do ambiente de teste necessário para executar o teste.

**## Observações**

- Quaisquer observações ou notas adicionais relevantes para o teste.

**## Resultados do teste**

- Descrição dos resultados reais após a execução do teste.

**## Conclusão**

- Conclusão do teste, incluindo se o teste foi bem-sucedido ou não e quaisquer problemas encontrados durante o teste.

**---------------------------------------------------------------------------------------------------------------**

* **Execução: O programa é executado com os casos de teste elaborados.**

**Saída:** Diário de teste (detalhes cronológicos da execução dos testes); Relatório de Incidente de teste (qualquer evento ocorrido na execução e que precise ser analisado depois); Relatório de encaminhamento de item de teste.

**--------------------------------------------------------------------------------------------------------------------**

**EXEMPLO**

**# Relatório de Incidente de Teste**

**## Identificação do Incidente**

- ID do incidente:

- Nome do incidente:

**## Descrição**

- Descrição detalhada do incidente encontrado durante o teste.

**## Ambiente de Teste**

- Descrição do ambiente de teste em que o incidente foi encontrado.

**## Passos para Reproduzir**

- Lista de passos necessários para reproduzir o incidente.

**## Dados de Entrada**

- Descrição dos dados de entrada usados ​​para reproduzir o incidente.

**## Comportamento Esperado**

- Descrição do comportamento esperado do sistema / software.

**## Comportamento Atual**

- Descrição do comportamento real do sistema / software.

**## Severidade**

- Nível de gravidade do incidente (por exemplo, alta, média ou baixa).

**## Prioridade**

- Nível de prioridade do incidente (por exemplo, alta, média ou baixa).

**## Status**

- Status atual do incidente (por exemplo, aberto, em andamento, resolvido).

**## Ações Corretivas**

- Lista de ações corretivas recomendadas para resolver o incidente.

**## Responsáveis**

- Lista de pessoas responsáveis ​​por resolver o incidente.

**## Data de Abertura**

- Data em que o incidente foi aberto.

**## Data de Fechamento**

- Data em que o incidente foi fechado.

**## Anexos**

- Anexos relevantes, como capturas de tela ou logs de erro.

**## Conclusão**

- Conclusão do relatório de incidente, incluindo se o incidente foi resolvido com sucesso ou se há ações adicionais necessárias.

**--------------------------------------------------------------------------------------------------------------------**

* **Análise dos resultados: Avalia-se o comportamento do programa testado, considerando os casos de teste.**

**Saída:** Relatório de Resumo de Teste.

**-**--------------------------------------------------------------------------------------------------------------

**Exemplo**

**# Relatório de Resumo de Teste**

**## Identificação**

- Nome do projeto / sistema testado:

- Período de tempo dos testes:

- Equipe responsável pelos testes:

**## Objetivos dos Testes**

- Descrição dos objetivos dos testes realizados.

**## Resumo dos Testes Realizados**

- Número total de casos de teste executados:

- Número total de casos de teste bem-sucedidos:

- Número total de casos de teste com falhas:

- Número total de casos de teste bloqueados:

- Percentual de sucesso geral dos testes:

**## Principais Problemas Encontrados**

- Lista dos principais problemas encontrados durante os testes, incluindo uma descrição breve do problema, sua gravidade e status atual.

**## Recomendações**

- Lista de recomendações para melhorar a qualidade do software / sistema testado.

**## Conclusão**

- Conclusão geral dos testes realizados, incluindo uma avaliação da qualidade do software / sistema testado e sugestões para melhorias futuras.

**## Anexos**

- Anexos relevantes, como capturas de tela ou logs de erro.

**---------------------------------------------------------------------------------------------------------------**

**FASES DA ATIVIDADE DE TESTE**

**Teste de unidade**

**Foco:** menores unidades de um software (funções, procedimentos, métodos, classes).

**Erros que podem ser relatados:** erros simples de programação, estruturas de dados incorretas, algoritmos mal implementados ou incorretos.

**Observação:** pode ser feito durante a implementação e pelo próprio desenvolvedor.

**Teste de Integração**

**Foco:** Construção estrutura do software.

**Erros que podem ser relatados:** Interfaces entre as unidades.

**Observação:** feito após testar as unidades; executado pela equipe de desenvolvimento, que conhece a estrutura interna do software.

**Teste de Sistema**

**Foco:** o Sistema.

**Erros que podem ser relatados:** erros de funções e características de desempenho que não estejam de acordo com a especificação.

**Observação:** feito após o sistema estar completo; pode ser feito por uma equipe independente; pode-se considerar requisitos funcionais e não funcionais (segurança, performance, etc).

**Teste de Regressão**

Manutenções podem inserir novos defeitos.

Por isso, este teste é feito **após as manutenções do software**, para garantir que as **modificações** estão corretas e os requisitos anteriormente testados **continuem válidos.**

**As fases de atividade de teste de software variam de acordo com a metodologia de desenvolvimento utilizada e as necessidades do projeto. No entanto, em geral, as fases de atividade de teste de software incluem:**

**1. Planejamento de teste:** nesta fase, é definido o escopo dos testes, o cronograma e o orçamento. Também são definidos os critérios de aceitação e os recursos necessários para a realização dos testes.

- Definição dos requisitos de teste

- Identificação dos riscos do projeto

- Definição de metas e objetivos de teste

- Definição do escopo do teste

- Definição do cronograma e orçamento do teste

- Definição dos critérios de aceitação do software

**2. Projeto de teste:** nesta fase, é criado um plano detalhado para a realização dos testes, incluindo a definição dos casos de teste, a seleção das ferramentas de teste e a preparação do ambiente de teste.

- Identificação dos casos de teste e dos dados de entrada e saída

- Definição das técnicas de teste a serem utilizadas

- Definição dos critérios de sucesso de cada caso de teste

- Criação de planos de teste para diferentes tipos de teste (por exemplo, teste de integração, teste de unidade, teste de aceitação do usuário)

**3. Preparação para teste:** nesta fase, são realizadas as atividades necessárias para configurar o ambiente de teste, como a instalação do software, a configuração dos sistemas de teste e a preparação dos dados de teste.

- Configuração do ambiente de teste

- Criação de dados de teste

- Instalação do software de teste e ferramentas de teste

- Preparação de testes automatizados

**4. Execução de teste:** nesta fase, os testes são executados de acordo com o plano estabelecido, os resultados são registrados e os erros são reportados para a equipe de desenvolvimento.

- Execução de testes manuais e automatizados

- Verificação dos resultados dos testes

- Registro dos resultados dos testes e dos erros encontrados

- Comunicação dos resultados do teste com a equipe de desenvolvimento

**5. Análise de resultados:** nesta fase, os resultados dos testes são analisados para determinar se o software atende aos critérios de aceitação. Os erros são classificados de acordo com sua severidade e é feita uma avaliação de risco para determinar quais erros precisam ser corrigidos antes do lançamento.

- Classificação dos erros encontrados de acordo com sua severidade

- Priorização dos erros a serem corrigidos

- Avaliação do risco de cada erro

- Análise das causas dos erros

**6. Relatório de teste:** nesta fase, é produzido um relatório de teste que documenta os resultados dos testes e as decisões tomadas em relação à aceitação do software. O relatório também pode incluir recomendações para melhorias futuras.

- Documentação dos resultados dos testes

- Documentação das decisões tomadas em relação à aceitação do software

- Recomendações para melhorias futuras

**7. Encerramento de teste:** nesta fase, é feita uma revisão final dos resultados dos testes e do processo de teste como um todo. O ambiente de teste é desmontado e os recursos são liberados para outras atividades.

- Revisão final dos resultados dos testes

- Revisão final do processo de teste

- Desmontagem do ambiente de teste

- Liberação dos recursos utilizados para outras atividades

É importante lembrar que essas fases não precisam ser realizadas em ordem estrita e algumas fases podem ser realizadas simultaneamente. Além disso, a equipe de teste deve estar em constante comunicação com a equipe de desenvolvimento para garantir que o software esteja sendo testado de maneira eficaz e que os erros sejam corrigidos em tempo hábil.

**Técnicas de Teste**

