

# PROJETO, IMPLEMENTAÇÃO E TESTE DE SOFTWARE

João Choma Neto

[joao.choma@unicesumar.edu.br](mailto:joao.choma@unicesumar.edu.br)

# O que veremos nesse semestre?

- O que é teste de software
- Artefatos de teste
  - Planos de teste. Documentação de estratégias de testes e outros artefatos. Revisão de software.

# O que veremos nesse semestre?

- Princípios e técnicas de teste de software
  - Técnicas de teste funcional (caixa preta)
  - Técnicas de teste estrutural (caixa branca)
  - Técnicas de teste baseados em defeitos

O que veremos nesse semestre?

- Testes de unidade
- Testes de integração
- Testes de sistema
- Testes de aceitação
- Testes de atributos de qualidade
- Testes de regressão

O que veremos nesse semestre?

- Desenvolvimento de casos de teste
- Testes automatizados e ferramentas de apoio
- Teste em ferramentas de integração contínua

O que veremos nesse semestre?

- Desenvolvimento dirigido por testes (TDD)
- Gerenciamento do processo de teste
- Registro e acompanhamento de problemas

# Referências utilizadas

- Bibliografia básica
  - PRESSMAN, Roger S. Engenharia de software - 7. ed. / 2011 São Paulo: Makron Books, 2011. (DIGITAL)
  - ORGANIZADOR PEDRO HENRIQUE CACIQUE BRAGA. Teste de software Pearson (DIGITAL)
  - Testes de software e gerência de configuração / 2019 Porto Alegre SAGAH 2019 (DIGITAL)

# Referências utilizadas

- Bibliografia complementar
  - FÁBIO CRUZ. Scrum e Agile em Projetos - 2ª Edição / 2018 Editora Brasport 2018 (DIGITAL)
  - SANTOS, Amélia Premebida. Código de ética profissional do administrador e a prática empresarial : algumas considerações e análise, O / 2000 Maringá, 2000. (DIGITAL)
  - Desenvolvimento de software com metodologias ágeis / 2021 Porto Alegre Grupo A 2021 (DIGITAL)
  - COHN, Mike. Desenvolvimento de software com Scrum / 2011 Porto Alegre Bookman 2011 (DIGITAL)
  - OKUYAMA, Fabio Yoshimitsu. Desenvolvimento de software i conceitos básicos - 1 / 2014 Porto Alegre Bookman



# Avaliação

- *1º Bimestre*

*1,0 - Atividade de Estudo Programada.*

*1,0 - Prova Integrada.*

*8,0 - Avaliação Prática e Teórica*

# Avaliação

- ***2º Bimestre***

- 1,0 - Atividade de Estudo Programada.*

- 1,0 - Prova Integrada.*

- 8,0 - Avaliação Prática e Teórica*



## History of Popular Programming Languages: A timeline



# O QUE É TESTE DE SOFTWARE?

# TESTE

- Tudo o que consumimos no dia a dia é testado antes de chegar às prateleiras
- Seja para garantir nossa segurança, como no caso de produtos físicos, seja para garantir nossa saúde, como no caso de alimentos
- Programas de computador também são testados

# TESTE DE SOFTWARE

## Definição técnica:

“Teste de software é todo e qualquer **procedimento** que ajuda a determinar se o programa **atinge** as **expectativas** para as quais **foi criado** (BRAGA, 2016 apud NETO, 2010).”

Exemplo: Editor de texto

Expectativa: criar ou editar textos.

# TESTE DE SOFTWARE

Testar um software consiste em:

- Verificar se ele atende às expectativas
- Se seu funcionamento é limpo, amigável e correto
- Se ele se enquadra no ambiente para o qual foi projetado

# TESTE DE SOFTWARE

O principal objetivo do teste de software é assegurar a qualidade do produto final



# TESTE DE SOFTWARE

- O teste tenta garantir que o programa funcione conforme o esperado, seja seguro, confiável e atenda às necessidades dos usuários

# TESTE DE SOFTWARE

- Teste de software é uma das principais atividades empregadas para garantir a qualidade do software em desenvolvimento
- Segundo a literatura, pode consumir mais de 50% do custo total de desenvolvimento

## Padrão IEEE 610.12

- Defeito fault: é um passo, processo ou definição de dados que está incorreto em um programa de computador
- Engano mistake: é entendida como uma ação humana que produz um defeito
- Erro error: é um estado interno incorreto de um programa que é a manifestação de algum defeito
- Falha failure: é quando o erro se propaga para a saída do programa, levando a uma saída diferente da esperada, ou seja, o programa produz um comportamento incorreto em relação a sua especificação

# DEFEITO

- Defeito (bug ou fault): É um problema ou imperfeição no código ou na lógica do software que causa um comportamento indesejado

# DEFEITO

Resultado de um código mal escrito

Implementação equivocada

Causa anomalia no funcionamento do sistema

O usuário final normalmente não vê o defeito propriamente dito

Os defeitos podem permanecer no software até que sejam descobertos e corrigidos por meio de atividades de teste

# ERRO

Falha humana e produz resultado incorreto

Exemplo: a falha na escrita de um código específico

É o ato humano de cometer um engano ou incorreção durante o desenvolvimento do software

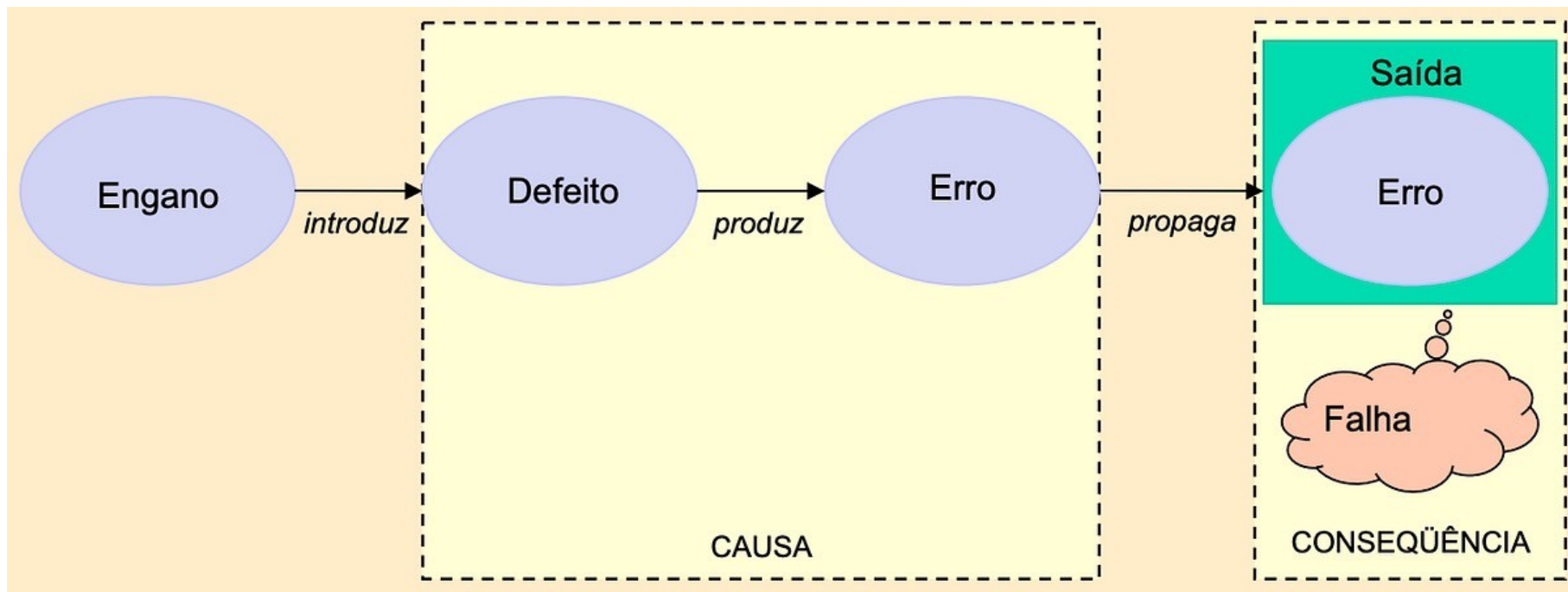
# FALHA

É a ocorrência indesejada de um defeito durante a execução do software em um ambiente de produção ou em condições reais de uso

# FALHA

Quando um defeito é ativado em tempo de execução, ele pode levar a falhas no software, resultando em comportamentos inesperados, travamentos, resultados incorretos ou outros problemas que afetam a funcionalidade do programa







# DEFEITO ERRO FALHA

<https://github.com/JoaoChoma/testedesoftware2025/tree/main/SEMANA01>



## RAZÕES PARA TESTAR

Demonstrar ao desenvolvedor e ao cliente que o software atende aos seus requisitos

Deve haver pelo menos um teste para cada requisito no documento de requisitos

Deve haver testes para todas as características do sistema que serão incluídas no lançamento do produto

## RAZÕES PARA TESTAR

Encontrar entradas ou sequências de entradas nas quais o software se comporta de modo incorreto, indesejável ou fora da conformidade de suas especificações

Esses fatores são causados por defeitos (bugs), e o teste de software busca encontrá-los

# REFLEXÃO

Os testes não conseguem demonstrar que o software está livre de defeitos ou que vai se comportar de acordo com a sua especificação em qualquer circunstância

Sempre é possível que um teste negligenciado descubra mais problemas com o sistema

## REFLEXÃO

Edsger Dijkstra (1972): ***“O teste só consegue mostrar a presença de erros, não a sua ausência.”***

## REFERÊNCIAS

Ian Sommerville – Engenharia de Software. 10ª Edição. São Paulo: Pearson Education do Brasil, 2019.

Roger S. Pressman – Engenharia de software: uma abordagem profissional. 7ª Edição. Porto Alegre: AMGH Editora Ltda, 2011.

Shari Lawrence Pfleeger – Engenharia de Software: teoria e prática. 2ª Edição. São Paulo: Pearson Education do Brasil, 2004.