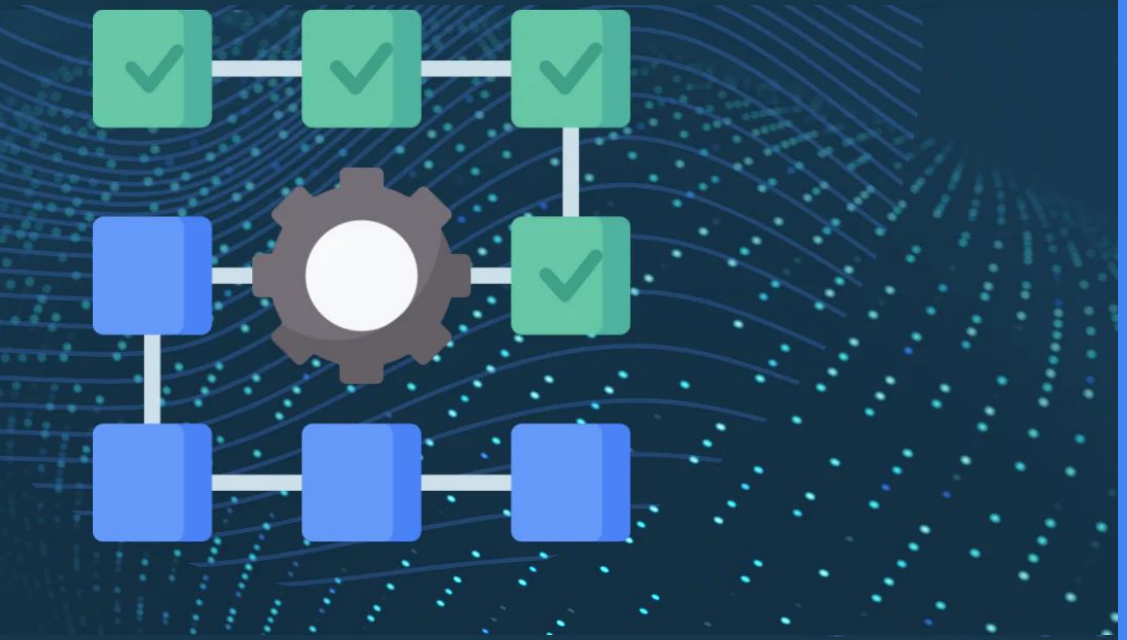
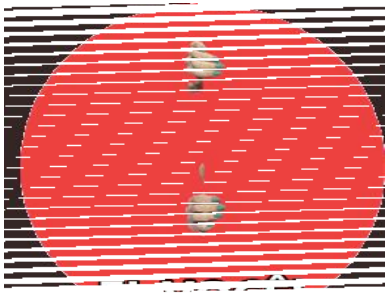


Projeto, implementação e Teste de Software

01 - Introdução a Disciplina

Prof. Esp. Dacio F. Machado


LOADING...



E vocês ?

Nome ? Idade ? Estado Civil ? CPF ?

Trabalha na área ?
Qual área/especificidade quer seguir ?
Qual sua linguagem de programação preferida ?
Uma coisa que gosta no curso / Uma coisa que odeia ?

Joga algum Jogo, Qual ?
Qual era seu desenho favorito na infância ?

Ementa

Disciplina	Distribuição de Carga Horária					
	Teórica	Prática			EAD	Total
PROJETO, IMPLEMENTAÇÃO E TESTE DE SOFTWARE	40	40				80

Ementa

Princípios e técnicas de teste de software. Verificação e Validação. Planos de teste. Documentação de estratégias de testes e outros artefatos. Revisão de software. Testes de unidade. Técnicas de teste funcional (caixa preta). Técnicas de teste estrutural (caixa branca). Testes de integração. Desenvolvimento de casos de teste. Testes de sistema. Testes de aceitação. Testes de atributos de qualidade. Testes de regressão. Testes automatizados e ferramentas de apoio. Teste em ferramentas de integração contínua. Desenvolvimento dirigido por testes. Gerenciamento do processo de teste. Registro e acompanhamento. Conceitos de qualidade de software. Padrões de qualidade de software. Visão geral de CMMI e MPS-BR.

Bibliografia

Bibliografia Básica

FREITAS, Janaína Aparecida de. **Projeto, Implementação e Teste de Software.** CENTRO UNIVERSITÁRIO DE MARINGÁ. Núcleo de Educação a Distância; Reimpressão - 2018.

Maringá-Pr.: UniCesumar, 2016.

FILHO, Wilson de Pádua P. **Engenharia de Software - Produtos** - 4. ed. - Rio de Janeiro : LTC, 2019.

POLO, Rodrigo Cantú. **Validação e teste de software**. Curitiba: Contentus, 2020.

PRESSMAN, Roger S; Maxim, Bruce R. **Engenharia de software: Uma abordagem profissional. 8. ed. – Porto Alegre: AMGH, 2016.**

Bibliografia Complementar

GONÇALVES, Priscila de, F. et al. **Testes de software e gerência de configuração**. Grupo A, 2019.

LAMOUNIER, Stella Marys D. **Teste e controle de software: técnicas e automatização** Editora Saraiva, 2021.

SOUZA, Luana; MIRANDA,Erica; LUCENA,Márcia; GOMES, Apuena. **Desafios e Práticas da Engenharia de Requisitos no Contexto de Fábrica de Software com foco na Documentação e Gestão do Conhecimento.** Cadernos do IME : Série Informática : Vol. 42 : Julho 2019. P.98-115.

SANTOS, Kleoson B. C.; OLIVEIRA, Sandro R. B**. Um Estudo de Caso de Aplicação de um Método Ágil para Desenvolvimento de Requisitos de Software: O REACT**. Cadernos do IME : Série Informática : Vol. 41 : Julho 2018. P.06-021.

ZANIN, Aline; JÚNIOR, Paulo A P.; ROCHA, Breno C.; et al. **Qualidade de software**. Grupo A, 2018.

Cronograma da Disciplina

1º	Apresentação da disciplina. Princípios e técnicas de teste de software	Teoria
2º	Princípios e técnicas de teste de software	Teoria
3º	Verificação e validação	Teoria
4º/5º	Verificação e validação	Prática
6º	Planos de Teste	Teoria
7º/ 8º	Planos de Teste	Prática
9º	Documentação de estratégias de testes e outros artefatos.	Teoria
10º/ 11º	Documentação de estratégias de testes e outros artefatos.	Prática
12º	Revisão de Software	Teoria
14º/ 13º	Revisão de Software	Prática

Cronograma da Disciplina

15º	Técnicas de teste funcional (caixa preta)	Teoria
16º/17º	Técnicas de teste funcional (caixa preta)	Prática
18º	Técnicas de teste estrutural (caixa branca)	Teoria
19º/20º	Técnicas de teste estrutural (caixa branca)	Prática
21º	Teste de unidade	Teoria
22º/ 23º	Teste de unidade	Prática
24º	Teste de unidade - Casos de teste	Teoria
25º/ 26º	Teste de unidade - Casos de teste	Prática
27º/28	Testes de aceitação	Teoria / Prática
29º	PROVA DO PRIMEIRO BIMESTRE	Prática

Método de Avaliação I

- *1º Bimestre*

  1,0 - Atividade de Estudo Programada.

  1,0 - Prova Integrada.

  8,0 - Avaliação do Professor

  Prova Dissertativa - 5,0

  Entrega de Prática em Laboratório - 3,0

  Total 8,0

Método de Avaliação II

- *1º Bimestre*

  1,0 - Atividade de Estudo Programada.

  1,0 - Prova Integrada.

  8,0 - Avaliação do Professor

  Projeto em Grupo de Implementação e Documentação de UMA Técnicas de Teste - 4,0

  (Testes de unidade / Casos de teste / Caixa preta / Caixa branca / Testes de Aceitação *)*

  Prática e participação em Laboratório - 2,0

  Prova Escrita -2,0

  Total 8,0

Método de Avaliação III

- *1º Bimestre*

  1,0 - Atividade de Estudo Programada.

  1,0 - Prova Integrada.

  8,0 - Avaliação do Professor

  Projeto Individual e Presencial de Implementação de um caso de Teste - 3,0

  Prática e participação em Laboratório - 2,5

  Prova Escrita -2,5

  Total 8,0

Teste ?

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

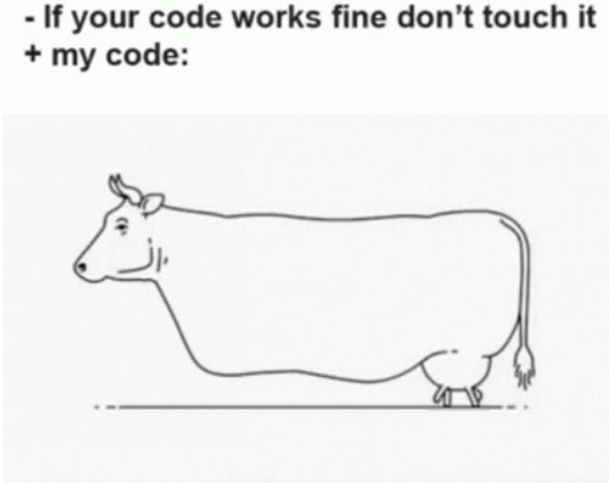
O teste tenta garantir que o programa funcione conforme o esperado, seja seguro, confiável e atenda às necessidades dos usuários

Porque Testar software ?

O teste é uma atividade de verificação e validação do software e consiste na análise dinâmica, isto é, na execução do produto de software com o objetivo de verificar a presença de defeitos no produto e aumentar a confiança de que está correto.

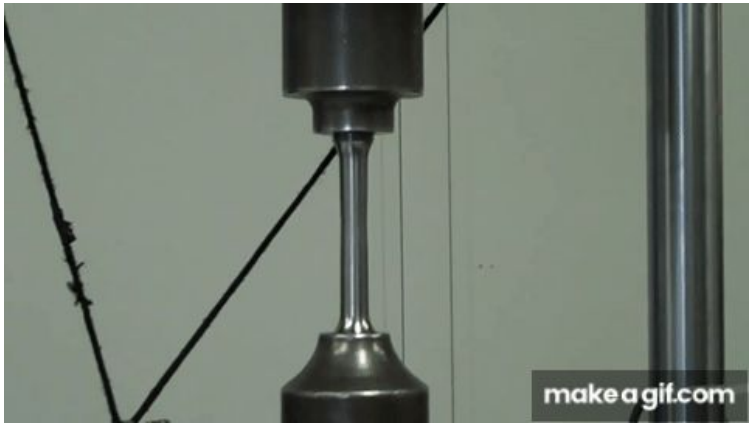
Os testes não conseguem demonstrar que o software está livre de defeitos ou que vai se comportar sempre de acordo com a sua especificação, em qualquer circunstância.

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



Porque Testar software ?

Porque testamos **TUDO**.



Porque é provável que o software possua defeitos

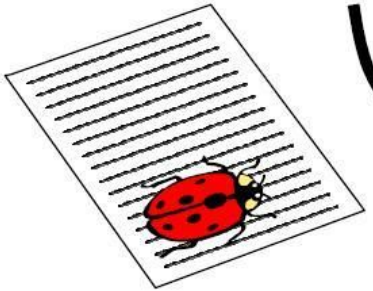
Muitos fatores podem ser identificados como causas de tais problemas, mas a maioria deles tem uma única origem: erro humano "(DELAMARO; MALDONADO; JINO, 2007,).

Erro

Uma pessoa comete um erro ...

Defeito

... que cria um defeito no software...

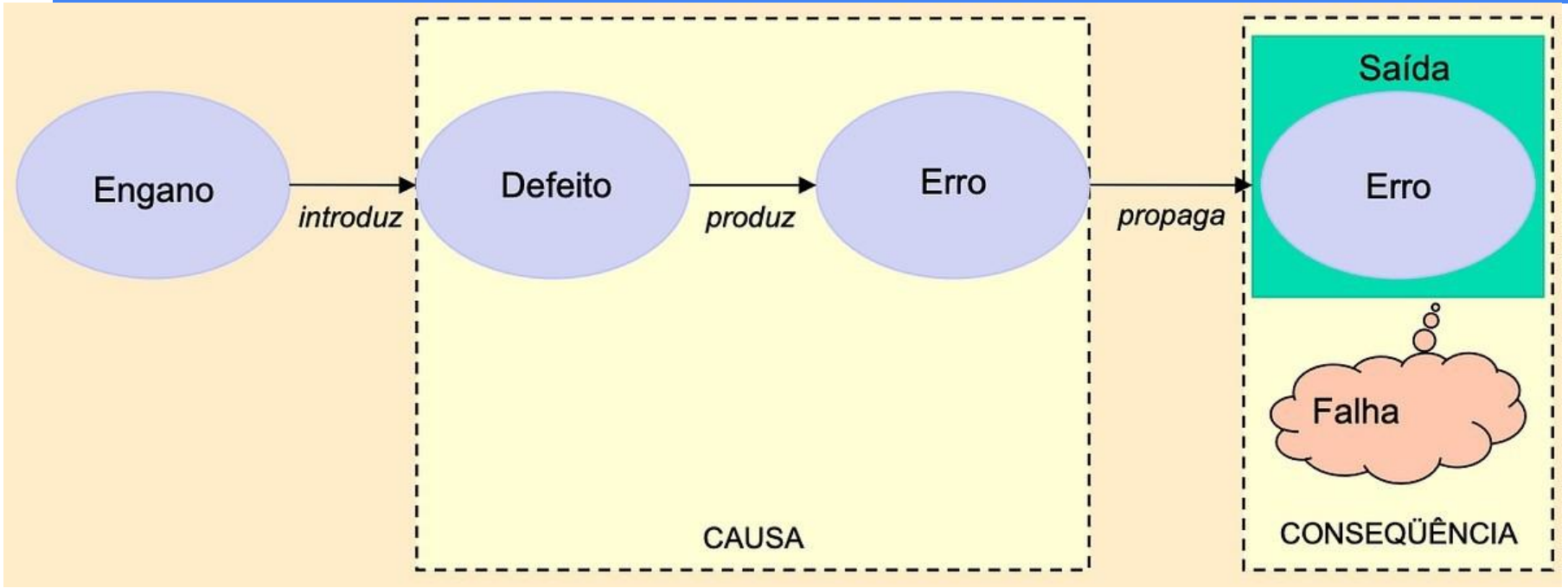


Falha

... que pode causar uma falha na operação

Erro, Defeito e Falha

- **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



- Erro: é uma ação humana que produz um resultado incorreto.
- Defeito: A manifestação de um erro no software. Também conhecido como Bug
- Falha: quando o sistema se comporta de forma inesperada devido ao defeito

Porque falhas podem custar muito caro.

O custo da correção de um defeito tende a ser cada vez maior quanto mais tarde ele for descoberto. [Myres, 2004]

Em 1996 - Um software com uma exceção não tratada foi responsável pela explosão do foguete Ariane-5, quando a 40 seg após a iniciação da seqüência de vôo, o foguete se desviou de sua rota, partiu e explodiu, tendo um prejuízo de 500 milhões de dólares

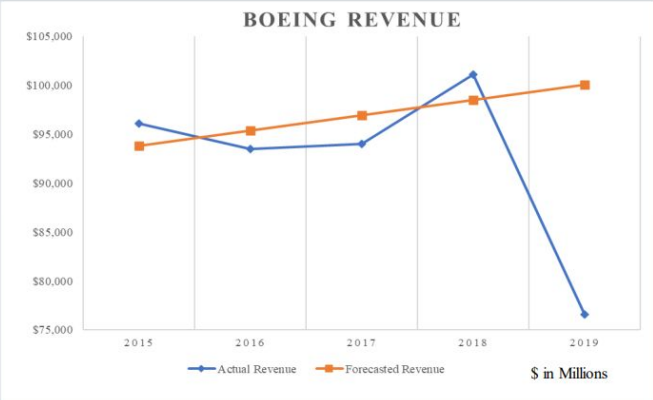
3ª Guerra Mundial (Quase!) (1983) Custo: Quase toda a humanidade
O sistema de alerta precoce soviético falsamente indicou que os EUA tinham lançado cinco mísseis balísticos. Felizmente, o oficial de serviço soviético tinha uma “sensação esquisita no estômago” e fundamentalmente, se os EUA estavam realmente atacando, eles lançariam mais de cinco mísseis, por isso ele relatou o aparente ataque como um alarme falso. Causa: Um bug no software soviético falhou ao detectar reflexos solares como falsos mísseis.

A Case Study of Management Shortcomings: Lessons from the B737-Max Aviation Accidents

Shivanee Cannon-Patron, Shakaynah Gourdet, Fathima Haneen, Carlos Medina, Sammy Thompson III
Faculty Advisor: Dr. Sohel M. Imroz – BA 520

Financial Toll

- Based on actual revenue earned between 2015-2018, Boeing's projected gain was $100,073 for 2019. As a result of the B737-Max mishap, Boeing collected only $76,559 (23.50%). In addition, the cost to shut down and restart the B737-Max production line exceeded $4 Billion.



TOTAL LOSS = $27,514,000,000

Grounding and Accident Investigation

- The Federal Aviation Administration (FAA) grounded the entire B737-Max fleet for over 20 months following the two accidents.
- The cause of both crashes were linked to unintended activation of the MCAS system. The AOA system was identified as a single point of failure in the design. Assumptions about pilot responses turned out to be incorrect, and insufficient training was provided to pilots to inform them of the differences of the B737-Max, the MCAS system, and proper procedures to regain control in the event of a malfunction.

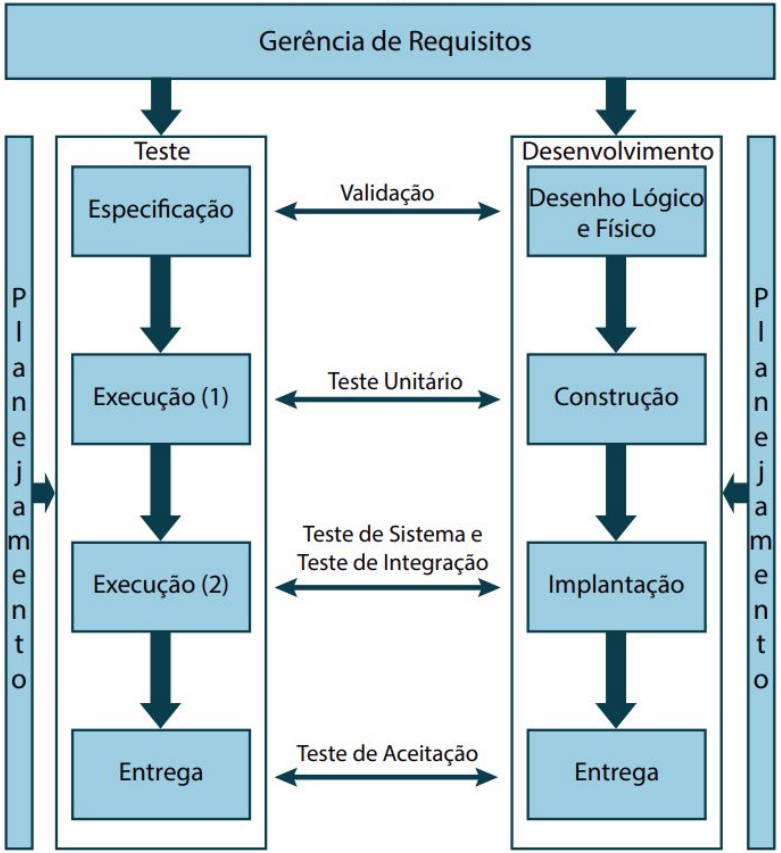
Boeing 737 MAX LIBERADO para voar EP. 713

https://www.youtube.com/watch?v=gU_LzoieJyg

Testes e Desenvolvimento

- 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

Modelo de Integração entre os processos de desenvolvimento e teste



Níveis de Teste

- **Teste de Unidade / Unitário**
- **Teste de Componentes**
- **Teste de Integração**
- **Teste de Sistema**
- **Teste de Regressão**
- **Teste de Aceitação**

Técnicas de Teste

- **Teste Caixa-Branca / *White-Box* / Estrutural**
- **Teste Caixa-Preta / *Black-Box* / Funcional**

Tipos de Teste

- **Teste de Usabilidade**
- **Teste de Desempenho / *Performance***
- **Teste de Carga**
- **Teste de Estresse / Esforço**
- **Teste de Segurança**

Reflexão

Edsger Dijkstra (1972):

"O teste só consegue mostrar a presença de erros, não a sua ausência."

OBRIGADO

daciofmf@gmail.com