
**UnB Gama**
O novo endereço da Tecnologia.

Conceitos Fundamentais




Teste de software


Semestre 1/2018

Profs. Ricardo Ajax

1

**UnB Gama**
O novo endereço da Tecnologia.



Verificação e Validação



- Conjunto de atividades conduzidas ao longo do ciclo de vida de um projeto de software para garantir a qualidade
- Ocorrem em cada estágio do processo de software
- Ocorrem continuamente, “Estilo de vida!”.

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

2

**Verificação e Validação**



- Conjunto de atividades conduzidas ao longo do ciclo de vida de um projeto de software para garantir a qualidade
- Ocorrem em cada estágio do processo de software
- Ocorrem continuamente, “Estilo de vida!”.

Objetivo

Estabelecer confiança de que o software está adequado ao seu propósito!

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

3

**Verificação e Validação**



No formal:

Verificação
Objetiva confirmar (ou não) a fidelidade às especificações técnicas dos requisitos do produto entregue.
Realizada usualmente em fases do desenvolvimento para avaliar se o produto foi feito corretamente. - *Boehm, 1979*

Mas o que é o produto?

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

4

**Verificação e Validação**

No formal

Validação


A validação está voltada para os critérios de aceitação do cliente, ou seja, tem o objetivo de avaliar se as necessidades (características, requisitos) solicitados pelo usuário (Cliente) estão compreendidas no produto entregue

Então?... Estamos construindo o produto correto (solicitado pelo usuário?) - *Boehm, 1979*

Mas o que é o produto?

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

5

**Processos de Engenharia e Verificação**

- Na engenharia tradicional repetição em altos níveis de automação nas linhas de produção reduzem a necessidade de verificação em produtos individuais
 - Poucos componentes são verificados individualmente
 - Os produtos finais são testados estatisticamente
- Mas alguns produtos, pela sua especificidade e complexidade, requerem técnicas de verificação mais sofisticadas: carros, avioes, computadores...e...

Softwares

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

6

UnB Gama O novo endereço da Tecnologia. **Processos de Engenharia e Verificação**

- Produção: Projetos ou serviços continuados gerando itens iguais (ou semelhantes) → verificação por amostragem estatística.
- A verificação se torna mais difícil de acordo com a complexidade e variedade dos produtos (a verificação é muito mais do que simplesmente uma medida de tamanho, uma avaliação de uma única especificação como cor, textura, peso, etc) → possivelmente a verificação se dará caso-a-caso



Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

7

UnB Gama O novo endereço da Tecnologia. **Software?**


Mas...O que é software?

Entende-se software ou produto de software (sinônimo) é o conjunto de programas de computador, procedimentos e possível documentação, além de dados associados

Glossário do IEEE

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

8



Software?


Mas...O que é software?

Entende-se software ou produto de software (sinônimo) é o conjunto de **programas** de computador, procedimentos e possível **documentação**, além de **dados** associados

Glossário do IEEE

Se Tudo isso é passível de problemas, Então deve ser avaliado (verificado / validado), ou seja, testado.

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)




Raizes do software?

- Requisitos
 - Funcionais
 - Não funcionais
 - De qualidade




Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)



UnB Gama
O novo endereço da Tecnologia.


Raizes do software?



- Tecnologias adotadas
 - Métodos
 - Técnicas
 - Ferramentas
 - Arquiteturas
 - Linguagens
 - Padrões
- Processos


Pode gerar Problemas?

É foco de teste!




Disciplina 206580 – Teste de Software – Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

11



UnB Gama
O novo endereço da Tecnologia.

Processos de Engenharia e Verificação



- Software é um dos produtos de engenharia mais **variáveis e complexos**
- Requisitos de qualidade em **ambientes diferentes** podem ser completamente distintos e incompatíveis
- Novas abordagens de desenvolvimento introduzem **novos tipos de defeitos**
- O desafio é escolher o **conjunto adequado de técnicas** para alcançar o nível de qualidade necessário, dentro das restrições de custo
- Não há “receita de bolo”** Verifica-se o que é necessário verificar, evitando desperdícios de recursos

Disciplina 206580 – Teste de Software – Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

12

UnB Gama O novo endereço da Tecnologia.

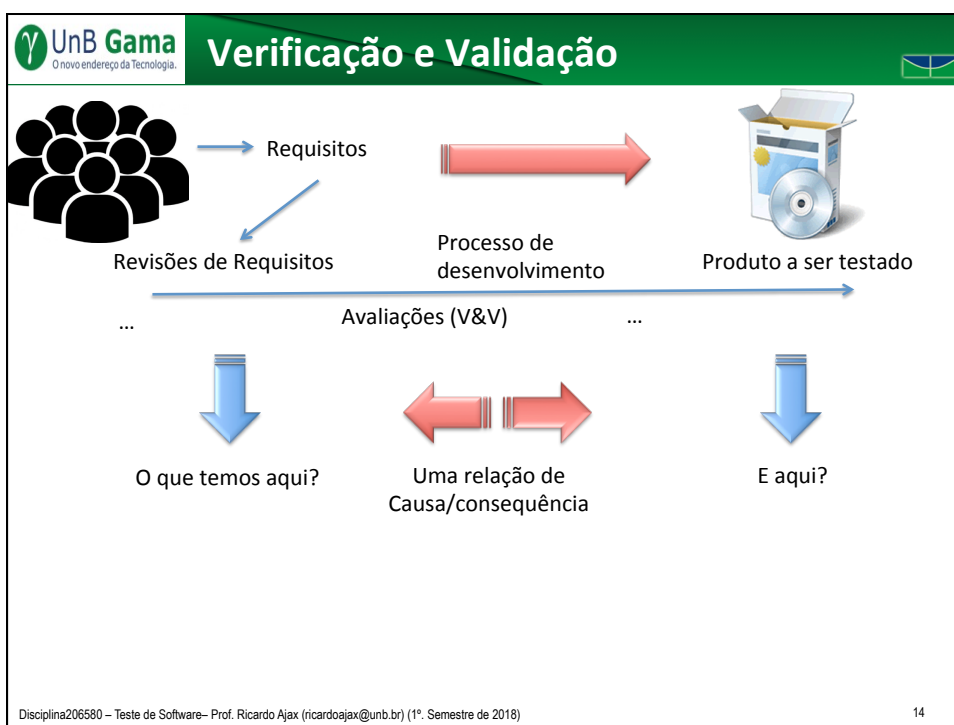
Perguntas envolvidas com V&V?

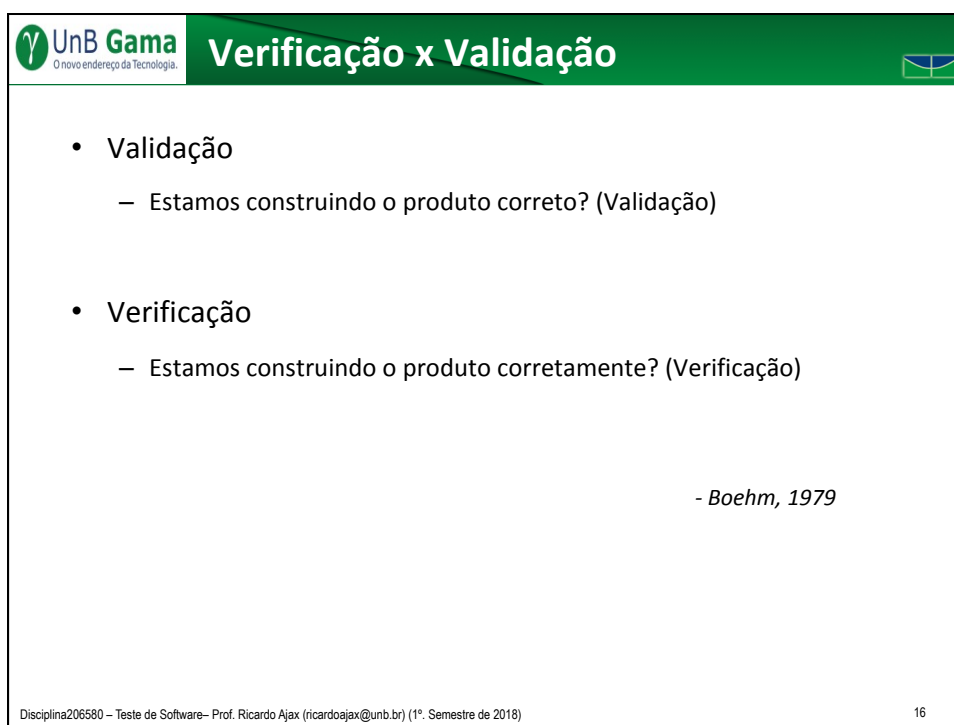
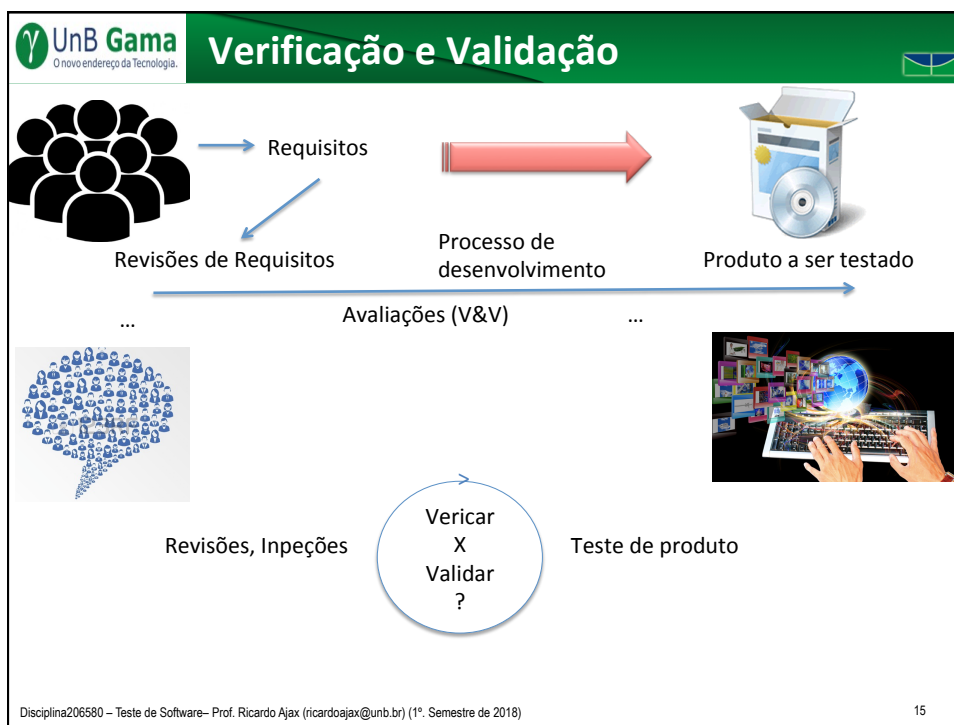
Algumas respostas possíveis de serem obtidas usando Testes:

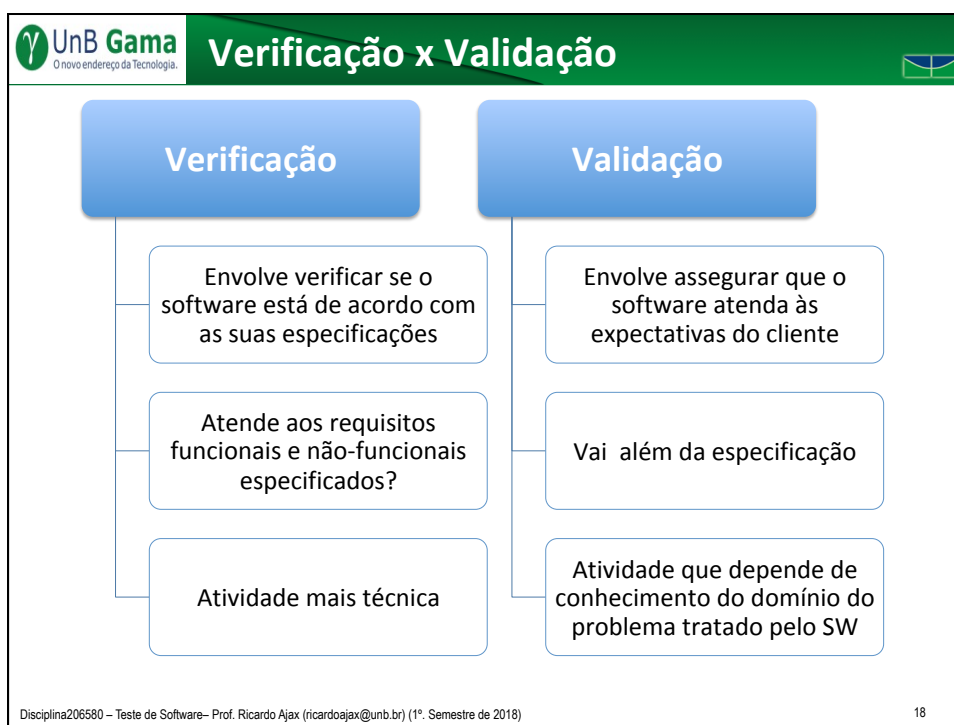
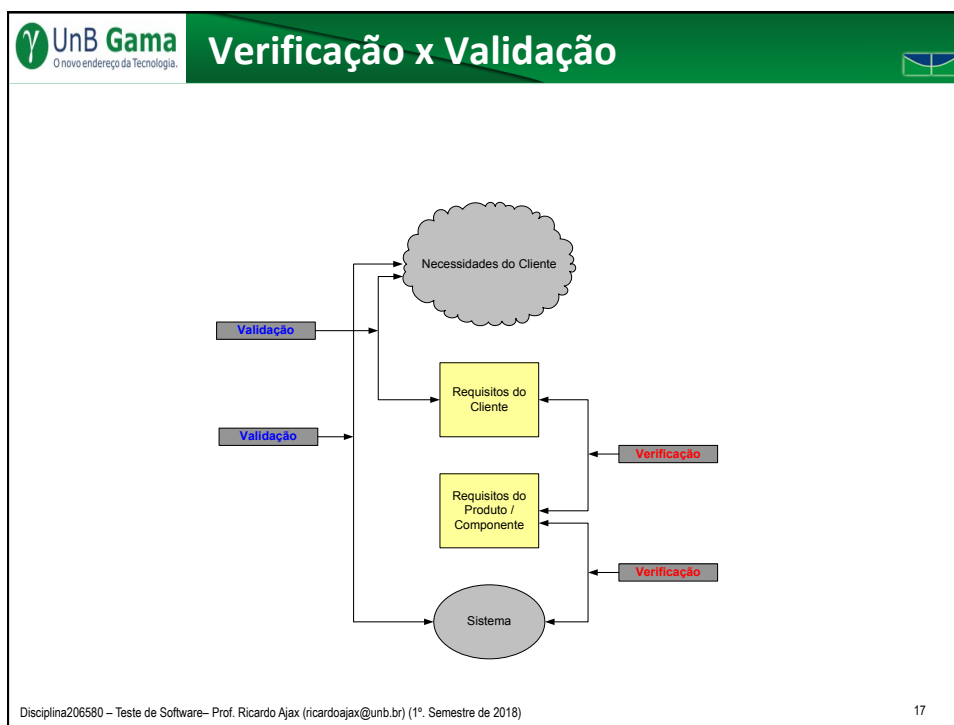
- Que técnicas devem ser utilizadas durante o desenvolvimento do produto para obter qualidade aceitável a um custo viável para a organização desenvolvedora?
- Como avaliar se um produto de uma release está pronto?
- Como controlar a qualidade de releases sucessivas?
- Como o processo de desenvolvimento pode ser melhorado ao longo dos projetos para melhorar os produtos e tornar os custos da verificação mais efetivos?



Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

13










Verificação


- Por que realizar Verificação?
 - 40% a 50% do esforço de um projeto é gasto em **retrabalho**
 - O custo de correção de defeitos **cresce** a medida que o projeto avança pelas fases de desenvolvimento
 - O custo de corrigir um defeito aumenta o equivalente a 1,6 horas se o defeito é descoberto na fase de codificação.
 - Se descoberto na fase de testes esse valor aumente para algo em torno de 12 horas
 - Se descoberto na manutenção este custo pode chegar a ser > que o equivalente a 25 horas.



Sutherland, J., Jakobsen, C.R., Johnson, K.. (2008), Scrum and CMMI level 5: The magic potion for code warriors,

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018) 19


Verificação


- Por que realizar Verificação?
 - 40% a 50% do esforço de um projeto é gasto em **retrabalho**
 - + esforço → + horas trabalhadas → +custo do projeto
 - O custo de correção de defeitos **cresce** a medida que o projeto avança pelas fases de desenvolvimento
 - Quanto mais cedo se descobrir os defeitos melhor será o desenvolvimento do software
 - Esforço, prazo, custo, produtividade ↔ Satisfação do cliente**

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018) 20



**Verificação**

- Continuando Definições:
 - A confirmação dos resultados se dá por **exame** e **fornecimento de evidência objetiva**, do atendimento aos **requisitos especificados**.
 - O objetivo da verificação é determinar se os produtos de software de uma atividade atendem completamente aos requisitos ou condições impostas a eles nas atividades anteriores.

[ISO/IEC 12207]

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)


21

**Verificação**


- Objetivos:
 - assegurar que os requisitos estabelecidos podem ser alcançados
 - identificar os requisitos que não podem ser alcançados
 - garantir que o software é desenvolvido de forma uniforme
 - identificar erros para tomar medidas corretivas o mais cedo possível
 - tornar o projeto mais gerenciável

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

22

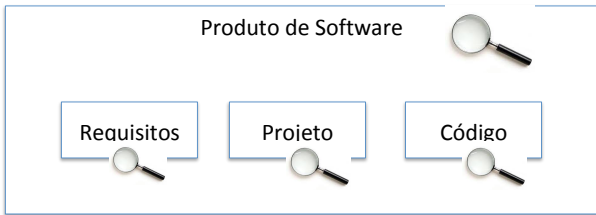
 **UnB Gama**
O novo endereço da Tecnologia.

Verificação



- Todos os artefatos de um projeto podem ser submetidos à verificação?


Produto de Software




- Devem!

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

23

 **UnB Gama**
O novo endereço da Tecnologia.

Validação




- Frequentemente, a **validação** de software está fortemente associada à **verificação** de software
- Normalmente, elas são executadas em conjunto, pois muitas vezes é difícil determinar onde uma começa e onde a outra termina


Exemplo: Testes unitários podem ser feitos para Validar?

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

24



Validação




- Continuando Definições:
 - Validação se refere ao processo de examinar um produto para determinar sua **conformidade** com as **necessidades do usuário**. [ISO/IEC 12207]


E se não estiver conforme?

- Evidências objetivas:** como evidenciar um problema?
 - Resultados de avaliações (testes ou técnicas estáticas)
 - Medições, Medições, Medições
 - Para melhorar, melhorar, melhorar
 - Porque melhorar?
- Esta norma define que a validação é feita normalmente no produto final sob condições de operação definidas, podendo, contudo, tornar-se necessária em fases anteriores (protótipos ou mesmo definições do software a ser construído).

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018) 25




Abordagens de V&V




- Estáticas:
 - Não** requerem a execução ou existência de um programa executável
 - Exemplos:
 - Revisões
 - Inspeções
 - Walkthrough
 - Auditorias
- Dinâmicas:
 - Baseiam-se na execução de um programa
 - Exemplo: Teste de Software (existem vários tipos de testes)

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018) 26

**UnB Gama**
O novo endereço da Tecnologia.


Teste




- Definições:
 - “Teste é o processo de **executar** um programa com a intenção de **encontrar erros.**” - *Myers, 1979*
 - “Avaliação do software pela **observação de sua execução.**” – Ammann & Offutt, 2008

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

27

**UnB Gama**
O novo endereço da Tecnologia.



Teste



- A maioria das definições associa o teste às **técnicas dinâmicas** de Verificação e Validação
- Porém, alguns autores definem teste como o processo de encontrar casos em que o programa não atende a sua especificação
 - Nestes casos, atividades como revisões, inspeções e análise estática de código são também chamadas de teste (teste estático)

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

28


Noções de qualidade de software




Erro, defeito ou falha ?


Erro: uma ação humana que pode resultar em um produto com defeitos. Um erro pode levar a inclusão de uma falha no sistema, fazendo com que o mesmo não funcione corretamente ou apresente resultados incorretos;

Defeito: em um sistema pode ocorrer devido a omissão de informações, definição de dados ou comandos / instruções incorretas nos programas, dentre outros fatores. Um defeito é uma consequência de um erro cometido. Se um determinado defeito não for encontrado, ele pode causar uma falha no funcionamento do software.

Falha: De acordo com o IEEE uma falha ocorre quando um programa, ou conjunto de programas (sistema) não se comporta conforme o esperado ou apresenta resultados diferentes dos planejados (previstos). Assim, uma falha é considerada uma propriedade do sistema em execução (operando).

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)
29


Noções de qualidade de software




```

graph LR
    Erro[Erro] --> Defeito[Defeito]
    Defeito --> Falha[Falha]
  
```

Erro

- O ser humano está sujeito a cometer um erro (engano)

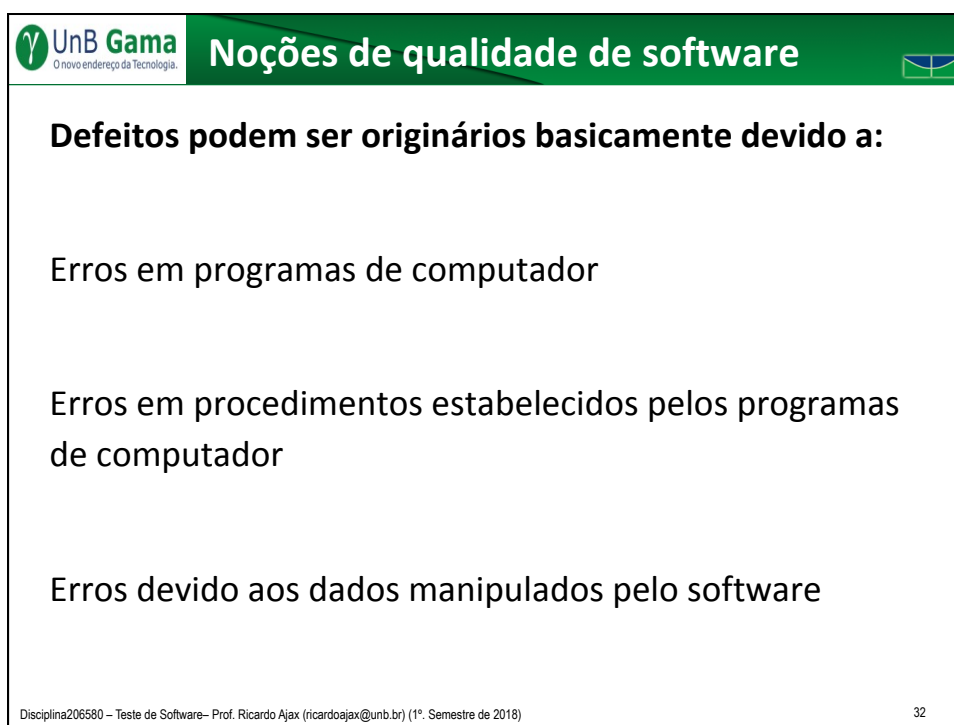
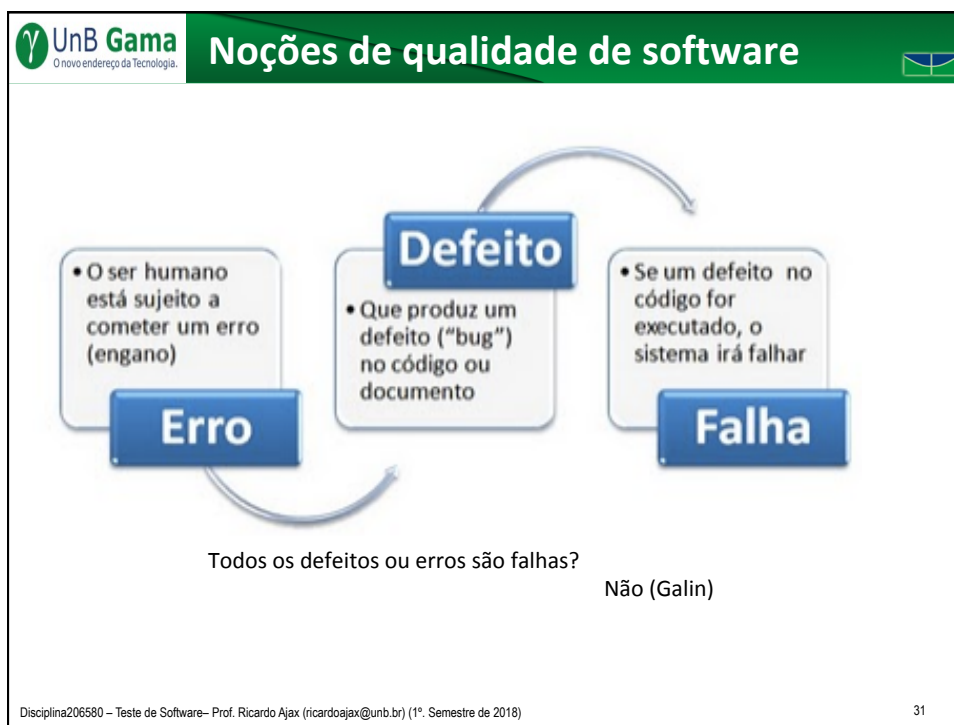
Defeito

- Que produz um defeito ("bug") no código ou documento

Falha

- Se um defeito no código for executado, o sistema irá falhar

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)
30



UnB Gama
O novo endereço da Tecnologia.

Noções de qualidade de software

Defeitos podem ser originários basicamente devido a:

- Erros em programas de computador
- Erros em procedimentos estabelecidos pelos programas de computador
- Erros devido aos dados manipulados pelo software

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

33

UnB Gama
O novo endereço da Tecnologia.

Noções de qualidade de software

Defeitos podem ser originários basicamente devido a:

- Erros em programas de computador
- Erros em procedimentos estabelecidos pelos programas de computador
- Erros devido aos dados manipulados pelo software

E por que eles acontecem?

Como evitar que eles aconteçam?

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

34

UnB Gama
O novo endereço da Tecnologia.

Noções de qualidade de software

Defeitos podem ser originários basicamente devido a:

- Erros em programas de computador
- Erros em procedimentos estabelecidos pelos programas de computador
- Erros devido aos dados manipulados pelo software

```

graph LR
    A([Causas humanas (erros)]) --> B([Defeitos])
    B --> C([Falhas do software])
  
```

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

35

UnB Gama
O novo endereço da Tecnologia.

Noções de qualidade de software

Causas mais comuns de erros

Requisitos:

- Definições de requisitos (dúbios ou incompletos)
- Ausência de requisitos críticos (ex: qualidade)

```

graph LR
    A([Causas humanas (erros)]) --> B([Defeitos])
    B --> C([Falhas do software])
  
```

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

36

UnB Gama
O novo endereço da Tecnologia.

Noções de qualidade de software

Causas mais comuns de erros

Técnicas:

Arquiteturas mal definidas, incompletas ou indefinidas causando dúvidas de codificação

Regras de negócios inconsistentemente implementadas

Dados não confiáveis (falta de críticas aos dados necessários à aplicação)

```

graph LR
    A([Causas humanas (erros)]) --> B([Defeitos])
    B --> C([Falhas do software])
  
```

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

37

UnB Gama
O novo endereço da Tecnologia.

Noções de qualidade de software

Causas mais comuns de erros

Códigos:

Uso de estruturas mal estabelecidas

Algoritmos pouco experimentados

Especificações imprecisas sobre tratamento de erros de usuários (requisitos)

Falta de experiência dos programadores



Linguagens ou arquiteturas pouco conhecidas

```

graph LR
    A([Causas humanas (erros)]) --> B([Defeitos])
    B --> C([Falhas do software])
  
```

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

38


Noções de qualidade de software



Causas mais comuns de erros

Documentação:

Lacunas ou erros de documentos críticos do processo de desenvolvimento podem conduzir a defeitos que, por sua vez, ao serem implementados, causem falhas no software.

Lacunas ou erros na documentação do usuário para a operação do sistema podem causar operações indevidas e falhas nos sistemas.

Manuais mal redigidos (ex.: Manuais Técnicos de instalação)





```

graph LR
    A([Causas humanas (erros)]) --> B([Defeitos])
    B --> C([Falhas do software])
  
```

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

39



Noções de qualidade de software


Antídotos

Verificação e Validação

Estáticas: Revisões técnicas

Dinâmicas: testes de software



```

graph LR
    A([Causas humanas (erros)]) --> B([Defeitos])
    B --> C([Falhas do software])
  
```

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

40

UnB Gama
O novo endereço da Tecnologia.

Noções de qualidade de software

Antídotos

Verificação e Validação

Estáticas: Revisões técnicas

Dinâmicas: testes de software

Erros em testes ou revisões técnicas podem fazer com que defeitos passem do desenvolvimento para a operação do software causando falhas

```

graph LR
    A([Causas humanas (erros)]) --> B([Defeitos])
    B --> C([Falhas do software])
  
```

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

41

UnB Gama
O novo endereço da Tecnologia.

Dúvidas?

- Ricardo Ajax
 - Ricardoajax@unb.br

Disciplina206580 – Teste de Software– Prof. Ricardo Ajax (ricardoajax@unb.br) (1º. Semestre de 2018)

42