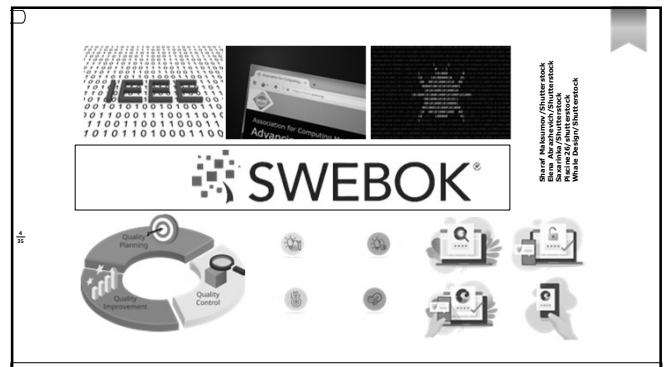
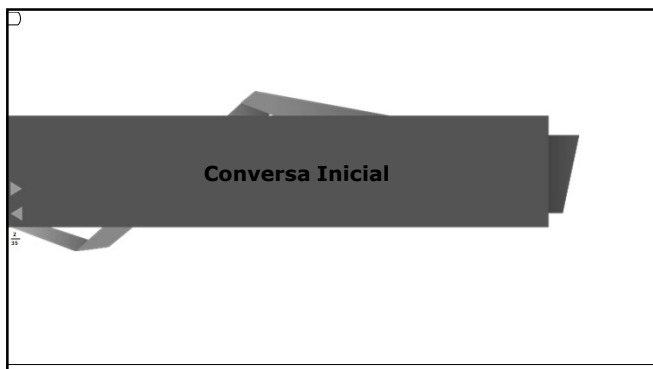




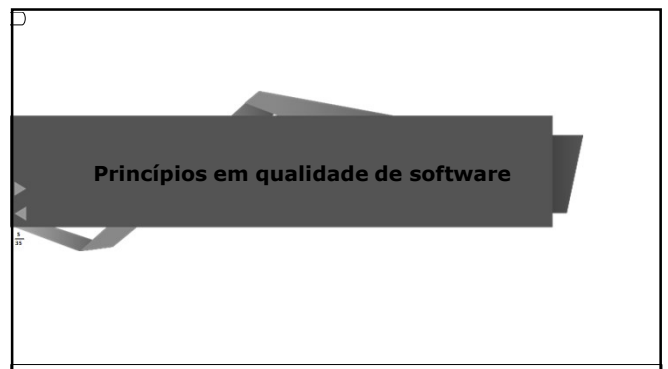
1



4



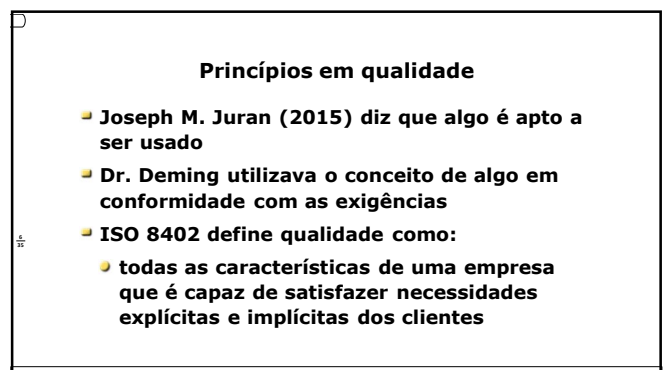
2



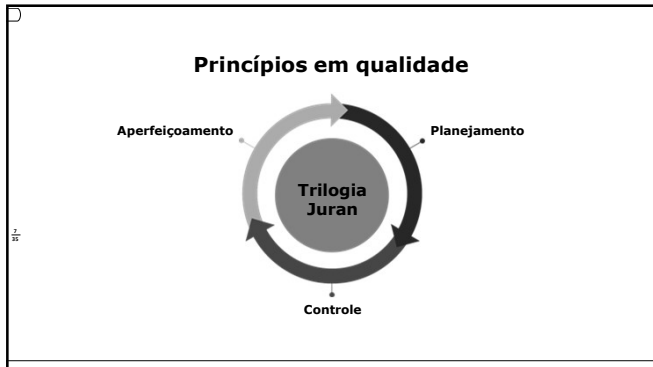
5



3



6



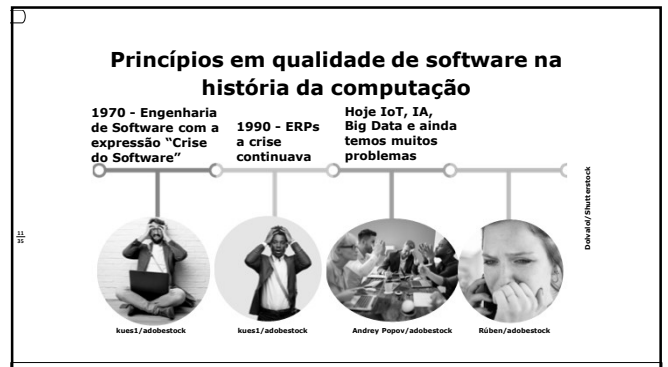
7



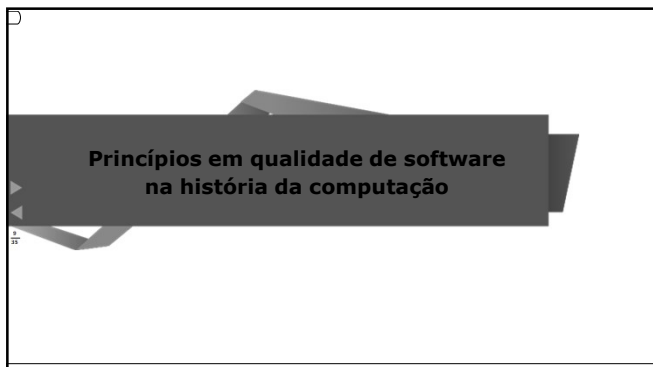
10

- Final dos anos 1980 - ISO 9000
- Redução de desperdícios
- Melhoria na eficiência de RH e equipamentos
- Análise de segurança e melhorias nas relações como mercado
- Aumento de produtividade e lucros

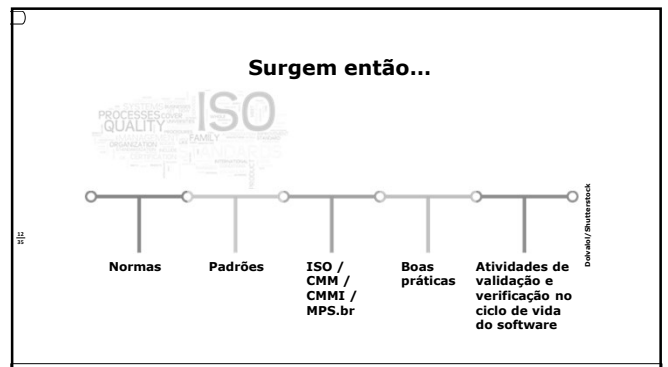
8



11



9



12

Referências importantes:

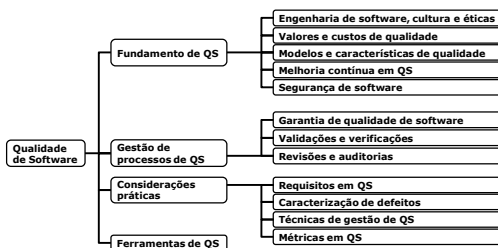
- ACM (acm.org)
- IEEE (ieee.org)
- Criação do Swebok (parceria ACM e IEEE)

13

Swebok: um guia para engenharia e qualidade de software

16

SWEBOK



14

Swebok: um guia para engenharia e qualidade de software

Fundamentos da qualidade de software:

- Engenharia de software, cultura e ética
- Custos e valores em qualidade
- Modelos e características da qualidade
- Melhorias em qualidade de software e
- Segurança de Software

17

- Qualidade de software e de projeto de software precisam eliminar o máximo possível de erros!

15

Swebok: um guia para engenharia e qualidade de software

Processos de gerenciamento da qualidade de software:

- Garantia da qualidade de software
- Validação e verificação e
- Revisões e auditorias

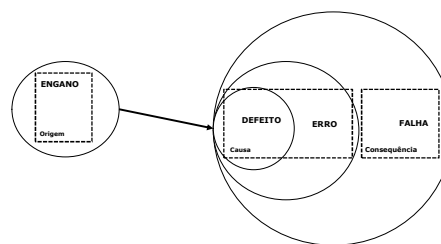
18

Swebok: um guia para engenharia e qualidade de software

- Considerações práticas:
 - Requisitos de qualidade de software
 - Caracterização de defeitos
 - Técnicas de gerenciamento de QS
 - Métricas em QS
- Ferramentas de qualidade de software

19

A qualidade de software e os BUGs



22

A qualidade de software e os BUGs

20

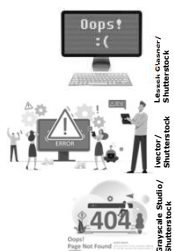
A qualidade de software e os BUGs

- Engano:** o engano é uma ação humana acidentalmente incluída dentro de uma classe, uma função, ou outro elemento qualquer de um código de programa. Por exemplo, numa fórmula de cálculo de impostos sobre vendas, podemos incluir a fórmula de forma errada ou até mesmo a fórmula errada.

23

A qualidade de software e os BUGs

- O sistema travou durante a produção
- A área de TI cometeu um erro
- Após uma revisão, encontramos um defeito no plano de teste
- Encontrei um bug em um aplicativo hoje
- O sistema quebrou
- Uma falha foi relatada no subsistema de monitoramento



21

A qualidade de software e os BUGs

- Defeito:** o defeito é a consequência de um engano cometido em um trecho de código de programa, o qual resulta em saídas inesperadas ou inconsistentes.

24

A qualidade de software e os BUGs

- **Erro:** o erro então se forma devido a um defeito causado por sua vez por um engano. E neste caso, o resultado do software é diferente do esperado

25

A importância da verificação e validação na qualidade de software

- Este tema V&V é um assunto importante dentro do pilar processos de gerenciamento de qualidade de software e estão intimamente relacionadas com testes de software

28

- **Falha:** a falha é uma consequência de um erro. Por exemplo, voltando ao cálculo dos impostos sobre vendas, que teve um defeito em decorrência da escrita de uma fórmula errada, gerou um erro que causou a falha na nota fiscal

26

A importância da verificação e validação na qualidade de software

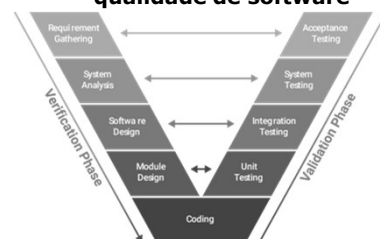
- **Verificação:** desenvolvemos o software da forma correta? Conseguimos atender aos requisitos especificados?
- **Validação:** desenvolvemos o software certo?

29

A importância da verificação e validação na qualidade de software

27

A importância da verificação e validação na qualidade de software



30

A importância da verificação e validação na qualidade de software

- A verificação pode ser pensada seguindo alguns questionamentos, tais como:
 - Arquitetura: foi avaliada? Em qual nível?
 - Requisitos: são testáveis?
 - Código: segue boas práticas? Padrões? Convenções?
 - Código: foi revisado antes de ser feito *commit* à *branch main*?

31

A importância da verificação e validação na qualidade de software

- Em relação à validação, podemos questionar:
 - Inconsistências: usuários relataram alguma inconsistência nos testes de aceitação?
 - Versionamento: a nova versão teve algum impacto negativo?
 - Documentação: manual é condizente com o software?

34

A importância da verificação e validação na qualidade de software

- A verificação pode ser pensada seguindo alguns questionamentos, tais como:
 - Testes unitários: foram executados com sucesso?
 - Integração contínua: *build* automático da pipeline quebrou após o *commit*?
 - Documentação técnica: foi produzida? (classes, modelo do banco)

32

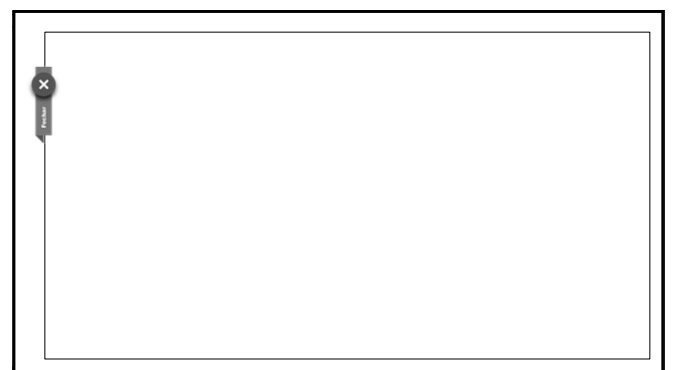
- A verificação e a validação são atividades principais dentro da garantia da qualidade, pois estabelece o que podemos fazer em cada uma das fases do desenvolvimento de software

35

A importância da verificação e validação na qualidade de software

- Em relação à validação, podemos questionar:
 - Partes interessadas: concordam com os prazos do projeto?
 - Feedback das partes interessadas: aceitou *wireframes*?
 - Demonstração: usuários avaliaram a demonstração?

33



36