

1

Qualidade 2

Tópicos

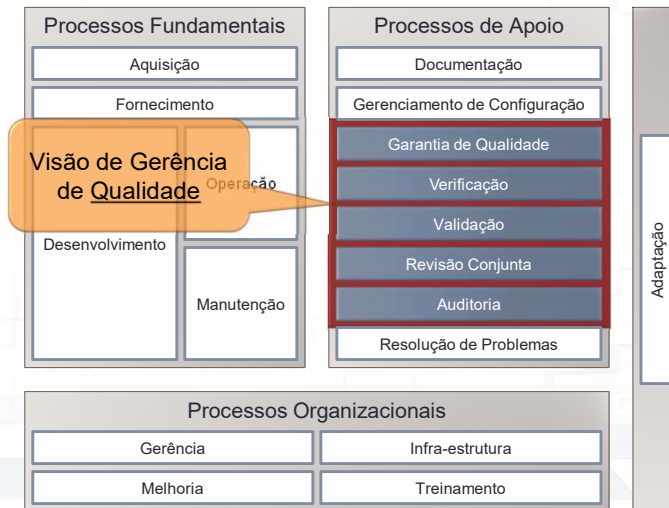
- Qualidade
- Qualidade Aplicada ao Software

2

Processo de Software

Processos Gerais do Ciclo de Vida – ISO 12207

- Norma ISO/IEC 12207
 - Outubro/1998
 - 48 processos



3

Qualidade

Visão Popular

- **Algo Abstrato:** Indefinido.
 - Algo **inatingível**, que **não pode ser medido** e nem avaliado
- **Perfeição:** imaginando que seja um conceito definitivo e imutável
 - Assim, a qualidade reflete a situação de se ter atingido um **patamar máximo**, não podendo mais ser alterada
- **Luxo e questão de gosto:** Produtos **caros**, **sofisticados** e mais **complexos** são considerados de maior qualidade que produtos similares mais simples.

4

Qualidade

Visão Profissional [1/3]

- **Conformidade aos Requisitos:** refere-se à qualidade em termos específicos, ou seja, em **itens que podem ser medidos**.
 - Os **requisitos** devem ser **claramente** expostos para que não sejam mal interpretados.
 - **Medidas** são feitas continuamente no processo de desenvolvimento, para determinar a adequação aos requisitos.
 - As **não conformidades** são consideradas como **defeitos**, como **falta de qualidade**.

5

Qualidade

Visão Profissional [2/3]

- **Adequação ao Uso** [1/2]: deve-se considerar as **especificações** e as **expectativas** dos clientes.
 - Cada um desses elementos é uma característica de qualidade e são classificados em categorias conhecidas **como parâmetros para adequação ao uso**.
 - Os parâmetros mais importantes são **Qualidade de Projeto** e **Qualidade de Adequação**.

6

Qualidade

Visão Profissional [3/3]

- **Adequação ao Uso [2/2]:**
 - **Qualidade de projeto:** refere-se ao grau em que o bem produzido está de acordo com as **especificações de projeto**.
 - **Qualidade de adequação:** refere-se aos diferentes níveis que o projeto do produto deve satisfazer a uma **necessidade humana** não relacionada à funcionalidade do produto.

7

Conceito de Qualidade

Visão Geral

- Os diferentes **enfoques** sobre a questão **qualidade** são organizados de maneira a agrupar os conceitos em **conjuntos específicos**.
- Garvin 1984 – Apresenta cinco **abordagens**
 - As abordagens proporcionam uma visão geral do conceito de qualidade em função da sua **abrangência e organização**.

8

Abordagens de Garvin

1. Abordagem Transcendental

- Trata a qualidade como um **conceito abstrato** que dificilmente pode ser fixado com precisão.
- Qualidade representa uma **característica**, uma **propriedade** ou um **estado** que torna um produto ou serviço **aceitável**,
 - embora essa aceitação não seja derivada de estudo realizado, mas da **constatação prática** e da **experiência**.
- Exemplo:
 - Os atributos estéticos do ser humano: são observáveis, mas difíceis de serem descritos.

9

Abordagens de Garvin

2. Abordagem Centrada no Produto

- Essa abordagem entende qualidade como uma **variável passível de medição** precisa.
- Diferenças de qualidade são observáveis no **produto** pela **quantidade** e **diversidade** de **elementos** ou atributos que o produto possui ou não.
 - Isso traz uma contribuição relevante para a fixação de um **conceito prático** do que seja qualidade.

10

Abordagens de Garvin

3. Abordagem Centrada no Valor

- Essa abordagem agrega qualidade ao **custo de produção**.
- Considera que um produto é de boa qualidade se apresentar alto grau de **conformidade** aos requisitos do produto por um **custo aceitável**.
- Um produto apresenta qualidade se oferecer **melhor desempenho** a um **preço reduzido**.

11

Abordagens de Garvin

4. Abordagem Centrada no Processo de Fabricação

- Essa abordagem fixa-se no **esforço** realizado em **produzir** um item de acordo com suas **especificações básicas**, determinadas durante o projeto.
- Se o **processo de fabricação** não pode **desenvolver** um produto conforme suas especificações, automaticamente, a qualidade estará **comprometida** logo no primeiro esforço para produzi-lo.

12

Abordagens de Garvin

5. Abordagem Centrada no Usuário [1/2]

- Essa abordagem concentra-se no **usuário** como fonte de toda a **avaliação** de qualidade de um produto.
 - define-se o que o **consumidor** deseja;
 - procura-se desenvolver um produto que esteja de acordo com as **vontades** do consumidor.
- A qualidade de um produto fica **condicionada** ao grau com que ele atende às **necessidades** e conveniências do **consumidor**.

13

Abordagens de Garvin

5. Abordagem Centrada no Usuário [2/2]

- Como um exemplo dessa abordagem torna-se compreensível o fato de que **produtos** com **ótimos projetos**, **bons padrões de desempenho**, **bonitas embalagens**, **sem nenhum defeito aparente**, saiam das fábricas destinadas ao sucesso (na **visão da empresa**) e acabam **não sendo comprados** pelos consumidores.

14

Qualidade Aplicada ao Software

- **Mito:**
 - Criar programas é uma arte que não pode seguir regras, normas ou padrões.
- **Causas desse mito:**
 - Produtos de software são complexos.
 - Software não tem produção em série. Custo está no projeto e desenvolvimento.
 - Software não se desgasta.
 - Software é invisível. Sua representação em grafos e diagramas não é precisa.
 - A Engenharia de Software é uma tecnologia em evolução.
 - Não há um acordo entre os profissionais sobre o que é qualidade de software.

15

Exigência de Qualidade no Software

- Com a constante **demanda** gerada pela vida moderna, cada vez mais os computadores passam a **integrar a rotina diária** e a **produção de software** vem **aumentando constantemente**.
- A **exigência por qualidade** estende-se também à **área de software**.
- Do ponto de vista dos **fornecedores** de software, qualidade não é mais um fator de **vantagem no mercado**, mas uma **condição necessária** e pode-se dizer indispensável para que seja possível competir com **sucesso**.



16

Como chegamos à Era da Qualidade

- Desde os tempos remotos, havia muitos **problemas** no desenvolvimento dos sistemas computacionais
- 1968
 - Foram reunidos 50 especialistas, cientistas e profissionais da **indústria de software** para discutir possíveis soluções para o que passou a ser conhecido como a **Crise do Software**.
 - Nesse encontro se firmou o termo **Engenharia de Software**, e foi definida formalmente a necessidade da aplicação de uma **abordagem sistemática, disciplinada e quantificável** para o desenvolvimento, operação e manutenção de produtos de software.

17

Engenharia de Software

Perspectiva Histórica

- Observando a Engenharia de Software por uma perspectiva histórica:
 - A **década de 60** e os anos que a antecedem podem ser chamados de **Era Funcional**
 - Os **anos 70**, de **Era do Método**
 - Os **anos 80**, a **Era do Custo**
 - Os **anos 90**, como a **Era da Qualidade**

18

Perspectiva Histórica

Anos 60 – A Era Funcional

- Nos **anos 60**, aprendeu-se a usar a **tecnologia da informação** para suprir as **necessidades institucionais** e começar a **integrar o software** nas operações diárias das instituições.

19

Perspectiva Histórica

Anos 70 – A Era do Método

- Nessa fase, como as organizações de software foram caracterizadas por maciços **atrasos** nos **planos** e constantes **ultrapassagens** dos **custos** planejados, a maior preocupação era **planejar** e **controlar** os projetos de software.
- Foi quando os **modelos de ciclo de vida (processos de software)**, baseados em várias fases, foram introduzidos e analisados.

20

Perspectiva Histórica

Anos 80 – A Era do Custo [1/2]

- O custo do hardware começou a cair e a tecnologia da informação se tornou acessível às pessoas, não mais apenas às instituições.
- A competição das indústrias tomou um rumo diferente, pois aplicações de baixo custo puderam ser largamente implementadas.

21

Perspectiva Histórica

Anos 80 – A Era do Custo [2/2]

- A importância da produtividade no desenvolvimento de software aumentou significativamente.
- Vários modelos de custo na Engenharia de Software foram implementados e usados.
- No final dessa década que se reconheceu a importância da qualidade de software.

22

Perspectiva Histórica

Anos 90 – A Era da Qualidade

- A **década de 90** e os anos que seguem podem, certamente, ser chamados de **Era da Qualidade**.
- Espera-se atender a **demanda dos clientes** com a crescente **exigência de alta qualidade**.



23

Interpretações de Qualidade de Software

- Qualidade é um termo que pode ter **diferentes interpretações** e para se estudar a qualidade de software de maneira efetiva é necessário, inicialmente, obter um **consenso** em relação à **definição de qualidade** de software que está sendo abordada.
- Existem muitas **definições de qualidade** de software propostas na literatura, sob diferentes pontos de vistas.

24

Definições de Qualidade de Software

*“Um produto de software apresenta qualidade dependendo do grau de **satisfação** das necessidades dos **clientes**, sob todos os aspectos do produto” [Sanders, 1994].*

*“Qualidade é a totalidade de **características** e **critérios** de um produto ou serviço que exercem suas habilidades para **satisfazer** as **necessidades** declaradas ou envolvidas” [ISO9126, 1994].*

*“Qualidade de software é a **conformidade a requisitos** funcionais e de desempenho que foram explicitamente declarados, **a padrões de desenvolvimento** claramente documentados, e **a características implícitas** que são esperadas de todo software desenvolvido por profissionais” [Pressman, 1994].*

25

Definições de Qualidade de Software

Aspectos Importantes [1/2]

- As definições enfatizam três aspectos importantes: [1/2]
 1. Os **requisitos de software** são a **base** a partir da qual a qualidade é medida.
 - A falta de conformidade aos requisitos significa falta de qualidade.
 2. **Padrões** especificados definem um conjunto de **critérios de desenvolvimento** que orientam a maneira segundo a qual o software passa pelo trabalho de engenharia.
 - Se os critérios não forem seguidos, o resultado quase que seguramente será a falta de qualidade.

26

Definições de Qualidade de Software

Aspectos Importantes [2/2]

- As definições enfatizam três aspectos importantes: [2/2]
- 3. Existe um conjunto de **requisitos implícitos** que, frequentemente, não são mencionados na especificação (por exemplo, o desejo de uma boa manutenibilidade).
- Se o software se adequar aos seus **requisitos explícitos**, mas deixar de cumprir seus **requisitos implícitos**, a qualidade do software pode ser comprometida.

27

Qualidade de Software

Existência de várias visões

- Existe, ainda, uma visão de qualidade de software do ponto de vista **gerencial** (da **organização**).
- O software desenvolvido dentro do **prazo** e do **orçamento especificados** pode ser um software de **alta qualidade**.
- Isso demonstra que, ainda dentro da **qualidade de software**, pode-se definir **várias visões** diferentes, como tem sido para a definição da qualidade como um termo geral.

28

Qualidade de Software

A qualidade depende do Ponto de Vista



Interesse fica concentrado principalmente no uso do software.

A qualidade fica mais voltada às características internas do software.

A qualidade do produto não pode ser desvinculada dos interesses da organização.

29

Qualidade de Software

Visões de Qualidade [1/2]

- **Usuário:** avalia o software sem conhecer seus aspectos internos, está apenas interessado na facilidade do uso, no desempenho, na confiabilidade dos resultados e no preço.
- **Desenvolvedores:** avaliam aspectos de conformidade em relação aos requisitos dos clientes e, também, aspectos internos do software.
- **Organização:** avalia aspectos de conformidade em relação aos requisitos dos clientes e desenvolvedores e, também, aspectos de custo e cronograma.

30

Qualidade de Software

Visões de Qualidade [2/2]



Usuário



Desenvolvedor



Organização

Facilidade de uso, desempenho,
confiabilidade dos resultados,
preços do software, etc.

Taxa de defeitos, facilidade de
manutenção e conformidade em relação
aos requisitos dos usuários, etc.

Cumprimento de prazo, boa previsão de
custo, boa produtividade.

31

Qualidade de Software

ISO/IEC 9126

- A Norma ISO/IEC 9126 define seis **características** para determinar a qualidade de um software:
 - **Funcionalidade:** O software satisfaz as necessidades?
 - **Usabilidade:** O software é fácil de usar?
 - **Confiabilidade:** O software é imune a falhas?
 - **Eficiência:** O software é rápido e “enxuto”?
 - **Manutenibilidade:** O software é fácil de modificar?
 - **Portabilidade:** O software é fácil de usar em outro ambiente?

32

ISO/IEC 9126

Características e Subcaracterísticas de Qualidade [1/6]

- **Funcionalidade:** O software satisfaz as necessidades?
 - **Adequação:** Propõe-se a fazer o que é apropriado?
 - **Acurácia:** Faz o que foi proposto de forma correta?
 - **Interoperabilidade:** É capaz de interagir com os sistemas especificados?
 - **Conformidade:** Está de acordo com as normas, leis etc?
 - **Segurança de Acesso:** Evita acesso não autorizado a programas e dados?

33

ISO/IEC 9126

Características e Subcaracterísticas de Qualidade [2/6]

- **Usabilidade:** O software é fácil de usar?
 - **Compreensibilidade:** É fácil entender o conceito lógico e sua aplicabilidade?
 - **Apreensibilidade:** É fácil aprender a usar?
 - **Operacionalidade:** É fácil operar e controlar?
 - **Encontrabilidade:** É fácil encontrar as informações?

34

ISO/IEC 9126

Características e Subcaracterísticas de Qualidade [3/6]

- **Confiabilidade:** O software é imune a falhas?
 - **Maturidade:** Com que frequência apresenta falhas por defeitos no software?
 - **Tolerância a falhas:** Ocorrendo falhas, como ele reage?
 - **Recuperabilidade:** É capaz de recuperar dados em caso de falhas?

35

ISO/IEC 9126

Características e Subcaracterísticas de Qualidade [4/6]

- **Eficiência:** O software é rápido e “enxuto”?
 - **Comportamento em relação ao tempo:** Qual o tempo de resposta (tempo de processamento) na execução de suas funções?
 - **Comportamento em relação aos recursos:** Quanto recurso usa? Durante quanto tempo?

36

ISO/IEC 9126

Características e Subcaracterísticas de Qualidade [5/6]

- **Manutenibilidade:** O software é fácil de modificar?
 - **Analisabilidade:** É fácil de encontrar uma falha, quando ocorre?
 - **Modificabilidade:** É fácil modificar e adaptar?
 - **Estabilidade:** Existe risco de efeitos inesperados quando se faz alterações?
 - **Testabilidade:** É fácil validar o software modificado?

37

ISO/IEC 9126

Características e Subcaracterísticas de Qualidade [6/6]

- **Portabilidade:** O software é fácil de usar em outro ambiente?
 - **Adaptabilidade:** É fácil adaptar a ambientes diferentes?
 - **Instalabilidade:** É fácil instalar?
 - **Conformidade:** Está de acordo com padrões de portabilidade?
 - **Capacidade de substituição:** É fácil usar para substituir outro?

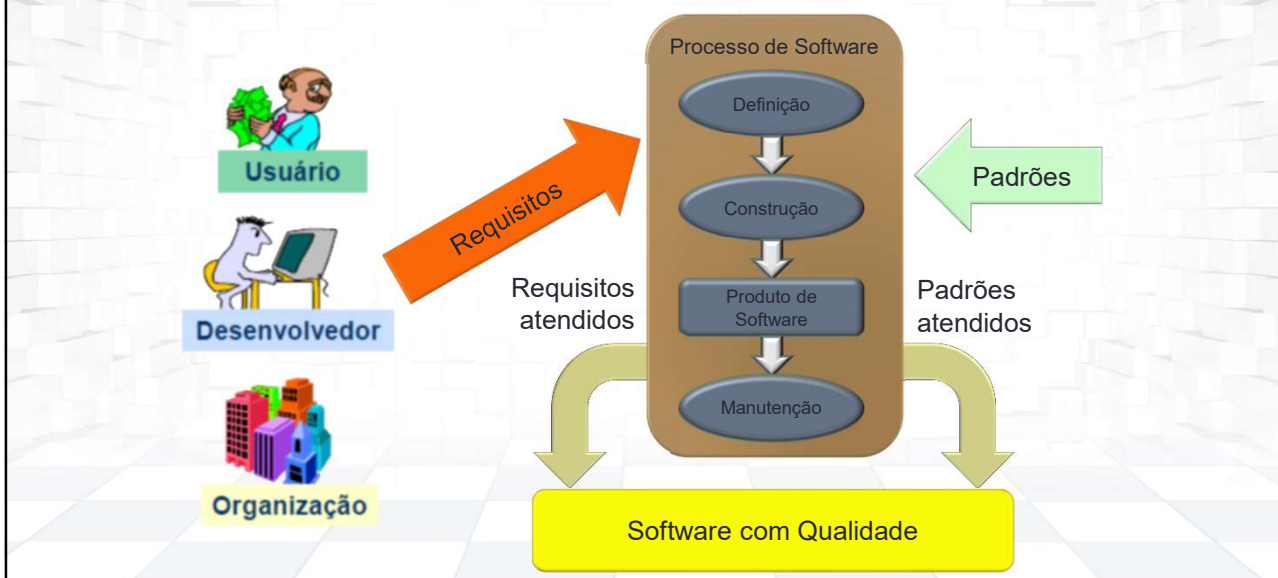
38

Requisitos de Software: Base da Qualidade [1/2]

- Os requisitos de software formam a base de onde a **qualidade** é **avaliada**.
- O software deve **atender** aos requisitos **explícitos** e **implícitos** para ter qualidade.

39

Requisitos de Software: Base da Qualidade [2/2]



40

Qualidade de Software

Definição Genérica

Conjunto de **atributos** de **software** que devem ser satisfeitos de modo que o software atenda às **necessidades** do **usuário** (seja ele um usuário final, um desenvolvedor ou uma organização).

41

Qualidade de Software

Determinação de Atributos Relevantes [1/3]

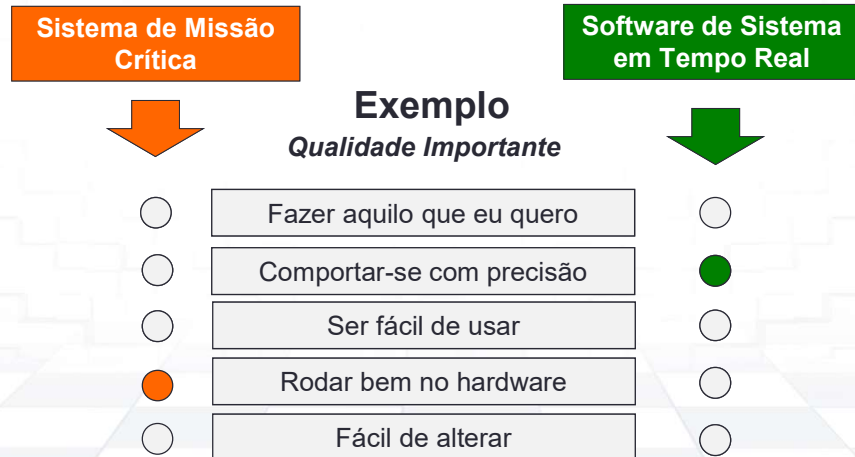
- A determinação dos **atributos** relevantes para cada software varia em função:
 - do **domínio** da aplicação
 - das **tecnologias** utilizadas
 - das características específicas do **projeto**
 - das **necessidades** do usuário e da organização

42

Qualidade de Software

Determinação de Atributos Relevantes [2/3]

- A qualidade depende do tipo de aplicação

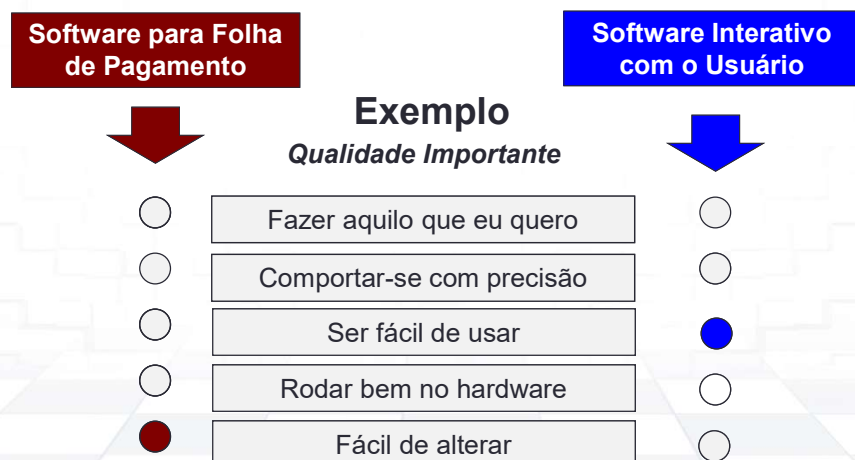


43

Qualidade de Software

Determinação de Atributos Relevantes [3/3]

- A qualidade depende do tipo de aplicação



44

Qualidade de Software

Incorporação da Qualidade

Todas as fases afetam a qualidade do produto final.



- Dos requisitos do usuário à entrega do produto final, existe um **processo de desenvolvimento** que é complexo e, frequentemente, envolve uma série de **estágios** que podem **comprometer** a qualidade do produto final.
- A qualidade **não** pode ser **incorporada** ao produto depois de **pronto**.
- Para obter um produto de software com qualidade, a qualidade deve ser um **objetivo constante** do processo de desenvolvimento do software.

45

Processo de Desenvolvimento

Considerações

- A **qualidade do produto** de software é o objetivo do **processo de desenvolvimento**.
- Um **processo bem estabelecido, compreendido e controlado** pode ajudar a obter qualidade no produto, desde que sejam **definidos claramente** os **requisitos de qualidade** desse produto.



46

Qualidade de Software

Áreas de Estudo da Qualidade

- Sem perder a visão da integração **processo-produto**, os estudos e o desenvolvimento das abordagens de qualidade de software tem sido realizados, dividindo-se qualidade de software em **duas grandes áreas**:

