Engenharia de Software

Qualidade de Software

Prof. Leonardo Vieira Barcelos

Conceito de Qualidade

- O clamor por maior qualidade de software começou quando o software passou a se tornar cada vez mais integrado.
- Na década de 1990, as principais empresas reconheciam que bilhões de dólares estavam sendo disperdiçados em software que não apresentava as características e as funcionalidades prometidas.
- De acordo com Standish Group, (empresa de pesquisa de mercado) que apesar das boas intenções, código mal feito continua a ser o “fantasma” do mercado de software, sendo responsável por 45% do tempo de inatividade dos sistemas computacionais.

Conceito de Qualidade

- Em 2005, a *ComputerWorld* lamentou que “software de má qualidade está em praticamente em todas as organizações que usam computadores, provocando horas de trabalho perdidas durante o tempo que a máquina fica parada, dados perdidos ou corrompidos, oportunidades de vendas perdidas, custo de suporte e manutenção de TI elevados e baixa satisfação do cliente.

Conceito de Qualidade

- Hoje em dia, a qualidade de software continua a ser um problema, mas a quem culpar?
  - Os clientes culpam os desenvolvedores, argumentando que práticas descuidadas levam a um software de baixa qualidade.
  - Os desenvolvedores de software culpam os clientes, argumentando que datas de entrega absurdas e um fluxo contínuo de mudanças os forçam a entregar software antes de eles estarem completamente validados.
- Quem está com a razão?
  - Ambos – e esse é o problema.

O que é qualidade de Software?

- Qualidade é um termo que pode ter diferentes interpretações.
- Existem muitas definições de qualidade de software propostas na literatura, sob diferentes pontos de vistas.

Qualidade de Software

Definição [David Garvin, 1984]***:***

Sugere que "qualidade é um conceito complexo e multifacetado" que pode ser descrito segundo cinco pontos de vista diferentes.

A visão transcedental*: é algo que reconhece imediatamente, mas não se consegue definir explicitamente.*

A visão do usuário*: vê a qualidade em termos das metas específicas, por exemplo se um produto atende a essas metas, ele apresenta qualidade.*

A visão do fabricante*: define em termos da especificação original do produto. Se o produto atende às especificações, ele apresenta qualidade.*

A visão do produto*: sugere que a qualidade pode ser ligada a características inerentes (por exemplo, funções e recursos) de um produto.*

A visão baseada em valor*: mede a qualidade tormando como base o quanto um cliente estaria disposto a pagar por um produto.*

Na realidade, qualidade engloba todas as visões e outras mais.

Qualidade de Software

Definição [David Garvin, 1984]***:***

Qualidade de Projeto*: referese-se às características que os projetistas especificam para um produto. No desenvolvimento de software, a qualidade de projeto engloba o grau de atendimento às funções e características especificadas no modelo de requisitos.*

Qualidade de Conformidade*: focaliza o grau em que a implementação segue o projeto e o sistema resultante atende suas necessidades e as metas de desempenho.*

Qualidade de Software

Definição:

“Argumenta que a qualidade é importante, mas se o usuário não estiver satisfeito, nada mais importa”

Satisfação do usuário = produto compatível + boa qualidade + entrega dentro do orçamento e do prazo previsto.

[DeMarco, 1998].

Qualidade de Software

Definição:

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

Qualidade de Software

Definição:

"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].

Qualidade de Software

Definição:

Qualidade é a totalidade de características e critérios de um produto ou serviço que exercem suas habilidades para satisfazer às necessidades declaradas ou envolvidas "[ISO9126 1994].

Como se define qualidade?

- Num sentido amplo, qualidade de software é definida como: *uma **gestão de qualidade efetiva** aplicada de modo a criar um **produto útil** que **forneça valor mensurável** para aqueles que o produzem e para aqueles que o utilizam.*
  - ***Gestão de qualidade efetiva**: estabelece a infraestrutura que dá suporte a qualquer tentativa de construir um produto de software de alta qualidade. Os aspectos administrativos do processo criam mecanismos de controle e equilibrio de poderes que ajudam a evitar o caos no projeto.*
  - ***Produto útil**: fornece o conteúdo, as funções e os recursos que o usuário final deseja, além disso, deve fornecer confiabilidade e isenção de erros.*
  - ***Agregar valor tanto para o fabricante quanto para o usuário**: um software de alta qualidade gera benefícios para a empresa de software bem como para a comunidade de usuários finais.*

Qualidade: Objetivo do Processo de Desenvolvimento

- A qualidade do produto de software é um objetivo do processo de desenvolvimento.
- Assim, ao desenvolver um produto, deve-se ter previamente estabelecidas, como perspectiva, as características de qualidade que se desejam alcançar.

Visões de Qualidade de Software

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.

Dimensões de qualidade Garvin

- David Garvin (1987) sugere oito dimensões de qualidade, que embora não tenham sido desenvolvidas especificamente para software, elas podem ser aplicadas quando se considera qualidade de software.
  - **Qualidade de desempenho**: O software fornece todo o conteúdo, funções e recursos que são especificados como parte do modelo de requisitos de forma a gerar valor ao usuário final?
  - **Qualidade de recursos**: O software fornece recursos que surpreendem e encantam usuários finais que os utilizam pela primeira vez?
  - **Confiabilidade**: O software fornece todo os recursos e capacidades sem falha? Fornece funcionalidade sem a ocorrência de erros?
  - **Conformidade**: O software está de acordo com os padrõs de software locais e externos relacionados com a aplicação? Segue as convenções de projeto e codificação de fato?

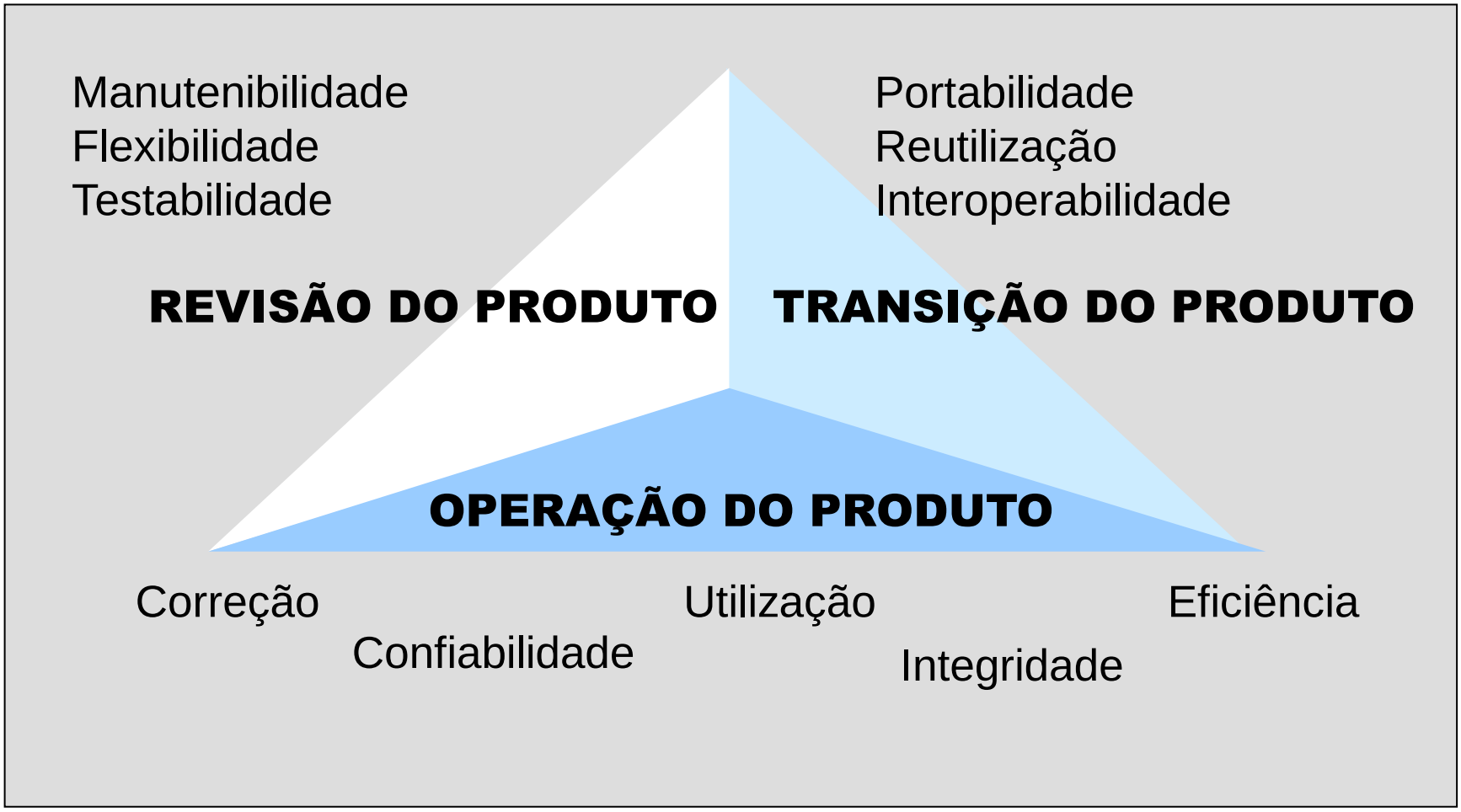
Dimensões de qualidade Garvin

- Continuação:
  - **Durabilidade**: O software pode ser mantido (modificado) ou corrigido (depurado) sem a geração involuntária de efeitos colaterias indesejados?
  - **Facilidade de manutenção**: O software pode ser mantido (modificado) ou corrigido (depurado) em um período de tempo aceitável e curto? O pessoal de suporte pode obter todas as informações necessárias para realizar alterações ou corrigir defeitos?
  - **Estética**: Não há dúvida nenhuma de que cada um de nós tem uma visão diferente e muito subjetivos do que é estética. Mesmo assim, a maioria de nós concordaria que uma entidade estética tem certa elegância que são difíceis de quantificar mas que, não obstante, são evidentes.
  - **Percepção**: Em algumas situações, temos alguns preconceitos que influenciarão nossa percepção de qualidade. Por exemplo, se for apresentado um produto de software, construído por um fornecedor que, no passado, havia produzido um software de má qualidade.

Fatores de qualidade McCall

- McCall, Richards e Walters [MCC77] propõem uma categorização útil de fatores que afetam a qualidade do software.
- Esses fatores concentram-se nos três aspectos importantes para o software:
  - suas características operacionais,
  - sua habilidade de passar por modificações e
  - sua adaptabilidade em novos ambientes.

Fatores de qualidade de software de McCall



Fatores de qualidade de software de McCall

- **Correção**: Quando um programa satisfaz sua especificação e preenche os objetivos da missão do cliente.
- **Confiabilidade**: Quando se pode esperar que um programa realize a função pretendida com a precisão exigida.
- **Eficiência**: Quantidade de recursos de computação e código necessários para um programa realizar sua função.
- **Integridade**: Quanto do acesso ao software ou dados por pessoas não-autorizadas pode ser controlado.
- **Usabilidade**: O esforço necessário para aprender, operar, preparar entradas e interpretar saídas de um programa.
- **Manutenibilidade**: O esforço necessário para localizar e consertar um erro em um programa.

Fatores de qualidade de software de McCall

- **Flexibilidade**: O esforço necessário para modificar um programa operacional.
- **Testabilidade**: Esforço necessário para testar um programa, a fim de garantir que ele realize a função esperada.
- **Portabilidade**: Esforço necessário para transferir o programa de um ambiente de hardware ou software para outro.
- **Reutilização**: Quando um programa (ou partes dele) pode ser reusado em outras aplicações – relativo ao empacotamento e escopo das funções que o programa realiza.
- **Interoperabilidade**: Esforço necessário para acoplar um sistema a outro.

Fatores de qualidade ISO 9126

- A norma ISO 9126 foi desenvolvida em uma tentativa de identificar atributos de qualidade para software de computador.
- A norma identifica seis atributos-chave de qualidade:
  - **Funcionalidade**: O grau que o software satisfaz às necessidades declaradas conforme indicado pelos seguintes subatributos: adequabilidade, exatidão, interoperabilidade, conformidade e segurança.
  - **Confiabilidade**: A quantidade de tempo que o software fica disponível para uso conforme indicado pelos seguintes subatributos: maturidade, tolerância a falhas, facilidade de recuperação.
  - **Usabilidade**: O grau de facilidade do uso do software conforme indicado pelos seguintes subatributos: facilidade de compreensão, facilidade de aprendizagem, operabilidade.

Fatores de qualidade ISO 9126

- Continuação.
  - **Eficiência:** O grau de otimização do uso, pelo software, dos recursos do sistema conforme indicado pelos seguintes subatributos: comportamento em relação ao tempo, comportamento em relação aos recursos.
  - **Facilidade de manutenção**: a facilidade com a qual uma correção pode ser realizada no software conforme indicado pelos seguintes subatributos: facilidade de análise, facilidade de realização de mudanças, estabilidade e testabilidade.
  - **Portabilidade**: A facilidade com a qual um software pode ser transposto de um ambiente a outro conforme indicado pelos seguintes subatributos: adaptabilidade, facilidade de instalação, conformidade, facilidade de substituição.

Fatores de qualidade desejados

- Até o momento foram apresentados indicações genérica da qualidade de uma aplicação.
- Agora, se fosse solicitado para revisar a interface do usuário e avaliar sua usabilidade, como poderia proceder?
  - Para conduzir esta avaliação, é preciso lidar com atributos específicos, mensuráveis (ou pelo menos, reconhecíveis) da interface.

Fatores de qualidade desejados

- Exemplo:
  - **Intuição**: o grau em que a interface segue padrões de uso esperados de modo que até mesmo um novato possa usá-la sem treinamento significativo.
    - O layout da interface favorece a fácil compreensão?
    - As operações da interface são fáceis de ser localizadas e iniciadas?
    - É especificada entrada para economizar toques de teclado ou cliques de mouse?
    - A estética ajuda no entendimento e uso?
  - **Eficiência**: A facilidade com a qual as operações e informações podem ser localizadas ou iniciadas.
    - O layout e o estilo da interface permitem a um usuário localizar eficientemente as operações e informações?
    - Uma sequência de operações (ou entrade de dados) pode ser realizada reduzindo-se o número de movimentos?
    - Os dados de saída ou o conteúdo são apresentados de modo a ser imediatamente compreendidos?

Fatores de qualidade desejados

- Exemplo:
  - **Robustez**: o grau com o qual o software trata dados incorretos de entrada ou interação inapropriada com o usuário.
    - O software reconhecerá erros caso sejam introduzidos dados dentro ou for a dos limites prescritos? Mais importante ainda, o software continuará a operar sem falha?
    - A interface reconhece erros cognitivos ou manipuladores comuns e orienta explicitamente o usuário para retomar o caminho certo?
  - **Riqueza**: O grau em que a interface oferece um conjunto rico de recuros importantes.
    - A interface pode ser personalizada de acordo com as necessidades específicas de um usuário?
    - A interface dispõe de recursos de macros que permitem ao usuário identificar uma sequência de operações comuns por meio de uma única ação ou comando?
- Se a resposta a essas perguntas for “sim”, é provável que a interface do usuário apresenta alta qualidade. Um conjunto de perguntas similares deveria ser desenvolvida para cada fator de qualidade a ser avaliado.

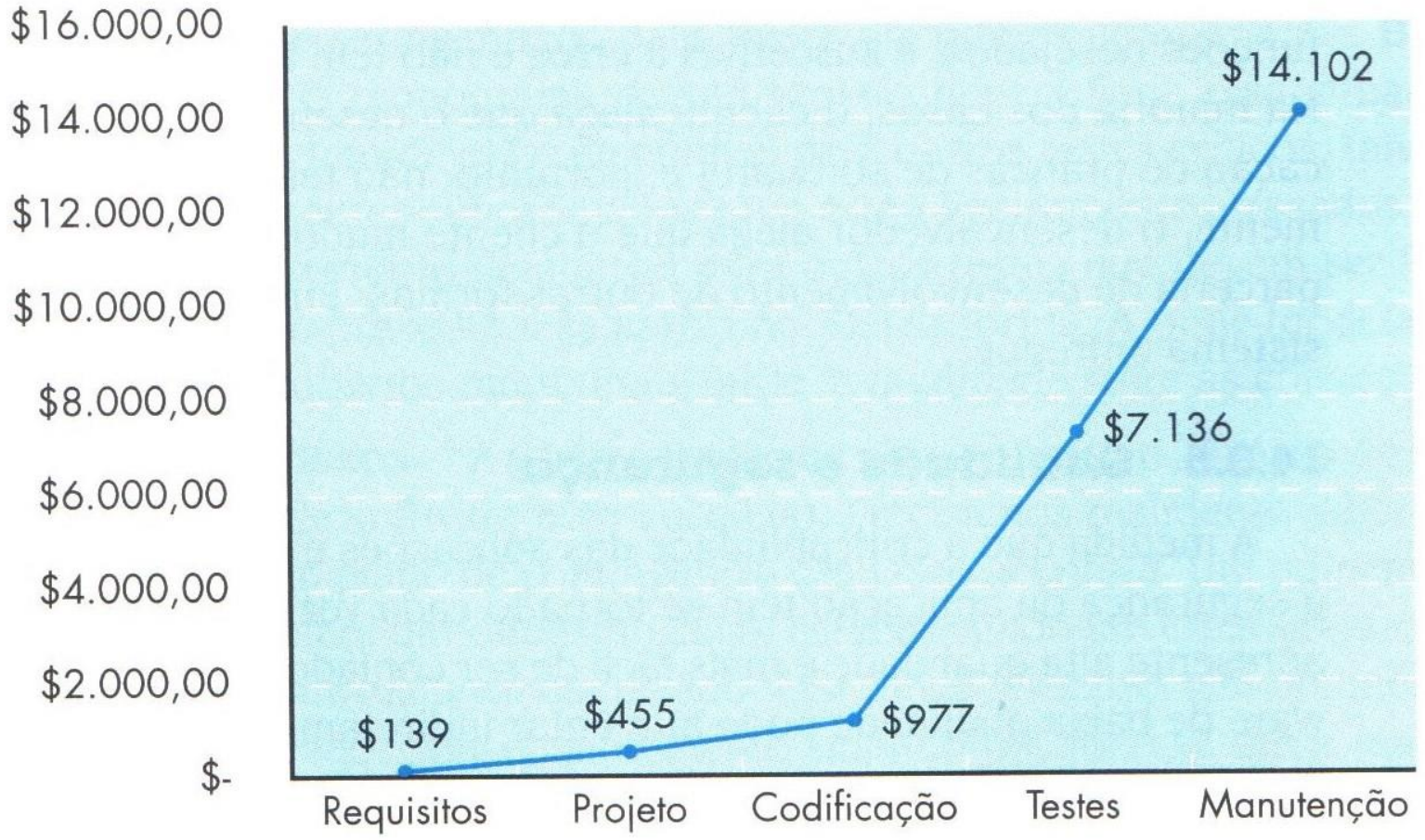
O Dilema da Qualidade de Software

- Riscos
  - "O foguete Ariane 5 explodiu na subida devido tão somente a um defeito de software (um erro) envolvendo a conversão de uma valor de ponto flutuante com 64 bits em um inteiro de 16 bits. O foguete e seus quatro satélites estavam sem seguro e valiam 500 milhões de dólares. Um teste de sistemas abrangente teria encontrado o erro, mas foi vetado por razões orçamentarias."

O Dilema da Qualidade de Software

- Custo da Qualidade
  - A qualidade é importante, mas ela custa tempo e dinheiro.
  - Não há dúvida nenhuma de que a qualidade tem um preço, mas a falta de qualidade também tem um preço – não apenas para os usuários finais, que terão de conviver com um software com erros, mas também para a organização que o criou que terá de fazer a manutenção.
  - O custo da qualidade pode ser dividido em custos associados à: ***prevenção***, ***avaliação*** e ***falha***.

O Dilema da Qualidade de Software



Custo relativo para correção de erros e defeitos

(Adaptado por Boehm e Basili)

Transição para visão Quantitativa

- Os fatores de qualidade McCall e ISO 9126 só podem ser julgados diretamente de forma subjetiva.
  - O julgamento precisa ser feito por especialistas que analisam o software qualitativamente.
- Foram desenvolvidas métricas de software que fornecem uma avaliação quantitativa da qualidade de software.
  - Essas medidas não medem a qualidade diretamente, mas alguma manifestação da qualidade.
  - O fator complicador é a relação precisa entre a variável medida e a qualidade do software.
- Estudaremos sobre métricas de software mais adiante.

Medição

- Um elemento-chave de qualquer processo de engenharia é a medição.
  - A medição é o processo pelo qual números são associados a propriedades de objetos do mundo real.
  - Serve para entender melhor as características dos objetos.
  - Serve para avaliar e monitorar a qualidade dos produtos.
- Ao contrário de outras engenharias, a engenharia de software não está ancorada nas leis quantitativas da física.
  - Medidas como massa, voltagem, temperatura, pressão não são apropriadas para software.
  - Isso quer dizer que software é não-mensurável?
    - Não: apenas outros critérios devem ser estabelecidos.

Vocabulário Básico

- **Medida:** fornece uma indicação quantitativa da extensão, quantidade, dimensão, capacidade ou tamanho de algum atributo de um produto ou processo.
- **Medição:** é o ato de determinar uma medida.
- **Métrica:** O IEEE *Standard Glossary* define como "uma medida quantitativa do grau em que um sistema, componente ou processo possui um determinado atributo".
  - Exemplo: Número médio de erros encontrados por revisão ou número médio de erros encontrados por teste de unidade.

Vocabulário Básico

- **Indicador:** é uma métrica ou conjunto de métricas que proporcionam informações sobre o processo de software, em um projeto de software ou no próprio produto.
- Um engenheiro de software coleta medidas e desenvolve métricas para obter indicadores.
- Um indicador proporciona informações que permitem o gerente de projeto ou aos engenheiros de software ajustar o processo, o projeto ou o produto para incluir melhorias.