



Conceitos de Qualidade de Software

Herysson R. Figueiredo
herysson.figueiredo@ufn.edu.br



Sumário

- O que é qualidade?
- Qualidade de Software
 - Dimensões de qualidade de Garvin
 - Fatores de qualidade de McCall
 - Fatores de qualidade ISO 9126
 - Fatores de qualidade desejados
- O Dilema da qualidade de software
- Alcançando a Qualidade de Software



O que é Qualidade

David Garvin [Gar84], sugere que “**qualidade é um conceito complexo e multifacetado**” que pode ser descrito segundo cinco pontos de vista diferentes.



O que é Qualidade

A visão transcendental sustenta (assim como Persig) que qualidade é algo que se reconhece imediatamente, mas não se consegue definir explicitamente

A visão do usuário vê a qualidade em termos das metas específicas de um usuário final. Se um produto atende a essas metas, ele apresenta qualidade.



O que é Qualidade

A visão do fabricante define qualidade 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 tomando como base o quanto um cliente estaria disposto a pagar por um produto.



O que é Qualidade

Qualidade de projeto refere-se às características que os projetistas especificam para um produto. A qualidade dos materiais, as tolerâncias e as especificações de desempenho, todos são fatores que contribuem para a qualidade de um projeto.



O que é Qualidade

No desenvolvimento de software, a qualidade de um projeto engloba o grau de atendimento às funções e características especificadas no modelo de requisitos. A 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

Segundo Roger S. Pressman [PRESSMAN, 2011], qualidade de software é: “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.”



Qualidade de Software

- **Uma gestão de qualidade** efetiva estabelece a infraestrutura que dá suporte a qualquer tentativa de construir um produto de software de alta qualidade.
- **Um produto útil** fornece o conteúdo, as funções e os recursos que o usuário final deseja, além disso, e não menos importante, deve fornecer confiabilidade e isenção de erros



Qualidade de Software

- **Ao agregar valor** tanto para o fabricante quanto para o usuário de um produto de software, um software de alta qualidade gera benefícios para a empresa de software bem como para a comunidade de usuários finais.



Dimensões de qualidade de Garvin

David Garvin [Gar87] sugere que a qualidade deve ser considerada adotando-se um ponto de vista multidimensional que começa com uma avaliação da conformidade e termina com uma visão transcendental (estética).



Dimensões de qualidade de Garvin

- Qualidade do desempenho.
- Qualidade dos recursos.
- Confiabilidade.
- Conformidade.



Dimensões de qualidade de Garvin

- Qualidade do 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 dos recursos.
- Confiabilidade.
- Conformidade.



Dimensões de qualidade de Garvin

- Qualidade do desempenho.
- Qualidade dos recursos.
 - O software fornece recursos que surpreendem e encantam usuários finais que os utilizam pela primeira vez?
- Confiabilidade.
- Conformidade.



Dimensões de qualidade de Garvin

- Qualidade do desempenho.
- Qualidade dos recursos.
- Confiabilidade.
 - O software fornece todos os recursos e capacidades sem falhas? Está disponível quando necessário? Fornece funcionalidade sem a ocorrência de erros?
- Conformidade.



Dimensões de qualidade de Garvin

- Qualidade do desempenho.
- Qualidade dos recursos.
- Confiabilidade.
- Conformidade.
 - O software está de acordo com os padrões de software locais e externos relacionados com a aplicação? Segue as convenções de projeto e codificação de fato?



Dimensões de qualidade de Garvin

- Durabilidade.
- Facilidade de manutenção.
- Estética.
- Percepção.



Dimensões de qualidade de Garvin

- Durabilidade.
 - O software pode ser mantido (modificado) ou corrigido (depurado) sem a geração involuntária de efeitos colaterais indesejados? As mudanças farão com que a taxa de erros ou a confiabilidade diminuam com o passar do tempo?
- Facilidade de manutenção.
- Estética.
- Percepção.



Dimensões de qualidade de Garvin

- Durabilidade.
- 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.
- Percepção.



Dimensões de qualidade de Garvin

- Durabilidade.
- Facilidade de manutenção.
- Estética.
 - Não há dúvida nenhuma de que cada um de nós tem uma visão diferente e muito subjetiva do que é estética.
- Percepção.



Dimensões de qualidade de Garvin

- Durabilidade.
- Facilidade de manutenção.
- Estética.
- Percepção.
 - Em algumas situações, temos alguns preconceitos que influenciarão nossa percepção de qualidade.



Fatores de qualidade de McCall

McCall, Richards e Walters [McC77] criaram uma proposta de categorização dos fatores que afetam a qualidade de software. Esses fatores de qualidade de software, focam em três importantes aspectos de um produto de software: as **características operacionais**, a **habilidade de suportar mudanças** e a **adaptabilidade a novos ambientes**.

Fatores de qualidade de McCall





Fatores de qualidade ISO 9126

O padrão ISO 9126 foi desenvolvido como uma tentativa de identificar os atributos fundamentais de qualidade para software de computador.



Fatores de qualidade ISO 9126

Funcionalidade. O grau com 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.



Fatores de qualidade ISO 9126

Usabilidade. O grau de facilidade de utilização do software conforme indicado pelos seguintes subatributos: facilidade de compreensão, facilidade de aprendizagem, operabilidade.

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.



Fatores de qualidade ISO 9126

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, 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

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?
- A interface utiliza uma metáfora reconhecível?
- É especificada entrada para economizar toques de teclado ou cliques de mouse?
- A estética ajuda no entendimento e uso?



Fatores de qualidade desejados

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 entrada 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

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 fora dos limites prescritos? Mais importante ainda, o software continuará a operar sem falha ou degradação?
- A interface reconhece erros cognitivos ou manipuladores comuns e orienta explicitamente o usuário para retomar o caminho certo?
- A interface oferece diagnósticos e orientação úteis quando é descoberta uma condição de erro (associada à funcionalidade do software)?



A transição para uma visão quantitativa

Tudo visto até aqui são diversos fatores qualitativos para a “medição” da qualidade de um software. A comunidade da engenharia de software se esforça ao máximo para desenvolver medidas precisas para a qualidade de software e algumas vezes é frustrada pela natureza subjetiva da atividade.



A transição para uma visão quantitativa

Nas próximas aulas será apresentado um conjunto de métricas de software que podem ser aplicadas para a avaliação quantitativa da qualidade de software.



A transição para uma visão quantitativa

Em todos os casos, as métricas representam medidas indiretas; isto é, **jamais medimos realmente a qualidade**, mas sim alguma manifestação dessa qualidade.



O Dilema da qualidade de *software*

“Se produzimos um sistema de software de péssima qualidade, perdemos porque ninguém irá querer comprá-lo. Se, por outro lado, gastamos um tempo infinito, um esforço extremamente grande e grandes somas de dinheiro para construir um software absolutamente perfeito, então isso levará muito tempo para ser completado, e o custo de produção será tão alto que iremos à falência. Ou perdemos a oportunidade de mercado ou então simplesmente esgotamos todos os nossos recursos.” [Bertrand Meyer]



O Dilema da qualidade de *software*

“Dessa maneira, os profissionais desta área tentam encontrar aquele meio-termo mágico onde o produto é suficientemente bom para não ser rejeitado logo de cara, como, por exemplo, durante uma avaliação, mas também não é o objeto de tamanho perfeccionismo e trabalho que levaria muito tempo ou que custaria demasiadamente para ser finalizado.”
[Bertrand Meyer]



Software “bom o suficiente”

Software bom o suficiente fornece funções e características de alta qualidade que os usuários desejam, mas, ao mesmo tempo, fornece outras funções e características mais obscuras ou especializadas e ainda contendo erros conhecidos.



Software “bom o suficiente”

O fornecedor de software espera que a grande maioria dos usuários ignore os erros pelo fato de estarem muito satisfeitos com as outras funcionalidades oferecidas pela aplicação.



Custo da qualidade

Sabemos que a qualidade é importante, mas ela nos custa tempo e dinheiro tempo e dinheiro em demasia para obter o nível de qualidade de software que realmente desejamos.



Custo da qualidade

O *custo da qualidade* inclui todos os custos necessários para a busca de qualidade ou para a execução de atividades relacionadas à qualidade, assim como os custos causados pela falta de qualidade. O custo da qualidade pode ser dividido em custos associados à **prevenção, avaliação e falhas**.



Custo da qualidade

Os custos de prevenção incluem:

1. o custo de atividades de gerenciamento necessárias para planejar e coordenar todas as atividades de controle e garantia da qualidade,
2. o custo de atividades técnicas adicionais para desenvolver modelos completos de requisitos e de projeto,
3. custos de planejamento de testes e
4. o custo de todo o treinamento associado a essas atividades.



Custo da qualidade

Os **custos de avaliação** incluem atividades para a compreensão aprofundada da condição do produto “a primeira vez através de” cada processo. Entre os exemplos de custos de avaliação, temos:

- Custo para realização de **revisões técnicas** para produtos resultantes de engenharia de software.
- Custo para **coleta de dados e avaliação de métricas**.
- Custo para **testes e depuração**.



Custo da qualidade

Os **custos de falhas** são aqueles que desapareceriam caso nenhum erro tivesse surgido antes ou depois da entrega de um produto a clientes. Esses custos podem ser subdivididos em **custos de falhas internas** e **custos de falhas externas**.



Custo da qualidade

Os **custos de falhas internas** ocorrem quando se detecta um erro em um produto antes de ele ser entregue e abrangem:

- Custo necessário para realizar retrabalhos (reparos) para corrigir um erro.
- Custo que ocorre quando retrabalhos geram, inadvertidamente, efeitos colaterais que devem ser reduzidos.
- Custos associados à reunião de métricas de qualidade que permitem a uma organização avaliar os modos de falha.



Custo da qualidade

Os **custos de falhas externas** estão associados a defeitos encontrados após o produto ter sido entregue ao cliente.

Exemplos de custos de falhas externas são resolução de reclamações, devolução e substituição de produtos, suporte telefônico/via e-mail e custos de mão de obra associados à garantia do produto.

Custo da qualidade

Custo relativo para correção de erros e defeitos





Riscos

“as pessoas apostam seus empregos, comodidades, segurança, entretenimento, decisões e as próprias vidas em software. Esperamos que elas estejam certas”. [PRESSMAN, 2011]

A implicação disso é que software de baixa qualidade aumenta os riscos tanto para o desenvolvedor quanto para o usuário final.



Negligência e responsabilidade civil

Na maioria dos casos, o cliente alega que o desenvolvedor foi negligente (na maneira de aplicação de práticas de software) e, portanto, não tem direito a receber seu pagamento. Normalmente, o desenvolvedor alega que o cliente mudou repetidamente os requisitos e subverteu a parceria de desenvolvimento de outras formas.



Qualidade e segurança

A segurança de software se relaciona inteira e completamente à qualidade. Devemos nos preocupar com a segurança, a confiabilidade, a disponibilidade e a dependência nas fases inicial, de projeto, de arquitetura, de testes e de codificação, ao longo de todo o ciclo de vida [qualidade] de um software.



O impacto das ações administrativas

A qualidade de software é normalmente tão influenciada pelas decisões administrativas quanto pelas decisões técnicas.



O impacto das ações administrativas

Decisões de estimativas. Raramente uma equipe de software pode se dar ao luxo de fornecer uma estimativa para um projeto antes das datas de entrega serem estabelecidas e um orçamento geral ser especificado.

Decisões de cronograma. Quando um cronograma de projeto de software é estabelecido, as tarefas são sequenciadas tomando-se como base as dependências.



O impacto das ações administrativas

Decisões orientadas a riscos. A administração de riscos é um dos atributos fundamentais de um projeto de software bem-sucedido. Precisamos realmente saber o que poderia dar errado e estabelecer um plano de contingência caso isso aconteça.



Alcançando a Qualidade de Software

A qualidade de software não aparece simplesmente do nada. Ela é o resultado de um bom gerenciamento de projeto e uma prática consistente de engenharia de software.



Métodos de engenharia de software

Para construir software de alta qualidade, temos de entender o problema a ser resolvido. Temos também de ser capazes de criar um projeto que seja adequado ao problema e, ao mesmo tempo, apresente características que levem a um software com as dimensões e fatores de qualidade.



Técnicas de gerenciamento de software

O impacto de decisões de gerenciamento inadequadas sobre a qualidade de software. As implicações são claras:

1. um gerente de projeto usar estimativas para verificar que as datas de entrega são plausíveis,
2. as dependências de cronograma forem entendidas e a equipe resistir à tentação de usar atalhos,
3. o planejamento de riscos for conduzido de modo que problemas não gerem caos, a qualidade do software será afetada de forma positiva.



Controle de qualidade

O controle de qualidade engloba um conjunto de ações de engenharia de software que ajudam a garantir que cada produto resultante atinja suas metas de qualidade. Os modelos são revistos de modo a garantir que sejam completos e consistentes.



Garantia da qualidade

A garantia da qualidade estabelece a infraestrutura que suporta métodos sólidos de engenharia de software, gerenciamento racional de projeto e ações de controle de qualidade todos fundamentais para a construção de software de alta qualidade.



Exercícios

Desenvolva os exercícios da lista de Conceitos de Qualidade de Software



Bibliografia

PRESSMAN, Roger S. Engenharia de software. 7. ed. Rio de Janeiro: McGraw Hill, 2011.

KOSCIANSKI, André; SOARES, Michel dos Santos. Qualidade de software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. São Paulo: Novatec, 2ª ed., 2007.

SOMMERVILLE, Ian. Engenharia de Software. 8. ed. São Paulo: Addison Wesley, 2007