

## QUALIDADE DE SOFTWARE

Com o avanço da tecnologia, a qualidade de um software se tornou um item indispensável, tornando-a obrigatória para a competitividade do mercado atual. O seu impacto é maior no produto final, visando a satisfação do cliente com o software requerido.

### Dimensões de qualidade de Garvin

David A. Garvin, nascido em 1952 em Nova York, e falecido em 2017, foi um acadêmico e professor da Harvard Business School. Ele se destacou por sua dedicação ao estudo da gestão, qualidade e excelência.

Garvin teve contribuições significativas para a área da gestão e da qualidade. O Modelo das Oito Dimensões da Qualidade é uma de suas contribuições mais notáveis. Ele especificou as características que definem a qualidade de um produto ou serviço: **desempenho, características, confiabilidade, conformidade, durabilidade, serviço, estética e percepção.**

- **Desempenho:** Refere-se à capacidade do produto ou serviço de executar suas funções de maneira eficaz e eficiente. Envolve características como velocidade, precisão, eficácia e confiabilidade.
- **Características:** São os recursos adicionais ou atributos que agregam valor ao produto ou serviço. Essas características podem ser únicas e se destacar em relação à concorrência, como um recurso especial em um produto eletrônico.
- **Confiabilidade:** Diz respeito à consistência do desempenho ao longo do tempo. Produtos ou serviços confiáveis são aqueles que não apresentam falhas ou defeitos frequentes. Além disso, são capazes de manter um bom funcionamento ao longo de sua vida útil.
- **Conformidade:** Refere-se à aderência do produto ou serviço a padrões, regulamentos e especificações predefinidos. Produtos ou serviços que estão em conformidade com esses requisitos têm maior probabilidade de serem confiáveis e seguros.

- Durabilidade: Relaciona-se à vida útil do produto ou serviço e à sua capacidade de resistir a desgaste e deterioração ao longo do tempo. Produtos duráveis têm uma longa vida útil e mantêm sua qualidade por mais tempo.
- Serviço: Envolve o suporte ao cliente e a assistência fornecida após a venda do produto ou prestação do serviço. Isso inclui atendimento ao cliente, garantias, manutenção e outros serviços pós-venda.
- Estética: Diz respeito à aparência visual do produto ou serviço. A estética pode influenciar a percepção do cliente sobre a qualidade e o valor do produto, afetando a decisão de compra.
- Qualidade percebida: Refere-se à percepção subjetiva do cliente sobre a qualidade do produto ou serviço. Isso pode ser influenciado por experiências anteriores, marca, reputação e outras considerações emocionais.

David A. Garvin deixou um legado inegável no campo da qualidade, excelência e gestão. Suas contribuições moldaram a maneira como as organizações buscam a excelência, promovendo a melhoria contínua e a aprendizagem organizacional.

## **McCall**

James McCall criou um dos primeiros modelos de qualidade de software, em 1977, onde se baseava em 3 pontos de vista, sendo cada um deles com diversos critérios, são eles a Operação, na qual são avaliadas as características relacionadas ao uso do produto; a Revisão, na qual é avaliada a capacidade do software de receber futuras evoluções e/ou modificações; e a Transição, na qual é avaliado a capacidade de se adaptar a diversos ambientes.

- operação
  - correção
  - confiabilidade
  - eficiência
  - integridade

- usabilidade
- revisão
  - manutenibilidade
  - flexibilidade
  - testabilidade
- transição
  - portabilidade
  - reusabilidade
  - interoperabilidade

## **ISO 9126**

ISO/IEC 9126 é uma norma ISO para qualidade de produto de software. Ela define um conjunto de parâmetros com o objetivo de padronizar a avaliação da qualidade de software. Ela se enquadra no modelo de qualidade das normas da família 9000.

A ISO/IEC 9126, sob o título geral "Engenharia de software - Qualidade do produto", consiste nas seguintes partes:

- Parte 1: Modelo de qualidade;
- Parte 2: Métricas externas;
- Parte 3: Métricas internas;
- Parte 4: Métricas de qualidade em uso

A norma brasileira correspondente é a NBR 13596, que foi substituída pela NBR ISO/IEC 9126-1 que por sua vez foi substituída pela Norma ISO/IEC 25010:2011.

Esta parte da NBR ISO/IEC 9126 descreve um modelo de qualidade do produto de software, composto de duas partes: a) qualidade interna e qualidade externa e b) qualidade em uso. A primeira parte do modelo especifica seis características para qualidade interna e externa, as quais são por sua vez subdivididas em subcaracterísticas. Estas subcaracterísticas são manifestadas

externamente, quando o software é utilizado como parte de um sistema computacional, e são resultantes de atributos internos do software. Esta parte da NBR ISO/IEC 9126 não apresenta o modelo de qualidade interna e externa além do nível de subcaracterísticas.

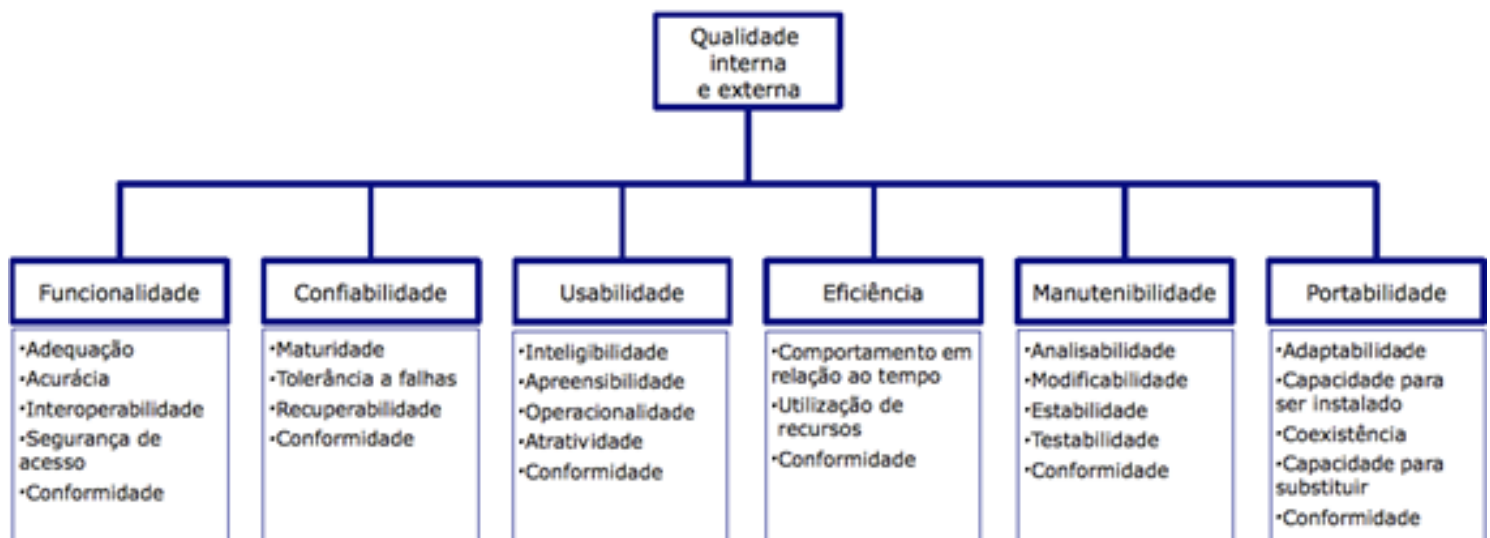
A segunda parte do modelo especifica quatro características de qualidade em uso, mas não apresenta o modelo de qualidade em uso além do nível de característica. Qualidade em uso é, para o usuário, o efeito combinado das seis características de qualidade do produto de software.

As características definidas são aplicáveis a todo tipo de software, incluindo programas de computador e dados contidos em firmware. As características e subcaracterísticas fornecem terminologia consistente para tratar de qualidade do produto de software. Elas também fornecem uma estrutura para especificar requisitos de qualidade de software e realizar comparações entre produtos de software.

A norma ISO/IEC 9126, ou conjunto de normas que tratam deste assunto no âmbito da ISO, estabelece um modelo de qualidade com os seguintes componentes: processo, produto e qualidade em uso.

#### Modelo de Qualidade da Norma ISO 9126

A norma 9126, foca na qualidade do produto de software, propondo Atributos de Qualidade, distribuídos em seis características principais: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.



Atualmente, esta norma foi substituída pela ISO 25010, que foi introduzida em 2011

## **Fatores de qualidade desejados**

Durante o desenvolvimento de um software, a participação do cliente é essencial, para que melhor sejam atendidos os seus interesses. Para isso, deve responder algumas questões

- Funcionalidade
  - satisfazem as necessidades do cliente?
- confiabilidade
  - é seguro e estável para que o desempenho permaneça o mesmo ao longo do tempo?
- manutenibilidade
  - suas correções e/ou manutenções são fáceis?
- usabilidade
  - É fácil de utilizar?
- eficiência
  - O desempenho é semelhante aos recursos e tempos usados para seu desenvolvimento?
- portabilidade
  - Pode usar o software em qualquer plataforma?

## **Transição para uma vida quantitativa**

A "Transição para uma Visão Quantitativa" refere-se ao processo de mudança na forma de analisar e interpretar dados, focando em abordagens quantitativas em vez de qualitativas. Essa transição é comum em áreas como ciência, economia, engenharia, entre outras, onde a coleta de dados numéricos, estatísticas e modelos matemáticos se torna fundamental para embasar decisões, previsões e explicações.

Em termos mais simples, significa passar de uma abordagem mais subjetiva ou qualitativa (que se baseia em observações e análises mais descritivas) para uma abordagem objetiva e baseada em números, gráficos e cálculos. Isso pode incluir:

- **Uso de Modelos Matemáticos e Estatísticos:** Para prever comportamentos, medir resultados ou comparar diferentes cenários de forma precisa.
- **Coleta e Análise de Dados Quantitativos:** Envolver a coleta de grandes volumes de dados numéricos para fornecer uma base sólida para as decisões.
- **Mensuração e Valoração:** Aplicar métricas, índices e indicadores que ajudam a entender o impacto de diferentes fatores de forma mais objetiva.

Em muitas áreas, essa transição permite uma maior objetividade, consistência e a possibilidade de replicar e validar resultados, o que pode ser mais desafiador em abordagens qualitativas.

No contexto de qualidade de software, a "Transição para uma Visão Quantitativa" refere-se ao movimento de adotar métricas, indicadores e abordagens baseadas em dados numéricos para avaliar, monitorar e melhorar a qualidade do software. Em vez de confiar apenas em métodos qualitativos, como revisões manuais, feedback de usuários ou análise subjetiva de código, a transição para uma visão quantitativa busca utilizar dados concretos e medíveis para tomar decisões sobre o processo de desenvolvimento, a performance do software e a satisfação do usuário.

## 1. Definição de Métricas Quantitativas

- **Cobertura de Testes:** Mede a porcentagem de código coberta por testes automatizados. Uma cobertura maior pode indicar que o software foi mais exhaustivamente testado.
- **Taxa de Defeitos:** Mede a quantidade de defeitos encontrados por unidade de código (por exemplo, defeitos por mil linhas de código). Isso ajuda a avaliar a estabilidade do software.
- **Velocidade de Entrega:** A quantidade de funcionalidades entregues por período de tempo, que reflete a agilidade e a eficiência do time de desenvolvimento.
- **Tempo de Resolução de Defeitos:** Mede quanto tempo leva para corrigir um erro ou defeito após ser identificado, ajudando a avaliar a eficácia da equipe de desenvolvimento na resolução de problemas.

## 2. Medição da Qualidade do Código

- **Complexidade Ciclomática:** Uma métrica que quantifica a complexidade do código, o que pode ajudar a identificar trechos de código difíceis de entender ou manter.
- **Manutenibilidade:** Indicadores relacionados à facilidade de modificar ou expandir o software no futuro, como a quantidade de código duplicado ou o número de dependências externas.

### 3. Monitoramento da Performance

- **Tempo de Resposta e Latência:** Mede o desempenho do software, como a rapidez com que o sistema responde às solicitações dos usuários. Isso é fundamental para avaliar a experiência do usuário.
- **Uso de Recursos:** Mede o consumo de recursos (como CPU, memória, largura de banda) durante a execução do software, ajudando a identificar gargalos e áreas de otimização.

### 4. Medição da Satisfação do Usuário

- **Net Promoter Score (NPS):** Uma métrica de satisfação que mede a lealdade dos usuários e sua disposição para recomendar o software.
- **Taxa de Retenção de Usuários:** Mede a porcentagem de usuários que continuam a usar o software ao longo do tempo, um indicativo direto da satisfação com a qualidade do produto.

### 5. Análise Preditiva e Proativa

Com uma base quantitativa sólida, as equipes podem utilizar técnicas analíticas avançadas, como modelos preditivos, para identificar tendências, riscos e problemas futuros antes que eles se tornem críticos. Por exemplo, analisar padrões históricos de defeitos pode ajudar a prever onde futuros problemas podem ocorrer, permitindo ações preventivas.

### Benefícios da Transição para uma Visão Quantitativa na Qualidade de Software:

- **Objetividade e Consistência:** As métricas quantitativas oferecem dados objetivos que podem ser comparados ao longo do tempo e entre diferentes projetos ou equipes.
- **Melhoria Contínua:** O monitoramento constante de métricas pode ajudar a identificar áreas que precisam de melhoria e permitir ajustes rápidos no processo de desenvolvimento.
- **Decisões Informadas:** Basear decisões sobre dados concretos permite que as equipes priorizem problemas com base em sua gravidade e impacto real.

- **Eficiência e Previsibilidade:** A análise quantitativa pode aumentar a eficiência do desenvolvimento ao permitir a automação de testes e o uso de ferramentas para medir a qualidade de forma contínua.

## **Desafios:**

- **Coleta e Interpretação de Dados:** A coleta excessiva de métricas pode se tornar um desafio, já que nem todos os dados são igualmente valiosos. É crucial escolher as métricas certas.
- **Equilíbrio entre Quantidade e Qualidade:** Focar apenas nas métricas quantitativas pode levar a uma ênfase excessiva em números, enquanto aspectos qualitativos da experiência do usuário e da inovação podem ser negligenciados.
- **Complexidade na Implementação:** Implementar uma abordagem quantitativa pode exigir ferramentas especializadas, integração com sistemas existentes e treinamento das equipes para usar essas ferramentas de maneira eficaz.

A transição para uma visão quantitativa na qualidade de software é uma mudança significativa que permite melhorar a objetividade, precisão e previsibilidade no desenvolvimento de software. Usar dados mensuráveis para avaliar e melhorar a qualidade do software ajuda a reduzir riscos, melhorar o desempenho e garantir uma experiência de usuário mais robusta e confiável. No entanto, a implementação bem-sucedida exige escolher métricas relevantes e equilibrá-las com fatores qualitativos para garantir que o produto final atenda às necessidades dos usuários.

## **BIBLIOGRAFIA**

<https://blogdaqualidade.com.br/gurus-da-qualidade-david-a-garvin/#:~:text=Ele%20delineou%20as%20características%20que,%2C%20serviço%2C%20estética%20e%20percepção.>

<https://www.devmedia.com.br/qualidade-de-software-engenharia-de-software-29/18209>

<https://blog.mhavila.com.br/2010/06/20/modelo-de-qualidade-de-software-de-mccall/>  
[https://pt.wikipedia.org/wiki/ISO/IEC\\_9126](https://pt.wikipedia.org/wiki/ISO/IEC_9126)