

DANILO S ABREU

RA: 614106799

## **DESENVOLVIMENTO QUALIDADE**

Trabalho de conclusão do curso de PÓS-GRADUAÇÃO LATO SENSU em Engenharia de Software, da Universidade Nove de Julho, sob a orientação do Professor MS. Marcos Ribeiro.

**UNIVERSIDADE NOVE DE JULHO  
SÃO PAULO - 2015**

## 1. QUALIDADE

---

### 1.1 Conceito

Qualidade é um conceito subjetivo e está relacionado às percepções de cada indivíduo sob os diversos fatores como: cultura, crenças, valores, posição social e entre outros. Em termos de produtos e serviços qualidade se relaciona diretamente com as exigências, expectativas e necessidades dos clientes, como por exemplo: valor agregado, relação custo-benefício, durabilidade e acessibilidade.

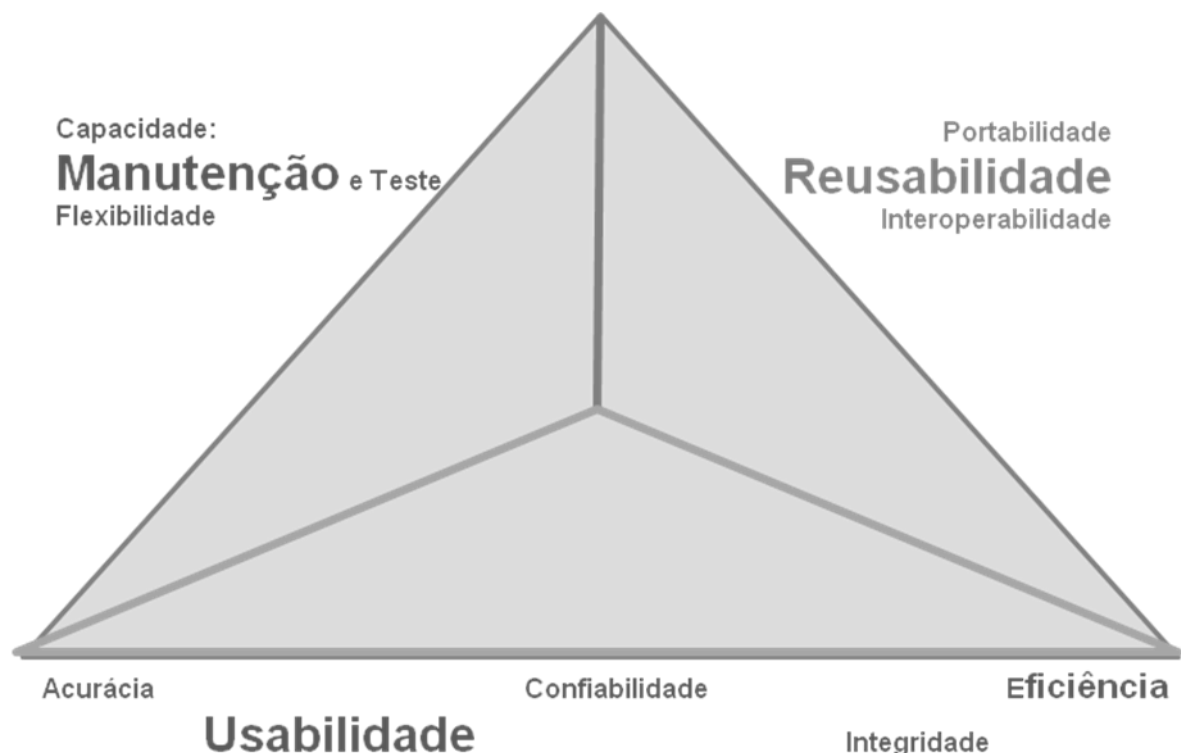
Seguindo nesta direção temos algumas definições importantes já conhecidas e estudadas. "Qualidade é a habilidade de um conjunto de características inerentes a um produto, componente de produto ou processo atenderem aos requisitos dos clientes" (SEI 2006). Já para a norma ISO 8402 (ABNT, 1994), "Qualidade é a totalidade das características de uma entidade, que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas dos *stakeholders*".

Podemos observar que qualidade está diretamente vinculada com as expectativas e necessidades do cliente. Segundo Philip Crosby (CROSBY, 1979) qualidade é estabelecer conformidade com os requisitos. Para a norma ISO9000 estabelece também que a "qualidade é o grau no qual um conjunto de características inerentes satisfaz a requisitos" (ISO9000: 2005).

Desta forma, assim como para outros setores, qualidade é fator crítico de sucesso para a indústria de software (MR-MPS-SW:2012, 2012), tendo em vista o acelerado avanço tecnológico rápido e inovador e também cada vez mais presente no cotidiano dos indivíduos.

## 1.2 Qualidade de Software

A Qualidade de Software é também algo complexo, no entanto, alguns modelos ou conjunto de atributos podem classificar a qualidade. McCall (MACCALL, 1977) propõe uma categorização dos fatores da qualidade de software, com ênfase em três aspectos importantes: Características manutenção, operação, e sua capacidade de adaptabilidade.



**Figura 1. Modelo McCall (1977)**

Boehm acrescentou algumas características com o modelo de McCall com ênfase na capacidade de manutenção do produto de software (BOTELLA, 2004). Além disso, o modelo inclui considerações envolvidas na avaliação de um software produto no que diz respeito à utilidade do programa. O modelo proposto por Boehm é similar ao modelo de McCall e que representa uma hierárquica estrutura de características, cada um dos quais contribui para a qualidade total. A noção de Boehm inclui as necessidades dos usuários, assim como McCall, no entanto, o modelo de Boehm contém apenas um diagrama, sem qualquer sugestão de como medir as características de qualidade (BEHKAMAL, 2008).

Outro modelo também muito conhecido foi o FURPS (WATSON, 2006) proposto por Robert Grady e HewlettPackard Co., que decompõe características em duas categorias diferentes de requisitos:

Os requisitos funcionais: Definido pela entrada e saída esperada.

Os requisitos não funcionais: Usabilidade, confiabilidade, desempenho e capacidade de suporte.

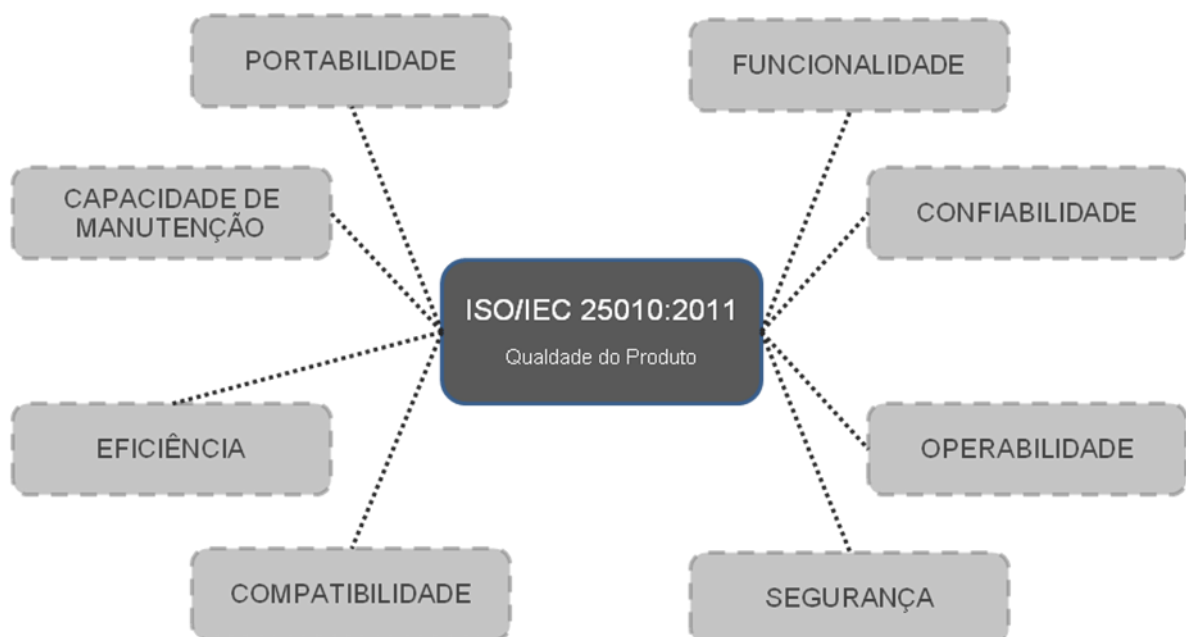
Uma desvantagem do modelo FURPS é que ele não leva em conta a portabilidade do produto de software (BEHKAMAL, 2008).

Muitos outros modelos surgiram, contudo, havia a necessidade de um modelo padrão, e com isso deu-se início a ISO/IEC JTC1 que buscou desenvolver consensos necessários e incentivar a normalização a nível mundial sobre os modelos de qualidade de software. As primeiras considerações foram feitas em 1978, e dando início em 1985 a ISO/IEC 9126. A ISO/IEC 9126 é parte da família ISO 9000, que é o padrão mais importante para a garantia da qualidade, no entanto, foi revisada em 2011 pela norma ISO/IEC 25010:2011 (ISO/IEC 25010, 2011), também conhecida como **SQuaRE** (*Software product Quality Requirements and Evaluation*).

A norma ISO/IEC 25010, propõe um conjunto de atributos de qualidade, distribuídos em 8 (oito) conjuntos principais, com cada uma deles divididos em subcaracterísticas descritas pela norma. A nova norma tem oito conjuntos de características de qualidade do produto, em contraste com a ISO 9126 com seis, e 31 subcaracterísticas.

- Funcionalidade: A capacidade de um software prover funcionalidades que satisfaçam o usuário em suas necessidades declaradas e implícitas, dentro de um determinado contexto de uso.
- Confiabilidade: O produto se mantém no nível de desempenho nas condições estabelecidas.

- Operabilidade: Observa a adequação, fácil utilização, não exposição de erros ao usuário, cuidado com a estética da interface com o usuário e acessibilidade,
- Segurança: Preocupação com confidencialidade, integridade, responsabilidade e autenticidade.
- Compatibilidade: coexistência e interoperabilidade.
- Eficiência: O tempo de execução e os recursos envolvidos são compatíveis com o nível de desempenho do software.
- Capacidade de manutenção: A capacidade (ou facilidade) do produto de software ser modificado, incluindo tanto as melhorias ou extensões de funcionalidade quanto às correções de defeitos, falhas ou erros.
- Portabilidade: A capacidade de o sistema ser transferido de um ambiente para outro.



**Figura 1. Figura ISO/IEC 25010 (2011)**

Podemos observar que temos a partir da ISO/IEC 15020 um conjunto bem amadurecido e suficientemente capaz que cobrir os diversos modelos já propostos de forma integrada.

Ainda assim podemos dizer que existem duas visões de qualidade de software, uma dos clientes e outra dos que desenvolvem o software, mas ambas concordam que o software não pode ter defeitos. O cliente avalia o software sem conhecer seus aspectos internos, está apenas interessado na facilidade do uso, no desempenho, na confiabilidade dos resultados obtidos e também no preço do software. Os que desenvolvem o software avaliam aspectos internos como taxa de defeitos, confiabilidade, facilidade de manutenção e também aspectos de conformidade em relação aos requisitos dos clientes (ISO, 1991), (CAROSIA, 2003). Para PRESSMAN (PRESSMAN, 1995) essas duas visões podem ser exemplificadas como:

1. Fatores que podem ser medidos diretamente (por exemplo, erros/KLOC/unidade de tempo)
2. Fatores que podem ser medidos apenas indiretamente (por exemplo, usabilidade ou capacidade de manutenção).

### **1.3 Qualidade do processo**

Antes de discutir sobre qualidade do processo, devemos definir o que é entendido como processo. Processo são um conjunto de passos ordenados, constituídos por atividades, métodos, práticas e transformações, usado para atingir uma meta. Esta meta geralmente está associada a um ou mais resultados concretos finais, que são os produtos da execução do processo (PÁDUA, 2003).

Um processo de software é um conjunto de ferramentas, métodos e práticas usadas para produzir software. O processo de software é representado por um conjunto sequencial de atividades, objetivos, transformações e eventos que integram estratégias para cumprimento da evolução de software (PRESSMAN, 1995). A

norma ISO 9000, define processo como um conjunto de atividades inter-relacionadas ou interativas que transformam entradas em saídas.

Como a qualidade do produto pode ser vista como uma consequência da qualidade do processo, os certificados mais valiosos são aqueles que certificam o processo de produção de um produto e não os que certificam simplesmente o produto. Os estudos sobre qualidade, mais recentemente, são voltados para o melhoramento do processo de software, pois ao garantir a qualidade do processo, já se está dando um grande passo para garantir também a qualidade do produto (CAROSIA, 2003).

Um processo de qualidade, de maneira geral, é ter um processo documentado, estabelecidos a partir de modelos amadurecidos, e que buscam a melhoria contínua. Os principais modelos da atualidade são o *Capability Maturity Model Integration* (CMMI) e a norma ISO/IEC 15504-5. No cenário nacional, o Modelo de Referência MPS que faz parte do MPS.BR (Melhoria de Processo do Software Brasileiro).

## REFERÊNCIAS

**BEHKAMAL**, Behshid., Mohsen Kahani, Mohammad Kazem Akbari. *Customizing ISO 9126 quality model for evaluation of B2B applications*. Copyright 2008 Elsevier B.V. September 2008.

**BOTELLA P.**, X. Burgues, J.P. Carvallo, X. Franch, C. Quer, Using Quality Models for Assessing COTS Selection, in: Proceeding of MPEC'04 and ICSE'04, 2004.

**CAROSIA**, J. S. Levantamento da qualidade do processo de software com foco em pequenas organizações / J. S. Carosia. – São José dos Campos: INPE, 2003.

**ISO/IEC 25010**. *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*, 2011. Disponível em [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=35733](http://www.iso.org/iso/catalogue_detail.htm?csnumber=35733), acessado em 14 Março 2015.

**MACCALL**, J., P. Richards e G. Walters, “*Factors in Software Quality*”, três volumes, NTIS AD-AO49-014, 015, 055. novembro de 1977.

**MEYER**, B., *Object-oriented Software Construction*, Prentice-Hall, 1988.

**MR-MPS-SW:2012**. MPS.BR - Melhoria de Processo do Software Brasileiro. Guia Geral MPS de Software. Copyright 2012 - SOFTEX, 2012.

**PÁDUA**, Wilson de P.F. Engenharia de Software: fundamentos, métodos e padrões, 2003

**WATSON**, Mike. *Managing Smaller Projects: A Practical Approach*. Multi-Media Publications Inc., 2006.