VOLTAR

Qualidade de Produto de Software - Norma ISO 25010

Apresentar a norma ISO/IEC 25000 em relação a medição da qualidade de produto de software.

NESTE TÓPICO

Qualidade de Software – Definição

A determinação do grau de qualidade de produtos é algo subjetivo, pois diferentes tipos de necessidade exigem diferentes critérios de avaliação. Isso não é diferente para os produtos de software, exigindo necessidade de definição clara dos objetivos que se pretende atingir em projetos de software.

Os autores Koscianski e Soares (2007) afirmam ser preciso enumerar características desejáveis do software, que idealmente devem incluir valores quantitativos a serem respeitados. Eles complementam dizendo que, quando o produto é medido de acordo com essas características, torna-se factível o diagnóstico em relação ao seu grau de qualidade.

Existem normas que determinam as peculiaridades que devem ser consideradas para avaliar a qualidade de um produto ou serviço. Elas trazem diretrizes para diferentes características de software, as quais podem ser mensuradas. Fornecem possíveis parâmetros que, quando estabelecidos no início do projeto, servem de critérios para identificar o grau de qualidade que o software deve atingir.

Uma importante série de normas é a ISO/IEC 25000: 2008. Essa série visa estabelecer Requisitos e Avaliação da Qualidade de Produto de Software (Sigla SQuaRe em inglês – Software product Quality Requirements and Evaluation) (ABNT, 2008).

Você estudará a seguir a evolução da série de normas ISO 25000 e mais especificamente a parte referente a requisitos de qualidade de software.

Norma ISO/IEC 25000 – Evolução Histórica

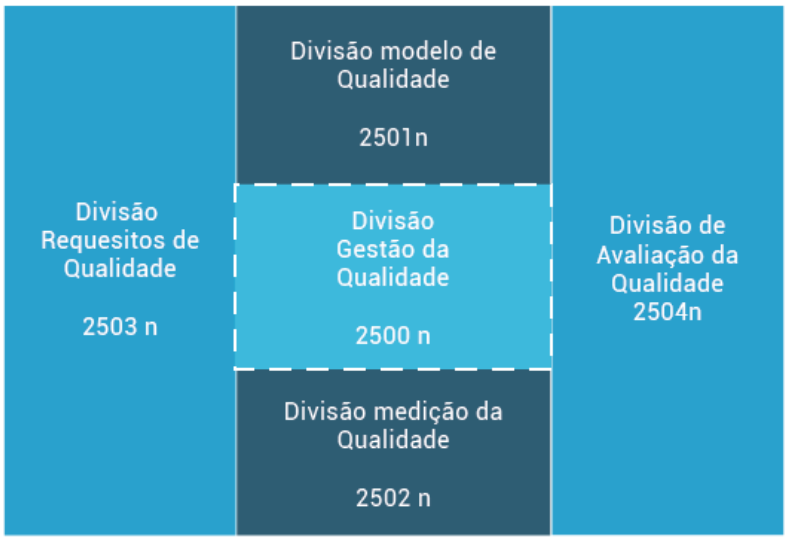
No ano de 1991 foi publicada a norma ISO 9126, com a definição de características a serem consideradas para a avaliação de um software. Ela foi dividida em quatro partes, sendo que a ISO 9126-1 descreveu o "Modelo de Qualidade" abordando um conjunto de peculiaridades e subcaracterísticas, estabelecendo, assim, um modelo de qualidade com os seguintes componentes:

- Processo de desenvolvimento: o modo como o software é produzido, isto é, o método utilizado para o seu desenvolvimento influencia na sua qualidade final.
- Produto: deve atender às necessidades que dele são esperadas e compreende requisitos que podem ser divididos entre internos e externos, assim divididos devido à forma como são aferidos.
- Qualidade em uso: refere-se à qualidade do produto, segundo o ponto de vista do usuário.

Essa norma, a ISO 9126 (Qualidade de Produto de Software) era utilizada juntamente com a ISO 14598 (Avaliação de Produto de Software). O conjunto dessas normas ficou conhecido como SQuaRE, cujo objetivo é "obter uma série logicamente organizada, rica e unificada, cobrindo dois processos principais: especificação de requisitos de software e avaliação da qualidade de software" (ABNT, 2008).

A série ISO/IEC 25000 é uma evolução das séries ISO/IEC 9128 e 14598, onde ocorreu uma reorganização do material existente em ambas, sem grandes mudanças, mas agrupando em uma única série, a 25000 (KOSCIANSKI e SOARES, 2007). O projeto de unificação na série 25000 ainda está em andamento, sendo que a ISO 9126-1 foi substituída pela ISO 25010. Porém, essa última ainda está aguardando tradução oficial para o português no Brasil pela Associação Brasileira de Normas Técnicas.

A organização da série ISO/IEC 25000 ficou da seguinte forma:



Organização da série ISO/IEC 25000 SQuaRE de Normas

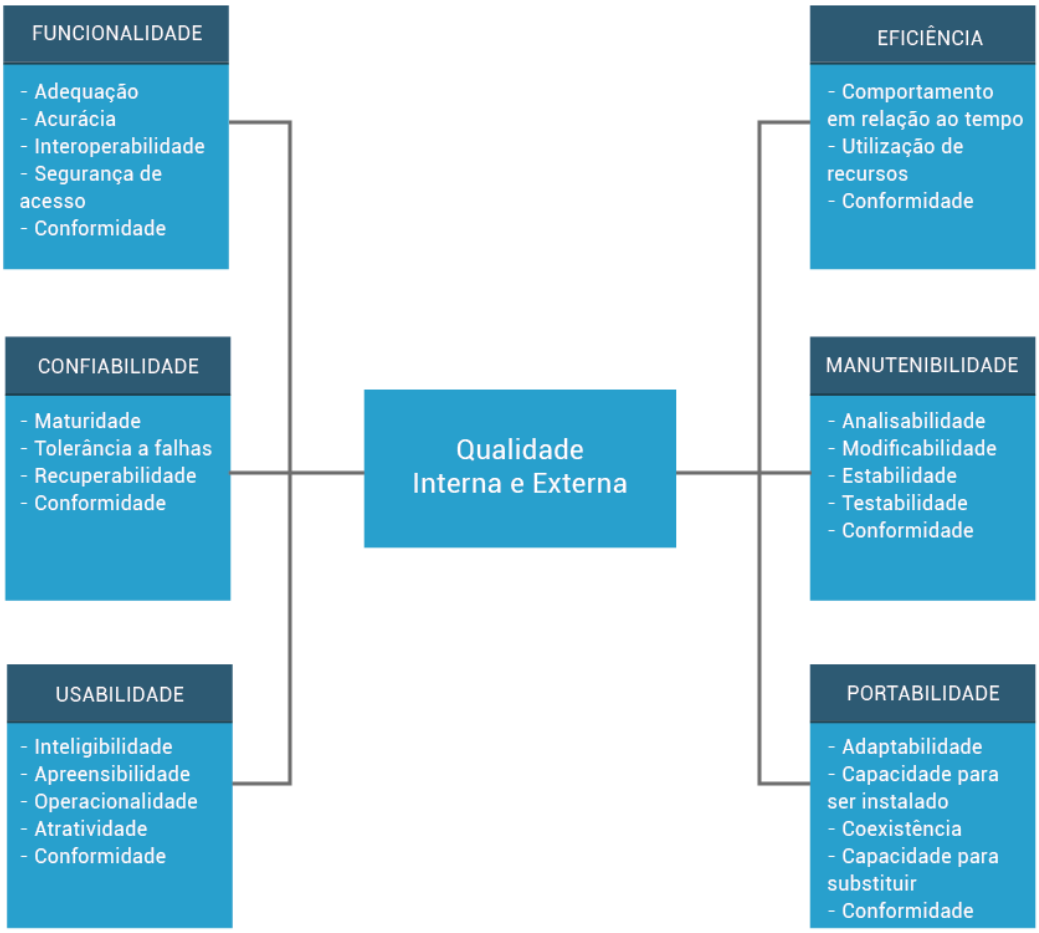
Fonte: ABNT NBR ISO/IEC 25000

A norma 9126 foi cancelada pela ABNT, sendo que a parte 9126-1 ainda é utilizada como referência, dentro da 25030, até que toda a série 25000 seja editada. A seguir será apresentada uma descrição do modelo de qualidade de produto de software descrito pelas normas.

Modelo de Qualidade de Software

Conforme vimos, a qualidade de um software está relacionada com o quanto o produto de software consegue atender as necessidades declaradas, sejam elas explícitas ou mesmo implícitas, ou seja, aquelas que mesmo não sendo formalmente declaradas pelo usuário do sistema, são necessárias para o seu correto funcionamento. (ABNTa, 2008).

O modelo recomendado pela série ISO 25000, propõem determinados atributos que estabelecem, de acordo com a sua aferição, o grau de qualidade do software. Esses atributos encontram-se distribuídos em seis características principais, cada uma se dividindo em subcaracterísticas.



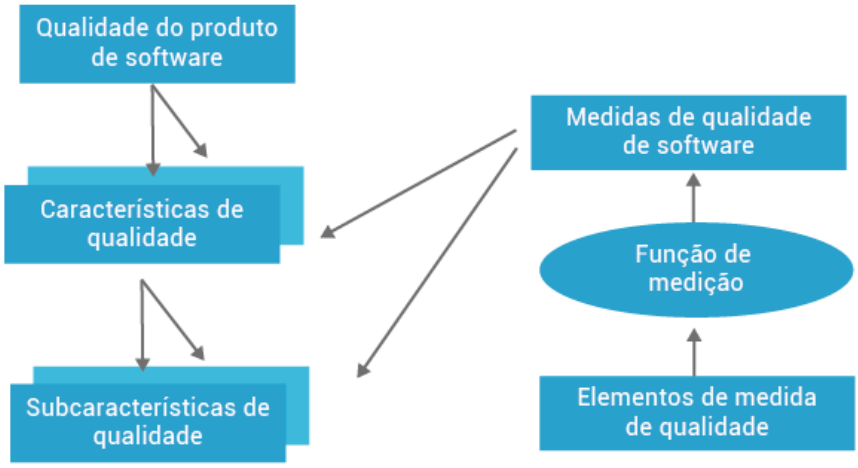
Características e subcaracterísticas dos atributos de qualidade de software = norma ISO/IEC 25030

De acordo com a figura apresentada acima, pode-se notar que, em todas as particularidades, temos um subaspecto com o nome de **conformidade**. A conformidade é utilizada para avaliar o quanto o software obedece aos requisitos, legislação, padronização ou normalização aplicável ao contexto.

Modelo de Medição de Qualidade de Software

Para a aferição da qualidade do software é necessário que se estabeleçam métodos de medição. Esses métodos possibilitam a quantificação de um atributo em relação a uma escala específica. Por meio da aplicação de funções de medição, é possível a determinação, em termos quantitativos, de quanto o software atende as características e subcaracterísticas dos atributos de qualidade, ou seja, qual o seu grau de qualidade.

A figura a seguir apresenta o modelo de referência da ISO 25030 para implementar um processo de medição da qualidade de produto de software.



Modelo de referência para a medição da qualidade de produto de software

Fonte: ABNT NBR ISO/IEC 25030

Com a aplicação do processo de medição é possível realizar interpretações, de forma objetiva, de quanto o produto de software atende aos atributos de qualidade destacados a ele, atribuindo um grau preciso de qualidade ao software.

Na sequência você terá a oportunidade de estudar os conceitos de cada uma das seis características e subcaracterísticas dos atributos de qualidade de software que a norma estabelece.

Funcionalidade

Característica que mede a capacidade de um software em prover o usuário de funcionalidades que o satisfaçam, tanto em suas necessidades declaradas e explícitas, dentro de um determinado contexto de uso, como nas suas necessidades não declaradas ou implícitas a um produto de software. Esse aspecto divide-se nas seguintes subparticularidades:

- **Adequação**: medida que avalia o quanto o conjunto de funcionalidades é adequado às necessidades do usuário.

- **Acurácia** (ou precisão): mede a capacidade do software de fornecer resultados precisos, ou dentro dos critérios necessários de precisão.
- **Interoperabilidade**: mede a capacidade do software de interagir com outros (Sistemas) instalados no ambiente.
- **Segurança**: mede a capacidade do software em proteger a informação que é de sua alçada e só fornecê-la às pessoas autorizadas.

Confiabilidade

Característica que avalia a capacidade do software de permanecer no nível de desempenho e nas condições estabelecidas, mantendo os dados íntegros e com reduzida quantidade de falhas. Essa peculiaridade possui as seguintes subcaracterísticas:

- **Maturidade**: capacidade do software de evitar falhas decorrentes de defeitos provocados nele próprio.
- **Tolerância a falhas**: capacidade do software de manter um funcionamento adequado, mesmo ocorrendo defeitos em suas interfaces externas.
- **Recuperabilidade**: capacidade de um software de restabelecer os níveis de desempenho e integridade dos dados após ter sofrido uma falha.

Usabilidade

Capacidade do produto de software de ser compreendido, ter seu funcionamento aprendido, ser operado e ser atraente ao usuário. Um software com um alto grau de usabilidade é conhecido no mercado como “de uso amigável”. Característica composta pelos seguintes subaspectos:

- **Inteligibilidade**: facilidade com que o usuário pode compreender as funcionalidades do software e avaliar se este pode ser usado para satisfazer suas necessidades específicas.
- **Apreensibilidade**: facilidade de aprendizado do sistema para os seus potenciais usuários.
- **Operacionalidade**: como o produto facilita a sua operação por parte do usuário, incluindo a maneira como tolera erros de operação.
- **Atratividade**: adequação das informações prestadas para o usuário e os requintes visuais utilizados na interface gráfica do software.

Eficiência

Avalia se os recursos utilizados pelo software são compatíveis com o desempenho. Possui as seguintes subparticularidades:

- **Comportamento em relação ao tempo**: avalia se o tempo de resposta e/ou de processamento estão dentro das especificações.
- **Utilização de recursos**: mede os recursos consumidos e a capacidade do sistema em utilizar os recursos disponíveis.

Manutenibilidade

Mede a capacidade do produto de software ser modificado, sejam essas alterações de caráter evolutivo, adaptativo, preventivo ou corretivo. Possui as seguintes subcaracterísticas:

- **Analisabilidade**: facilidade em diagnosticar eventuais problemas e identificar as causas das deficiências ou falhas.
- **Modificabilidade**: facilidade com que o comportamento do software pode ser modificado.
- **Estabilidade**: capacidade do software de evitar efeitos colaterais decorrentes de modificações introduzidas.
- **Testabilidade**: capacidade de testar o sistema modificado.

Portabilidade

Capacidade de um software ser transferido de um ambiente para outro. Possui as seguintes subcaspectos:

- **Adaptabilidade**: capacidade do software de ser adaptável a diferentes ambientes, sem a necessidade de alterações profundas.
- **Capacidade para ser instalado**: facilidade com que se pode instalar o sistema em um novo ambiente.
- **Coexistência**: facilidade de o novo software conviver com outros, instalados previamente no mesmo ambiente.
- **Capacidade para substituir**: capacidade que o software tem de substituir outro sistema, no mesmo contexto de uso e ambiente.

Qualidade no Ciclo de Vida do Software

O Modelo SQuaRe é voltado à medição da qualidade em todo o ciclo de vida dos produtos de software. A partir de uma primeira mensuração, é possível analisar o andamento do projeto (desenvolvimento ou manutenção de software) e gerenciar a qualidade dos produtos de software (KOSCIANSKI e SOARES, 2007).

São considerados três tipos de qualidade envolvendo um produto de software:

- Qualidade do software em uso

- Qualidade externa de software
- Qualidade interna do software

O primeiro tipo, **Qualidade em Uso**, se refere à visão do usuário. É o software em seu sistema operacional realizando tarefas específicas para usuários específicos.

A visão da **Qualidade Externa** funciona como uma "caixa-preta", onde não se tem conhecimento da arquitetura do software, sobre seu código ou como é o funcionamento interno. A validação de um produto, com a realização de testes de funcionamento, corresponde a qualidade externa.

Quando o ponto de vista é em relação à arquitetura interna do software, o tipo de qualidade a ser avaliado é o da **Qualidade Interna**. São levadas em consideração as propriedades estáticas do software como, por exemplo, a qualidade da organização do código ou sua complexidade algorítmica.

Essas diferentes visões são utilizadas para organizar o gerenciamento da qualidade do ciclo de vida do software. A qualidade em uso tem probabilidade de alcançar maior êxito quando se é assegurado que o software tenha uma boa qualidade externa, que pode ser alcançada com um alto grau de qualidade interna.

Agora que você já estudou esta aula, resolva os exercícios e verifique seu conhecimento. Caso fique com alguma dúvida, leve a questão ao fórum e divida com seus colegas e professor.

Quiz

Exercício

Qualidade de Produto de Software - Norma ISO 25010

INICIAR

Referências

ABNT. *ABNT NBR ISO/IEC 25000*: 2008. Rio de Janeiro: [s.n.], 2008.

ABNTa. *ABNT NBR ISO/IEC 25030*. Rio de Janeiro: [s.n.], 2008.

KOSCIANSKI, A.; SOARES, M. D. S. *Qualidade de software*: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2. ed. São Paulo: Novatec, 2007.

PAULA FILHO, W. D. P. *Engenharia de Software*: fundamentos, métodos e padrões. Rio de Janeiro: LTC, 2003.

PFLEEGER, S. L. *Engenharia de Software*: teoria e prática. 2. ed. São Paulo: Pearson, 2004.

PRESSMAN, R. S. *Engenharia de Software*: Uma Abordagem Profissional. 7. ed. Porto Alegre: McGraw Hill, 2011.

SOMMERVILLE, I. *Engenharia de Software*. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.

Avalie este tópico

ANTERIOR
Melhoria Contínua nos Processos

Índice

PRÓXIMO
CMMI - Definição e Conceitos

Ajuda?
(https://ava.un
idCurso=)

® Todos os direitos reservados

Biblioteca (https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)
Portal Uninove (http://www.uninove.br)
Mapa do Site