

ISO/IEC 9126

Origem: Wikipédia, a enciclopédia livre.

ISO/IEC 9126 é uma norma ISO para qualidade de produto de software, que se enquadra no modelo de qualidade das normas da família 9000. A norma brasileira correspondente é a NBR ISO/IEC 9126.

Índice

- 1 Modelo de Qualidade de Software
- 2 Modelo de Qualidade da Norma ISO 9126
 - 2.1 Funcionalidade
 - 2.2 Confiabilidade
 - 2.3 Usabilidade
 - 2.4 Eficiência
 - 2.5 Manutenibilidade
 - 2.6 Portabilidade
- 3 Ver também
- 4 Bibliografia

Modelo de Qualidade de Software

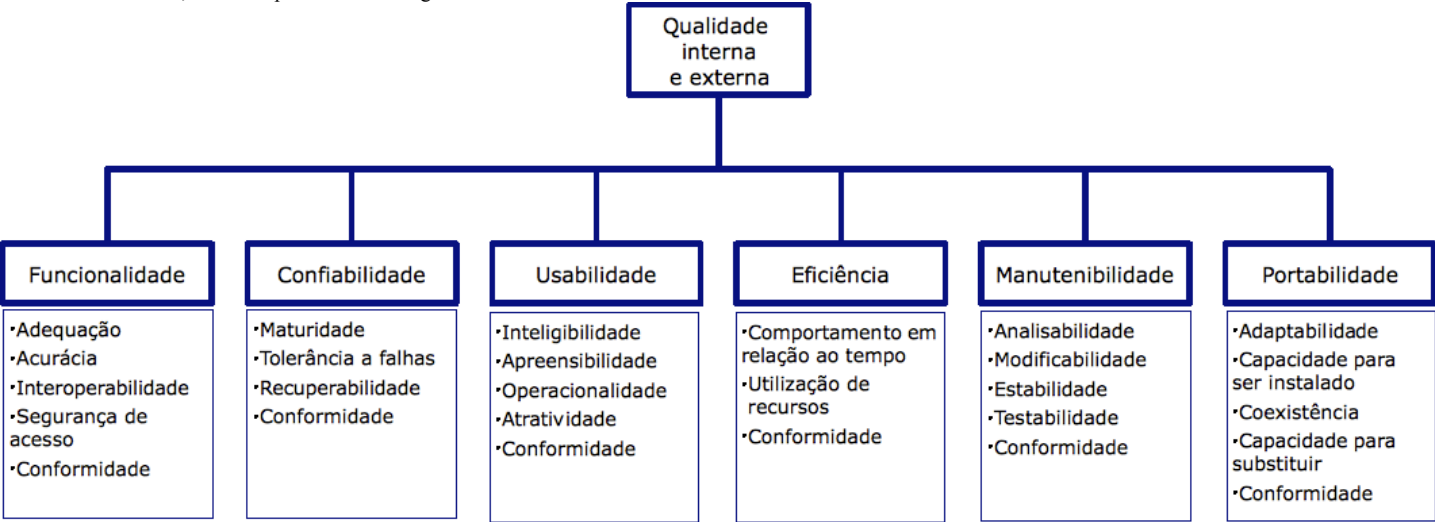
A qualidade de um sistema de software pode ser entendida de diversas formas e utilizando diferentes abordagens.

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** de desenvolvimento, cuja qualidade afeta a qualidade do produto de software gerado e é influenciado pela natureza do produto desenvolvido;
- **Produto**, compreendendo os atributos de qualidade do produto (sistema) de software. Estes atributos de qualidade podem ser divididos entre atributos **internos** e **externos**. Estes se diferenciam pela forma como são aferidos (interna ou externamente ao produto de software) e em conjunto compõem a qualidade do produto de software em si;
- **Qualidade em uso** que consiste na aferição da qualidade do software em cada contexto específico de usuário. Esta é, também, a qualidade percebida pelo usuário.

Modelo de Qualidade da Norma ISO 9126

A norma 9126 se foca na qualidade do produto de software, propondo **Atributos de Qualidade**, distribuídos em seis características principais, com cada uma delas divididas em sub-características, conforme podemos ver na figura abaixo:



No nível mais alto temos as características de qualidade e nos quadros abaixo as suas sub-características. Cada característica/sub-característica compõe um **Atributo de Qualidade** do software.

Note que em todas as características temos uma sub-categoria com o nome de **Conformidade**. A conformidade é utilizada para avaliar o quanto o software obedece aos requisitos de legislação e todo o tipo de padronização ou normalização aplicável ao contexto.

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.

Suas sub-características são:

- **Adequação**, que mede o quanto o conjunto de funcionalidades é adequado às necessidades do usuário;
- **Acurácia** (ou precisão) representa a capacidade do software de fornecer resultados precisos ou com a precisão dentro do que foi acordado/solicitado;
- **Interoperabilidade** que trata da maneira como o software interage com outro(s) sistema(s) especificados;
- **Segurança** mede a capacidade do sistema de proteger as informações do usuário e fornecê-las apenas (e sempre) às pessoas autorizadas
- „, Segurança também pode estar dirigida em, processar gerar e armazenar as informações.

Confiabilidade

O produto se mantém no nível de desempenho nas condições estabelecidas.

Suas sub-características são:

- **Maturidade**, entendida como sendo a capacidade do software em evitar falhas decorrentes de defeitos no software;
- **Tolerância a Falhas** representando a capacidade do software em manter o funcionamento adequado mesmo quando ocorrem defeitos nele ou nas suas interfaces externas;
- **Recuperabilidade** que foca na capacidade de um software se recuperar após uma falha, restabelecendo seus níveis de desempenho e recuperando os seus dados;

Usabilidade

A capacidade do produto de software ser compreendido, seu funcionamento aprendido, ser operado e ser atraente ao usuário.

Note que este conceito é bastante abrangente e se aplica mesmo a programas que não possuem uma interface para o usuário final. Por exemplo, um programa batch executado por uma ferramenta de programação de processos também pode ser avaliado quanto a sua usabilidade, no que diz respeito a ser facilmente compreendido, aprendido, etc. Além disto, a operação de um sistema é uma interface Humano-Computador (ver IHC) sujeita às avaliações de usabilidade.

Suas sub-características são:

- **Inteligibilidade** que representa a facilidade com que o usuário pode compreender as suas funcionalidades e avaliar se o mesmo pode ser usado para satisfazer as suas necessidades específicas;
- **Apreensibilidade** identifica a facilidade de aprendizado do sistema para os seus potenciais usuários;
- **Operacionalidade** é como o produto facilita a sua operação por parte do usuário, incluindo a maneira como ele tolera erros de operação;
- **Atratividade** envolve características que possam atrair um potencial usuário para o sistema, o que pode incluir desde a adequação das informações prestadas para o usuário até os requintes visuais utilizados na sua interface gráfica;

Eficiência

O tempo de execução e os recursos envolvidos são compatíveis com o nível de desempenho do software.

Suas sub-características são:

- **Comportamento em Relação ao Tempo** que avalia se os tempos de resposta (ou de processamento) estão dentro das especificações;
- **Utilização de Recursos** que mede tanto os recursos consumidos quanto a capacidade do sistema em utilizar os recursos disponíveis;

Manutenibilidade

A capacidade (ou facilidade) do produto de software ser modificado, incluindo tanto as melhorias ou extensões de funcionalidade quanto as correções de defeitos, falhas ou erros.

Suas sub-características são:

- **Analisabilidade** identifica a facilidade em se diagnosticar eventuais problemas e identificar as causas das deficiências ou falhas;
- **Modificabilidade** caracteriza a facilidade com que o comportamento do software pode ser modificado;
- **Estabilidade** avalia a capacidade do software de evitar efeitos colaterais decorrentes de modificações introduzidas;
- **Testabilidade** representa a capacidade de se testar o sistema modificado, tanto quanto as novas funcionalidades quanto as não afetadas diretamente pela modificação;

Portabilidade

A capacidade do sistema ser transferido de um ambiente para outro.

Como "ambiente", devemos considerar todo os fatores de adaptação, tais como diferentes condições de infra-estrutura (sistemas operacionais, versões de bancos de dados, etc.), diferentes tipos e recursos de hardware (tal como aproveitar um número maior de processadores ou memória). Além destes, fatores como idioma ou a facilidade para se criar ambientes de testes devem ser considerados como características de portabilidade.

Suas sub-características são:

- **Adaptabilidade**, representando a capacidade do software se adaptar a diferentes ambientes sem a necessidade de ações adicionais (configurações);
- **Capacidade para ser Instalado** identifica a facilidade com que pode se instalar o sistema em um novo ambiente;
- **Coexistência** mede o quão facilmente um software convive com outros instalados no mesmo ambiente;
- **Capacidade para Substituir** representa a capacidade que o sistema tem de substituir outro sistema especificado, em um contexto de uso e ambiente específicos. Este atributo interage tanto com *adaptabilidade* quanto com a *capacidade para ser instalado*;

Ver também

- [1] (http://pt.wikipedia.org/wiki/Processo_de_desenvolvimento_de_software) Processo de Desenvolvimento de Software
- [2] (http://pt.wikipedia.org/wiki/ISO/IEC_12207) ISO 12207, que trata dos processos do ciclo de vida do software
- CMMI, Capability Maturity Model Integration, é um modelo de referência que contém práticas necessárias à maturidade em disciplinas específicas.

Bibliografia

- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBRISO/IEC9126-1 **Engenharia de software - Qualidade de produto - Parte 1: Modelo de qualidade**. 2003.
- BARBACCI, M et al **Quality Attributes (em inglês)**, Pittsburgh, Carnegie Mellon University, 1995. Disponível em <http://www.sei.cmu.edu/publications/documents/95.reports/95.tr.021.html>.

Obtida de "http://pt.wikipedia.org/w/index.php?title=ISO/IEC_9126&oldid=37202023"