

# **Qualidade de Produto de Software**

**Ana Cervigni Guerra  
Regina Maria Thienne Colombo**

## **Sobre as autoras**

### **Ana Cervigni Guerra**

É engenheira elétrica e doutora em Engenharia Mecânica pela Unicamp. É pesquisadora em Qualidade de Software no Centro de Tecnologia da Informação Renato Archer – CTI/MCT, e participante da ABNT/CB-21 Computadores e processamento de dados. Tem experiência na área de automação, com ênfase em análise numérica e modelos matemáticos, em Ciência da Computação, qualidade de produto de software, atuando principalmente nos seguintes temas: qualidade de software, qualidade de processos, Engenharia de Software, sistema de gestão, métodos de avaliação de processos de software, aquisição de produtos de TI.

### **Regina M. Thienne Colombo**

É graduada em Ciência da Computação pela UFSCAR, mestre em Engenharia Mecânica pela Unicamp na área de qualidade e doutoranda na Poli-USP. Com vasta experiência em Engenharia e Qualidade de Software, trabalha como pesquisadora e coordenadora da Divisão de Qualificação em Software, no Centro de Tecnologia da Informação Renato Archer – CTI/MCT. É participante da ABNT/CB-21 Computadores e processamento de dados e da ISO/IEC SC7/WG6 (Engenharia de Software).

## *Agradecimentos*

Este trabalho não poderia ter sido realizado sem o apoio do Centro de Tecnologia da Informação Renato Archer - CTI, que nos possibilitou a oportunidade de aprofundar nossos conhecimentos teóricos e práticos na área de Qualidade de Software. Além das contribuições técnicas dos profissionais dessa instituição, algumas pessoas ajudaram, indiretamente, na elaboração deste trabalho, às quais prestamos homenagem:

Ao Sr. Romildo Monte, que iniciou o programa de Qualidade de Software no CTI (antigo CenPRA), agradecemos pela boa vontade, paciência e presteza que sempre dedica a todos que o procuram.

Em especial, agradecemos à grande equipe de profissionais que passaram pelo CTI e que têm desenvolvido trabalhos nessa área desde 1994. A essa equipe agradecemos pelo apoio técnico deixado como legado na nossa instituição. Sem ela não nos teria sido possível tanto conhecimento a ser disseminado. Reunimos esse material valioso, para que não se tornasse mais um documento técnico interno e para que seja acessível a quem interessar. Tudo isso é resultado de pesquisas da instituição, de interação com empresas, outras instituições de Pesquisa e Desenvolvimento - P&D e Universidades, e de trabalhos acadêmicos em cooperação com a Unicamp - trabalhos que garantiram o título de mestre a algumas pessoas da equipe, entre elas uma das autoras.

Ao Sr. Kival Chaves Weber, um ícone no país, quando falamos em Qualidade de Software, agradecemos pelo incentivo constante ao nosso trabalho, por todas as vezes que nos motivou na pesquisa e aplicação do conhecimento.

Aos revisores do livro, pelas sugestões tão apropriadas e pertinentes e pela paciência. Ainda agradecemos em especial o Sr. Danilo Scalet, a Sra. Denise Carneiro, a Sra. Diva da Silva Marinho e o Prof. Gilberto Cláudio P. Balthazar.

A todos o nosso carinho.

# Sumário

<b>Introdução .....</b>	<b>10</b>
<b>Capítulo 1.....</b>	<b>13</b>
<b>Panorama tecnológico.....</b>	<b>13</b>
<b>Capítulo 2.....</b>	<b>17</b>
<b>Qualidade de Software.....</b>	<b>17</b>
2.1 Qualidade de produto.....	17
2.2 Qualidade de produto de software .....	19
2.2.1 Aspectos gerais e evolução do software .....	20
2.2.2 Software e suas características como produto.....	23
2.2.3 Iniciativas para a qualidade de produto de software .....	25
2.3 Qualidade de processo de software.....	27
2.3.1 Aspectos gerais .....	28
2.3.2 Iniciativas para a qualidade de processo de software.....	30
2.3.3 Comparação entre os modelos .....	39
<b>Capítulo 3.....</b>	<b>42</b>
<b>Categorias de produtos de software e avaliação.....</b>	<b>42</b>
3.1 Categorias segundo Pressman.....	42
3.2 Caracterizações da norma IEEE 1062 .....	43
3.3 Categorias para prêmios de software .....	44
3.4 Avaliação de produto de software .....	47
3.4.1 Método de avaliação genérico (MEDE-PROS®) .....	48
3.4.2 Método de avaliação especialista.....	51
3.5 Avaliação e certificação de produto de software .....	54
3.6 Avaliação e teste de software .....	58
<b>Capítulo 4.....</b>	<b>60</b>
<b>Modelos de qualidade .....</b>	<b>60</b>
4.1 Modelo de McCall .....	61
4.2 O modelo da norma NBR ISO/IEC 9126-1 .....	64
4.2.1 Diretrizes para uso da norma NBR ISO/IEC 9126-1 .....	65
4.2.2 Características e subcaracterísticas de qualidade de software.....	65
4.3 A série ISO/IEC 25000.....	70
4.4 O modelo de qualidade do MEDE-PROS®.....	77
4.5 Modelo de qualidade do PNAFM.....	81
4.6 Modelo de qualidade de componentes de software .....	83
4.6.1 Descrição do componente .....	86
4.6.2 Documentação do usuário.....	89
4.6.3 Componentes no DBC .....	91
4.6.4 Componente integrado no SBC .....	94
<b>Capítulo 5.....</b>	<b>98</b>

<b>Requisitos e avaliação da qualidade de software .....</b>	<b>98</b>
5.1 Processo de avaliação da qualidade de produto de software .....	98
5.1.1 Norma NBR ISO/IEC 14598-1 e NBR ISO/IEC 14598-5 .....	102
5.2 Especificação da avaliação .....	107
5.3 Requisitos para pacotes de software .....	111
5.3.1 Requisitos de qualidade para pacotes .....	113
5.3.2 Requisitos para documentação de teste .....	116
5.3.3 Instruções para avaliação de conformidade .....	116
5.4 Outros requisitos de qualidade .....	117
5.4.1 Requisitos de usabilidade na interface .....	117
5.4.2 Requisitos para documentação de usuário .....	120
<b>Capítulo 6.....</b>	<b>125</b>
<b>Metodologia do processo de avaliação .....</b>	<b>125</b>
6.1 Estabelecer requisitos de avaliação .....	126
6.1.1 Estabelecer o propósito da avaliação .....	127
6.1.2 Identificar o tipo de produto a ser avaliado .....	128
6.1.3 Especificar o modelo de qualidade .....	129
6.2 Especificar a avaliação .....	130
6.2.1 Selecionar medidas .....	130
6.2.2 Estabelecer níveis de pontuação para as medidas .....	141
6.2.3 Estabelecer critérios para julgamento .....	144
6.3 Projetar a avaliação .....	146
6.3.1 Método de avaliação .....	146
6.3.2 Instruções ao avaliador .....	147
6.3.3 Recursos e cronograma .....	148
6.4 Executar a avaliação .....	148
6.4.1 Obter as medidas .....	148
6.4.2 Comparar com critérios .....	149
6.4.3 Julgar os resultados .....	149
6.5 Conclusão da avaliação .....	149
6.6 Processo de avaliação para componentes de software .....	151
6.6.1 Estabelecer os requisitos da avaliação .....	151
6.6.2 Especificação da avaliação .....	152
6.6.3 Projeto da Avaliação .....	153
6.6.4 Execução da Avaliação .....	153
6.6.5 Conclusão da Avaliação .....	153
<b>Capítulo 7.....</b>	<b>155</b>
<b>Avaliação de um software.....</b>	<b>155</b>
7.1 Apresentação do software .....	155
7.2 Processo de avaliação .....	155
7.2.1 Estabelecer requisitos de avaliação .....	155
7.2.2 Especificar a avaliação .....	158
7.2.3 Projetar a avaliação .....	159
7.2.4 Executar a avaliação .....	160
7.2.5 Conclusão da avaliação .....	161
<b>Considerações finais.....</b>	<b>163</b>

***Apêndice A***

**Guia de Avaliação da Qualidade de Produto de Software**

***Apêndice B***

**Modelo da identificação da avaliação**

***Apêndice C***

**Modelo de relatório de avaliação**

***Apêndice D***

**Conceitos de qualidade para produtos de software – uma síntese**

***Apêndice E***

**Relatório de Avaliação do Software: “Easy Learn”**

***Apêndice F***

**Glossário**

***Referências Bibliográficas***

## ÍNDICE DE FIGURAS

FIGURA 2.1 – ASPECTOS GERAIS DO SOFTWARE: A EVOLUÇÃO DOS SISTEMAS DE SOFTWARE. ....	20
FIGURA 2.2 – DEFINIÇÃO DE PROCESSO DE SOFTWARE. FONTE: CMMI (2002).....	28
FIGURA 2.3 – SITUAÇÃO DE MUITAS ORGANIZAÇÕES DE SOFTWARE. FONTE: MAGNANI (1998). ....	29
FIGURA 2.4 – RELAÇÃO DA NORMA NBR ISO/IEC 15504 COM OS MODELOS DE PROCESSO. ....	34
FIGURA 3.1 – ESTRUTURA DA LISTA DE VERIFICAÇÃO DE UM MÉTODO DE AVALIAÇÃO (MEDE-PROS®) .....	49
FIGURA 3.2 – ESTRUTURA A SER DESENVOLVIDA NUMA AVALIAÇÃO ESPECIALISTA.....	53
FIGURA 3.3 – QUALIDADE DE SOFTWARE DURANTE SEU DESENVOLVIMENTO (FONTE:NBR ISO/IEC 9126-1).....	57
FIGURA 3.4 – CICLO DE VIDA CLÁSSICO PARA DESENVOLVIMENTO DE SOFTWARE.....	59
FIGURA 4.1 – MODELO DE QUALIDADE DE MCCALL, ORGANIZADO EM FUNÇÃO DE TRÊS TIPOS DE CARACTERÍSTICAS DA QUALIDADE. ....	61
FIGURA 4.2 – MODELO BÁSICO DE QUALIDADE.....	66
FIGURA 4.3 – CARACTERÍSTICAS E SUBCARACTERÍSTICAS DE QUALIDADE. ....	67
FIGURA 4.4 – ARQUITETURA ATUAL DA SÉRIE ISO/IEC 25000.(2009) .....	71
FIGURA 4.5 – DIVISÃO DE GESTÃO DA QUALIDADE DA SÉRIE ISO/IEC 25000. ....	72
FIGURA 4.6 – DIVISÃO MODELO DE QUALIDADE DA SÉRIE ISO/IEC 25000.....	73
FIGURA 4.7 – DIVISÃO DE MEDIÇÃO DE QUALIDADE DA SÉRIE ISO/IEC 25000. ....	74
FIGURA 4.8 – DIVISÃO REQUISITOS DE QUALIDADE DA SÉRIE ISO/IEC 25000.....	74
FIGURA 4.9 – DIVISÃO DE AVALIAÇÃO DA QUALIDADE DA SÉRIE ISO/IEC 25000. ....	74
FIGURA 4.10– COMPONENTES DE SOFTWARE.....	79
FIGURA 4.11 – ETAPAS DE AVALIAÇÃO.....	80
FIGURA 4.12 – MODELO DE QUALIDADE NUMA VISÃO MAIS DETALHADA. ....	81
FIGURA 4.13 – CONJUNTO DE SISTEMAS APLICATIVOS DO PNAFM.....	82
FIGURA 4.14 – MODELO DE QUALIDADE DO PNAFM.....	83
FIGURA 4.15 – MODELO DE QUALIDADE PARA COMPONENTES DE SOFTWARE COTS.....	86
FIGURA 5.1 – RELACIONAMENTO ENTRE AS PARTES DA SÉRIE ISO/IEC 14598.....	100
FIGURA 5.2 – RELACIONAMENTO ENTRE AS SÉRIES 9126 E 14598. ....	101
FIGURA 5.3 – PROCESSO DE AVALIAÇÃO SEGUNDO NORMA NBR ISO/IEC 14598-1. ....	104
FIGURA 5.4 – PROCESSO DE AVALIAÇÃO SEGUNDO NORMA NBR ISO/IEC 14598-5. ....	105
FIGURA 5.5 – CONCEITOS NUM PROCESSO DE AVALIAÇÃO.....	108
FIGURA 5.6 – NÍVEIS DE PONTUAÇÃO PARA AS MEDIDAS FONTE: ISO/IEC 14598-1 (2002).....	109
FIGURA 5.7– VALOR MEDIDO E NÍVEL DE PONTUAÇÃO (NBR ISO/IEC 9126-1, 2001). ....	110
FIGURA 5.8 – ESTRUTURA DA NORMA NBR ISO/IEC 25051. ....	113
FIGURA 6.1 – PROCESSO DE AVALIAÇÃO SEGUNDO NORMA NBR ISO/IEC 14598-1.....	126
FIGURA 6.2 – ESTRUTURA DO MODELO DE QUALIDADE. ....	130
FIGURA 7.1 – CRITÉRIOS DE ACEITAÇÃO.....	159
FIGURA 7.2 – NOTAS OBTIDAS DO “EASY LEARN” .....	161

## ÍNDICE DE TABELA

TABELA 2.1 – PESQUISA DE UTILIZAÇÃO DE MÉTODOS DE ENGENHARIA DE SOFTWARE (MCT, 2001).....	24
TABELA 2.2 – NÍVEIS DE CAPACIDADE E ATRIBUTOS DE PROCESSO.....	33
TABELA 2.3 – CATEGORIAS E PROCESSOS DA SÉRIE NBR ISO/IEC 15504 (WEBER, 2001).....	34
TABELA 2.4 – CATEGORIAS E ÁREAS DE PROCESSO DO MODELO CMMI ...	37
TABELA 2.5 – COMPARAÇÃO ENTRE OS MODELOS PARA PROCESSO DE SOFTWARE .....	40
TABELA 3.1 – CARACTERÍSTICAS DE PRODUTOS DE SOFTWARE SEGUNDO A IEEE 1062 .....	44
TABELA 4.1 – CARACTERÍSTICAS DE QUALIDADE EM USO (NBR ISO/IEC 9126-1, 2001) .....	66
TABELA 4.2 – CARACTERÍSTICAS DE QUALIDADE INTERNA E EXTERNA (NBR ISO/IEC 9126-1, 2001) .....	68
TABELA 4.3 – SUBCARACTERÍSTICAS DE QUALIDADE DE SOFTWARE .....	68
TABELA 4.4 – NORMAS DA SÉRIE ISO/IEC 9126: “STATUS” EM 2009 .....	70
TABELA 4.5 – ETAPAS DE PROJETO E DOCUMENTOS ASSOCIADOS .....	75
TABELA 4.6 – SQUARE STATUS: 2009 .....	76
TABELA 5.1 – NORMAS DA SÉRIE ISO/IEC 14598: “STATUS” EM 2009. ....	100
TABELA 5.2 – REQUISITOS DE QUALIDADE PARA A DESCRIÇÃO DO PRODUTO. ....	114
TABELA 5.3 – REQUISITOS DE QUALIDADE PARA A DOCUMENTAÇÃO DO USUÁRIO. ....	115
TABELA 5.4 – REQUISITOS DE QUALIDADE PARA SOFTWARE .....	115
TABELA 5.5 – REQUISITOS DA DOCUMENTAÇÃO DE TESTE .....	116
TABELA 5.6 – INSTRUÇÕES PARA AVALIAÇÃO DE CONFORMIDADE .....	116
TABELA 5.7 – REQUISITOS BÁSICOS DE INCLUSÃO DE UM DOCUMENTO DE USUÁRIO DO SOFTWARE. ....	120
TABELA 5.8 – ITENS DE INFORMAÇÃO RECOMENDADOS PARA UM DOCUMENTO DE USUÁRIO .....	122
TABELA 5.9 – ITENS DE INFORMAÇÃO RECOMENDADOS PARA O CONTEÚDO DA INFORMAÇÃO DE CAPA.....	123
TABELA 6.1 – SUBCONJUNTO DO MODELO DE QUALIDADE .....	132
TABELA 6.2 – TIPO DE QUESTÃO 01 .....	142
TABELA 6.3 – TIPO DE QUESTÃO 02 .....	142
TABELA 6.4 – TIPO DE QUESTÃO 03 .....	143
TABELA 6.5 – TIPO DE QUESTÃO 04 .....	143
TABELA 6.6 – TIPO DE QUESTÃO 05 .....	143
TABELA C.1 – CONFIGURAÇÃO DO EQUIPAMENTO UTILIZADO NA AVALIAÇÃO.....	<b>ERRO! INDICADOR NÃO DEFINIDO.</b>





## Introdução

O conteúdo deste livro aborda conceitos, teorias e, principalmente, normas de qualidade de produto de software, já publicadas em âmbito nacional. Adicionalmente apresenta os resultados da experiência do desenvolvimento e da utilização de uma metodologia para avaliação da qualidade de produto de software. É direcionado a alunos, gestores, implementadores, avaliadores e outros interessados no tema qualidade de software.

As possibilidades abertas pelo computador, para a realização de serviços, mostram consumidores deslumbrados, mas sem o mínimo de conhecimento necessário para distinguir as mais elementares características da qualidade de produto na área de TI - Tecnologia da Informação. Assim, a questão da discussão da qualidade de software, num mercado composto de milhares de pequenos produtores, é o grande desafio, o qual vem sendo trabalhado por várias frentes, e as autoras fazem parte de uma dessas iniciativas. Com a intenção de popularizar os resultados, descrevem-se aqui os potenciais benefícios da atividade de avaliar produto de software:

- o desenvolvedor pode utilizar os resultados da avaliação de seu produto, para identificar ações corretivas, com o objetivo de melhorá-lo ou de tomar decisões sobre a estratégia de evolução do produto;
- o fornecedor de software pode obter confiança no valor do produto, além de lhe ser possível o uso do resultado da avaliação com finalidades comerciais;
- o adquirente de produto de software pode conhecer o nível de qualidade do software, para ajudá-lo na tomada de decisão para aquisição;
- nos negócios em geral, por meio da disseminação da qualidade de produto de software, pode-se ajudar no marketing, nas vendas e na satisfação do cliente.

O objetivo principal de uma avaliação de produto de software é fornecer resultados qualitativos e quantitativos sobre a qualidade desse produto, resultados que sejam compreensíveis, aceitáveis a quaisquer das partes interessadas, e confiáveis.

Em várias regiões do mundo, existem centros de desenvolvimento tecnológico em software, com o objetivo de acompanhar a área de Qualidade de Software, empenhados em repassar para organizações interessadas o estado da prática empregado com sucesso em projetos reais.

Alguns exemplos do que esses centros têm declarado, em suas respectivas missões, são citados a seguir:

- *Software Engineering Institute – SEI, localizado em Pittsburg, EUA, e mantido pelo DoD (US Department of Defense, ou Departamento de Defesa Americano). Faz parte de sua missão "assegurar o*

desenvolvimento e a operação dos sistemas com o custo, o cronograma e a qualidade melhores e previsíveis".

- *European Software Institute* – ESI, localizado na Espanha e mantido com verbas da Comunidade Européia e receita própria, cuja missão é "contribuir para o desenvolvimento da sociedade de informação e aumentar a competitividade da indústria, por meio do conhecimento, da inovação, da melhoria contínua e a promoção e a disseminação da tecnologia de informação".

No Brasil, desde 1993 de modo mais intenso, têm surgido várias iniciativas na área de qualidade de produto de software, tanto por parte do governo, como de centros de pesquisas, universidades, associações de classes e organismos normatizadores, com objetivos semelhantes aos já citados.

O setor de software é uma das prioridades governamentais na área de informática. Nesse setor, o governo brasileiro vem incentivando a criação de novas empresas e a capacitação das já existentes. Algumas das iniciativas são: o SOFTEX, com o Programa Nacional para a Excelência em Software, desde 2002, tem o objetivo de apoiar as empresas de software para que aumentem sua competitividade no mercado internacional e nacional (SOFTEX, 2009); o PBQP Software – Programa Brasileiro de Qualidade e Produtividade, com o objetivo de estimular a adoção de normas, métodos, técnicas e ferramentas da qualidade e da Engenharia de Software, promovendo a melhoria da qualidade dos processos, produtos e serviços de software brasileiro, de modo a tornar as empresas mais capacitadas a competir em um mercado globalizado. Especificamente, busca-se a melhoria contínua do grau de satisfação dos clientes, da qualidade de vida no trabalho e no país, e da lucratividade e competitividade das empresas brasileiras de software. Deve ser citada também a SEPIN – Secretaria de Política de Informática do Ministério da Ciência e Tecnologia (MCT), que tem conduzido periodicamente pesquisas sobre qualidade e produtividade entre as empresas de software no país.

O organismo normatizador brasileiro, Associação Brasileira de Normas Técnicas – ABNT, criou em 1993 o Subcomitê de Software, para elaborar normas nessa área, com o objetivo de gerar normas brasileiras relacionadas à Qualidade de Software.

De 1993 até o presente momento, várias normas sobre qualidade de produto de software foram publicadas no Brasil, acompanhando a evolução das normas internacionais. O conteúdo dessas normas será abordado neste livro.

Entidades de classes, como a Associação de Empresas de Software – ASSESPRO, com seu prêmio "Melhor Software do Ano" (ROCHA, 2001); programas de mestrado e doutorado em diversas Universidades Brasileiras e Institutos de Pesquisas, como o Centro de Tecnologia da Informação Renato Archer – CTI/MCT têm realizado pesquisas e aplicações práticas na área de qualidade de software, gerando teses e publicações com reconhecimento nacional e internacional.

As experiências adquiridas com as avaliações de produto de software e a metodologia desenvolvida no CTI, que aqui é apresentada, trazem conhecimento para novos pesquisadores na área de qualidade de produto de software darem continuidade ao assunto.

O MEDE-PROS® – Método de Avaliação de Qualidade de Produto de Software foi desenvolvido para avaliar a Qualidade de Produto de Software, tendo como referência as normas nacionais e internacionais. Esse método se encontra registrado na Fundação Biblioteca Nacional, sob número 135.620, livro 216, folha 84, e com o registro de marca no INPI sob número 820166243. Encontra-se na versão 2006 e é propriedade do CTI.

O MEDE-PROS® – é um exemplo de aplicação das normas, sendo que outras metodologias podem ser desenvolvidas para este contexto. Aqui será citado apenas esse método, como um exemplo para melhorar produtos de software, sob o ponto de vista do usuário final.

Este trabalho foi realizado numa fase de reestruturação das normas de qualidade de produto de software. Assim, as nomenclaturas que se encontram em ajustes nos órgãos normatizadores foram aqui tratadas apresentando, muitas vezes, os nomes e conceitos das normas anteriores. Procura-se minimizar problemas dessa transição, mencionando os nomes das normas reiteradamente.

Este livro está organizado da seguinte forma: No capítulo 1 é apresentado, rapidamente, um panorama tecnológico, sendo que a ideia de qualidade como um aporte básico e se estendendo para qualidade de produto e processo de software está no capítulo 2. As categorias de produtos de software e como tratá-las para uma avaliação estão no capítulo 3. Os modelos de qualidade se apresentam no capítulo 4. Outros requisitos de qualidade e o processo de avaliação sugeridos nas normas estão no capítulo 5. Uma sugestão de metodologia do processo de avaliação de software e um processo de avaliação específico para componentes de software estão apresentados no capítulo 6. Um exemplo de avaliação de software está no capítulo 7. As considerações finais fazem uma reflexão sobre os assuntos abordados. No apêndice A está o Guia de Avaliação da Qualidade de Produto de Software originado da metodologia do MEDE-PROS®, em sua formatação especial, que se apresenta de forma auto explicativa. No apêndice B se apresenta um modelo de formulário para identificação da avaliação. No apêndice C, uma sugestão de modelo de relatório de avaliação. O apêndice D apresenta uma síntese dos conceitos de qualidade para produtos de software, sendo útil para uma consulta rápida e familiarização desses conceitos. O apêndice E mostra o relatório de um produto avaliado. O apêndice F contém um glossário e finalmente, as referências bibliográficas.

## CAPÍTULO 1

### Panorama tecnológico

Nos últimos anos, tem ocorrido um crescimento sem precedentes na área de TI – Tecnologia da Informação. Com a globalização e a internet facilitando ainda mais a abertura de mercado, a produção de software é escoada independentemente das fronteiras nacionais.

A realidade mundial, nos últimos 20 anos, mostra que a utilização de sistemas computacionais comandados por software tem tido uma abrangência crescente, integrando-se no dia a dia de todos os setores de atividades da sociedade, atingindo um grande e diversificado número de usuários. No meio científico, vê-se ainda uma busca de como produzir software com qualidade. Em 1993, o projeto SCOPE – *Software CertificatiOn Programme in Europe* – foi exemplo de uma iniciativa internacional para melhorar a qualidade do produto de software, que gerou os primeiros trabalhos para elaboração de normas na área de Qualidade de Software (SCOPE, 2009). No meio industrial existe, por parte dos desenvolvedores e empresários de software, uma boa vontade em atender bem o mercado; mas, com a demanda cada vez mais crescente, o produto de software se apresenta sem os níveis de qualidade em conformidade com as normas, até os dias de hoje.

Qualidade de software constitui uma área cuja demanda está crescendo significativamente, pois os usuários exigem cada vez mais eficiência, eficácia, dentre outras características de qualidade importantes para um produto tão especial como o software. Paralelamente à demanda do mercado, existe um movimento nacional e internacional, no sentido de estabelecer normas na área de Engenharia de Software, como é o caso da ISO – Organização Internacional de Normalização, no Subcomitê de Engenharia de Software, e da ABNT – Associação Brasileira de Normas Técnicas.

Denomina-se ISO – *International Organization for Standardization* – a organização internacional para a padronização. Ela foi estabelecida em 1947, como uma organização mundial não governamental e conta atualmente com mais de 100 organizações nacionais de padronizações, representando mais de 130 países, responsáveis por mais de 95% da produção industrial mundial. Com sede em Genebra, Suíça, tem como principal atividade a elaboração de padrões para especificações e métodos de trabalho nas mais variadas áreas. O principal objetivo da ISO é o desenvolvimento de padrões mundiais, com vistas a facilitar o intercâmbio internacional de produtos e serviços e a criar uma cooperação intelectual, científica, econômica e técnica.

A IEC – *International Electrotechnical Commission*, fundada em 1906, é a organização mundial que publica normas internacionais relacionadas com eletricidade, eletrônica e áreas afins, tendo a participação de mais de 50 países.

A ISO, em conjunto com a IEC, elaborou um conjunto de normas que tratam, especificamente, da atual padronização mundial para a qualidade dos produtos de software.

No Brasil, a ABNT foi fundada em 1940. Reconhecida como Foro Nacional de Normalização, é o único órgão responsável pela normalização técnica no país, fornecendo a base necessária ao desenvolvimento tecnológico brasileiro. É uma entidade privada, sem fins lucrativos, e representa o Brasil nas entidades de normalização internacional, como a ISO e a IEC.

As normas recomendam modelos de qualidade e processos de avaliação da qualidade de software que se têm firmado como valioso auxílio à obtenção de produtos de software com qualidade aprimorada e mais confiável. Por um lado, surge a oportunidade do desenvolvimento de metodologias para avaliar a qualidade de produto de software, com base em atributos de qualidade das normas reconhecidas internacionalmente, dando subsídios para que compradores e usuários adquiram produtos com características mais próximas de suas necessidades. Por outro, surge para produtores de software a possibilidade de fornecer produtos de melhor qualidade.

A avaliação de produto de software tem sido uma das formas empregadas por organizações que produzem ou adquirem software para obtenção de melhor qualidade nesses produtos, sejam eles completos ou partes a serem integradas a um sistema computacional mais amplo. Para que a avaliação seja mais efetiva, é importante utilizar um modelo de qualidade que permita estabelecer e avaliar requisitos de qualidade. É necessário que o processo de avaliação seja bem definido e estruturado. Os conjuntos de normas ISO/IEC JTC1/SC7 9126 e ISO/IEC 14598, atualmente sendo reestruturadas na série ISO/IEC 25000 - *Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE)*, descrevem um modelo de qualidade, um processo de avaliação, uma diretriz para requisitos de qualidade e um conjunto de medidas de qualidade. Alguns exemplos de medidas são encontrados nessas normas e podem ser utilizados por organizações que pretendam fazer avaliação de produto de software. A norma NBR ISO/IEC 25051 (2008) estabelece requisitos de qualidade para pacotes de software e instruções para avaliação da conformidade de software. Pacotes de software, denominados COTS – *Commercial Off-The-Shelf* –, são produtos de software definidos por uma necessidade orientada pelo mercado, disponíveis comercialmente, cuja adequação para uso foi demonstrada por um grande número de usuários.

As organizações produtoras de software devem procurar soluções compatíveis com seu porte e objetivos de negócio. Um exemplo disso pode ser visto em Pivka (1995), que propõe a certificação do produto de software baseada na norma ISO/IEC 12119<sup>1</sup> como alternativa para a garantia da qualidade de software produzido por pequenas empresas produtoras de software. Esse tipo de certificação, não concretizado ainda, é tema de estudo e esforços individuais.

---

<sup>1</sup> O autor Pivka, na época, referia-se à norma ISO/IEC 12119

Deve-se, ainda, considerar que, diante de um mercado consumidor imaturo, a pessoa ou entidade que se coloca no papel de avaliador deve assumir uma responsabilidade adicional: primeiro identificar os desejos já expressos por esse mercado; em seguida, antecipar outras necessidades importantes ainda não definidas e, finalmente, influir no mercado, para que tanto o produtor como o consumidor possam evoluir.

As organizações nacionais de software devem repensar as premissas fundamentais que norteiam uma disputa de mercado e enfrentar o grande desafio de oferecer preços competitivos, menores prazos de desenvolvimento, menos defeitos e maior satisfação dos clientes e usuários. Esses são alguns dos requisitos de mercado para um software de qualidade.

As organizações se têm deparado com projetos de software cada vez maiores, mais complexos e de grande impacto na sociedade. O software faz parte do dia a dia de toda a sociedade. Ele transfere fundos entre instituições financeiras, pilota aviões, controla equipamentos em centros médicos, diverte as crianças, torna possível pesquisa científica de grande complexidade aritmética e muito mais. O grande problema é que, em geral, a qualidade do software não é satisfatória, por não atender as necessidades dos usuários e apresentar falhas.

A comunidade científica, o governo e as próprias organizações, em resposta a essas necessidades, cadê vez mais estão buscando a qualidade de software, por meio de abordagens voltadas tanto para o produto final quanto para o processo de desenvolvimento e manutenção de software.

Estudos sobre a qualidade, no setor de software brasileiro, mostraram a necessidade de um esforço significativo capaz de aumentar a maturidade dos processos de software das empresas brasileiras e deram início ao Programa MPS.Br – Melhoria de Processo do Software Brasileiro. É uma iniciativa que envolve universidades, grupos de pesquisa e empresas, sob a coordenação da Sociedade Softex (Associação para Promoção da Excelência do Software Brasileiro). O programa tem como premissa a definição e disseminação de um Modelo de Referência e um Modelo de Negócio para melhoria de processo de software MR-MPS e MN-MPS, respectivamente.

A qualidade de software continua, no entanto, requerendo melhorias. Um processo de qualidade não garante a produção de um produto de software de qualidade. Percebe-se, nesse ponto, uma lacuna nos esforços que vêm sendo realizados na busca pela qualidade de software. O processo, que irá resultar no produto de software, concentra seus esforços na busca pela qualidade do “modo de produção e manutenção” do software, ao passo que a qualidade do produto de software é focada com mais intensidade apenas quando ele já está pronto, por meio da avaliação de sua performance.

As iniciativas pela busca da qualidade de software descritas anteriormente, denominadas, respectivamente, abordagem de processo e abordagem de produto, são de grande valor e tratam seus objetivos de forma exemplar, mas agem de forma isolada. Há uma expectativa de que o programa MPS.BR da Softex ([www.softex.br](http://www.softex.br)), que teve início em 2003, no Brasil, tenha outro resultado e consolide a ideia de que um processo bem definido ajuda muito a qualidade do produto.

A qualidade do produto de software precisa fazer parte, de maneira intensa e formal, das preocupações do processo de desenvolvimento e manutenção. As características de qualidade de um produto de software precisam ser alocadas e verificadas também nos produtos de software intermediários, ao longo do processo, e não apenas no produto já acabado. Tal procedimento permite que desvios no produto sejam detectados durante seu processo de desenvolvimento e manutenção, originando ações necessárias ao processo para direcioná-lo à produção de um produto de software de qualidade.

O software é um produto complexo, que depende em parte da interpretação das necessidades do usuário que irão se converter nos requisitos do produto para o seu desenvolvimento; dessa forma, necessita de técnicas verificação e validação durante o ciclo de desenvolvimento do produto e técnicas de avaliação do produto intermediário e final que garantam que o produto é correto e seguro, entre outros atributos de qualidade.



## CAPÍTULO 2

### Qualidade de Software

A Engenharia de software tem uma de suas áreas, a Qualidade de software, que objetiva garantir que especificações explícitas e necessidades implícitas estejam presentes no produto, por meio da definição e normatização de processos de desenvolvimento. Apesar de os modelos aplicados na garantia da qualidade de software atuarem, em especial, no processo, o principal objetivo é garantir um produto final que satisfaça às expectativas do cliente, dentro daquilo que foi acordado inicialmente.

#### 2.1 Qualidade de produto

Existem várias propostas de definição para qualidade de produto. Crosby (1979) afirma: “Qualidade é conformidade com os requisitos”. Essa definição foi utilizada na manufatura, porém serve para engenharia de software. Além disso, menciona que é preciso definir qualidade dessa forma, principalmente porque a qualidade deve ser gerenciada. Para entender o significado de qualidade em termos práticos, é necessário conhecer os cinco principais erros, “En”, cometidos por gerentes e saber como eles devem ser tratados, “Sn”, de acordo com esse autor:

- |    |   |
|----|---|
| E1 | Qualidade significa ‘ótimo’, ou ‘luxo’, ou ‘brilhante’, ou ‘de grande valor’.   |
| S1 | A palavra ‘qualidade’, muitas vezes, é usada em expressões do tipo: ‘boa qualidade’, ‘má qualidade’ e até ‘qualidade de vida’. Mas, cada um que a ouve atribui um significado para o que seja ‘qualidade de vida’; por exemplo, um sentido que não corresponde ao que um falante deseje, realmente, dizer com a referida expressão. É preciso definir ‘qualidade de vida’ em termos específicos, tais como: renda familiar; saúde; controle de poluição; programas políticos e quaisquer outros itens que possam ser medidos. |
| E2 | Qualidade é intangível, portanto não mensurável.  |
| S2 | Na verdade, qualidade é precisamente mensurável por meio da mais antiga e respeitada medida, o dinheiro. Ignorar esse fato tem levado gerentes a perder muito dinheiro. A qualidade é medida pelo ‘custo da qualidade’, que é a despesa, ou custo da ‘não conformidade’, que é o custo de fazer coisas erradas.   |
| E3 | As desculpas dos gerentes para não fazer nada, em relação à qualidade de seus produtos, é que ‘seu negócio é diferente’ e que a ‘ciência’ da qualidade não os ajudaria a fazer o que já fazem, e de forma ainda melhor.   |
| S3 | Eles ainda não compreenderam o real significado de ‘qualidade’ e continuam acreditando que ela significa ‘luxo’. Nesses casos, é importante explicar o real significado de ‘qualidade’ e que é sempre mais barato fazer certo na primeira vez.  |

- E4 Os problemas de qualidade são originados por trabalhadores, principalmente aqueles que trabalham diretamente na área de produção.
- S4 Os funcionários da produção de uma fábrica são acusados de provocar os problemas. Na realidade eles pouco contribuem para a prevenção, ou não, de defeitos, pois todo planejamento e criação foram definidos previamente, e eles são apenas seus executores.
- E5 Qualidade é responsabilidade do departamento da qualidade.
- S5 O departamento da qualidade tem como atribuição: medir a conformidade de acordo com o que foi previamente determinado; reportar os resultados das medidas de forma clara e objetiva; liderar uma atitude positiva da empresa, na busca da melhoria da qualidade; prover e capacitar os funcionários com ferramentas que podem auxiliar na melhoria da qualidade. Porém, o departamento da qualidade 'não deve' executar o trabalho, pois, caso o faça, a empresa nunca mudará sua conduta.

Resumidamente, “qualidade” é conformidade com requisitos, e estes devem estar definidos para permitir que sejam gerenciados com o uso de medidas, de forma a reduzir o retrabalho e aumentar a produtividade. A melhoria da qualidade deve estar focada nos processos, e não nas pessoas; certamente, é responsabilidade de todos os envolvidos no processo.

A satisfação com o produto está relacionada com seu desempenho e com a ausência de defeitos, erros ou falhas. Portanto, a satisfação com o produto é alcançada quando as necessidades do cliente são supridas, e o produto se comporta como é esperado. Os requisitos representam as necessidades explícitas dos clientes e devem procurar cobrir a maior parte das necessidades por eles declaradas em relação ao produto.

Sobre as necessidades do cliente, algumas não estão na especificação dos requisitos, mas também são de fundamental importância. Trata-se das necessidades implícitas, definidas por Joseph Juran (1988), e que serão explicadas a seguir.

Necessidades explícitas nada mais são do que a definição dos requisitos, as condições e os objetivos propostos para o produto. Quanto às necessidades implícitas, têm foco na visão subjetiva do consumidor/usuário com relação ao produto, tais como: as necessidades razoáveis; implicações estéticas; itens de segurança, entre outras. As necessidades explícitas para um carro popular podem ser, por exemplo: ser econômico, fazendo 20 km com 1 litro de combustível, e custar no máximo R\$ 25 mil. Como necessidades implícitas poderiam constar: ser confortável; ter um design sofisticado; possuir itens de segurança e seu motor funcionar.

Dessa forma, o melhor a fazer é definir muito bem os requisitos de um produto e gerenciar seu processo, para que as necessidades implícitas e os defeitos possam ser reduzidos ao mínimo. Tudo isso sem deixar de monitorar a satisfação do cliente com o produto, o que irá orientar os esforços na busca pela qualidade.

Nessa busca pela qualidade, as organizações procuram basear-se em conceitos e diretrizes reconhecidos internacionalmente como normas, que são elaboradas e revisadas por órgãos responsáveis por normalizações técnicas.

A norma NBR ISO 9000<sup>2</sup> sobre terminologias de gestão e garantia da qualidade define qualidade como: “A totalidade das características de uma entidade que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas”.

Essa definição é abrangente o bastante para incorporar toda a visão de qualidade, basicamente relacionada a produtos manufaturados, transmitida por Crosby (1979) e Juran (1988), e pode também orientar as atividades para obtenção e avaliação da qualidade de software. O termo “totalidade das características” indica que tanto aspectos funcionais quanto os não funcionais devem ser considerados; o termo “satisfazer necessidades implícitas e explícitas”, por sua vez, ressalta a importância da conformidade, tanto com os requisitos – necessidades explícitas – como com a satisfação daquelas necessidades que não necessariamente estão descritas nos requisitos – as necessidades implícitas, mencionadas anteriormente.

Na próxima seção, o software será abordado quanto aos aspectos gerais e suas características como produto, além de serem apresentadas as iniciativas e abordagens na busca por sua qualidade.

## **2.2 Qualidade de produto de software**

Pode-se definir qualidade de produto de software como a conformidade a requisitos funcionais e de desempenho declarados explicitamente, padrões de desenvolvimento claramente documentados e as características implícitas que são esperadas de todo software desenvolvido profissionalmente. Por necessidades explícitas, pode-se entender requisitos do usuário; necessidades implícitas relacionam-se, por exemplo, com a performance de execução do sistema ou até mesmo com o cumprimento do cronograma e o orçamento do desenvolvimento do produto. As definições de qualidade podem variar em alguns aspectos, porém o aspecto que se refere à satisfação do cliente ou usuário não deve ser esquecido.

A ISO, que objetiva promover o desenvolvimento de padronização, no sentido de facilitar o intercâmbio de bens e serviços entre os povos, foi criada em razão da necessidade das empresas de exportarem seus produtos, por causa da chamada barreira técnica. Os padrões da ISO estabelecem especificações técnicas e definem características para garantir que produtos, serviços ou processos sejam adequados a seus propósitos. A vantagem maior fica para o cliente que, ao adquirir um produto ou contratar um serviço, tem a garantia e também a segurança de que o produto que está adquirindo tem padrões internacionais conhecidos e familiares no mundo inteiro. A norma NBR ISO/IEC 9126-1, futura

---

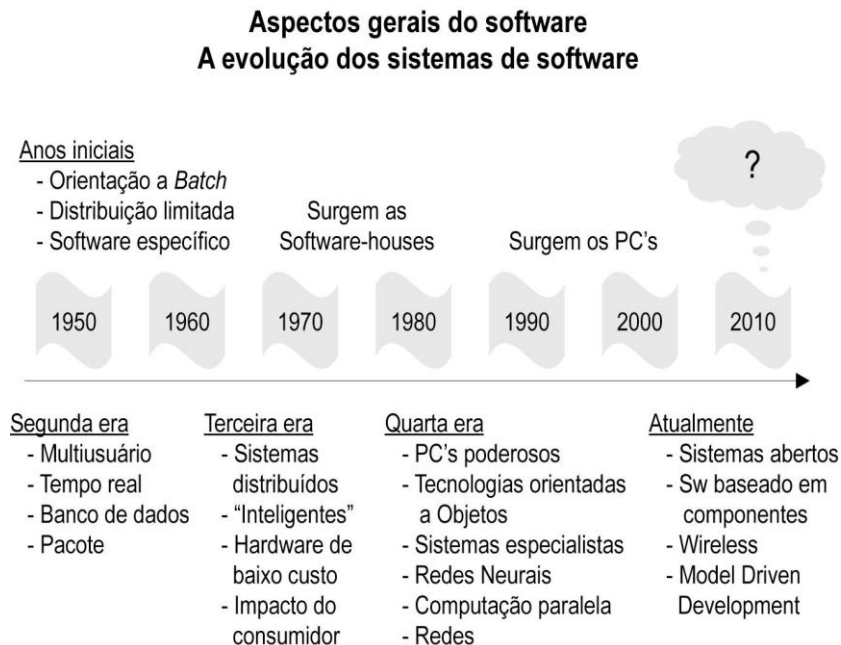
<sup>2</sup>A família de normas NBR ISO 9000:1994 (9001, 9002 e 9003) foi cancelada e substituída pela série de normas ABNT NBR ISO 9000:2000.

ISO/IEC 25010 – *Quality Model*, faz parte desse conjunto de normas ISO. Essa norma propõe características que um software deve possuir, bem como subcaracterísticas, para incentivar o uso na prática dessa padronização de qualidade de produto de software.

## 2.2.1 Aspectos gerais e evolução do software

Os produtos de software estão presentes nas mais diversas áreas de atuação da vida cotidiana das pessoas. Algumas vezes, simplesmente são usados sem grande ênfase; outras vezes, são a ferramenta de trabalho; e ainda, em outras, vidas são colocadas sob sua influência. Algumas dessas áreas são: educação, entretenimento, transporte, comunicação, sistema financeiro, meio ambiente, indústria, comércio, medicina, pesquisa e muitas outras de igual ou maior importância.

Um pouco da história do desenvolvimento de software relata os problemas e a busca pela qualidade dos produtos. Os sistemas de software tiveram uma evolução muito densa e rápida. Os primeiros sistemas surgiram na década de 1950, e desde então existe uma evolução contínua. Pode-se classificá-los em quatro eras, como mostra a Figura 2.1.



**Figura 2.1 – Aspectos gerais do software: a evolução dos sistemas de software.**

Inicialmente, nos anos 1950, a maioria dos produtos de software eram específicos, com distribuição limitada e orientados a *batch*, ou seja, eram executados em lotes. As exceções a esse quadro eram o sistema interativo da *American Airlines* e o sistema em tempo real do Departamento de Defesa Americano - DoD, em função da necessidade que esses sistemas tinham de prover informações para possibilitar a tomada de decisão no exato momento em que determinadas situações ocorriam.

Nos anos 1960 surgiu a demanda por novas aplicações e com níveis de sofisticação maiores. Foi nesse período que apareceram os produtos de software multiusuários, o processamento em tempo real, o banco de dados e os pacotes de software, que são software que atendem a uma gama de usuários com necessidades semelhantes. Com toda essa demanda, era natural o surgimento de empresas de desenvolvimento de software, chamadas *software-houses*.

Problemas começaram a partir desse momento. Quando era necessária uma adaptação em um software que fora comprado ou quando era detectada uma falha, muitas linhas de código precisavam ser introduzidas, revisadas, corrigidas e testadas. O tempo envolvido nessas atividades começou a aumentar de forma descontrolada, e sua eficácia não podia ser garantida.

Todos esses fatos provocaram a chamada Crise do software e, para combatê-la, foi intensificado o desenvolvimento de métodos, ferramentas e procedimentos destinados ao desenvolvimento de software e integrados na disciplina Engenharia de software.

A Engenharia de software, segundo Pressman (2002), "...é uma disciplina que pode ser vista, de forma objetiva, como o estabelecimento e o uso dos princípios básicos da engenharia, com a finalidade de desenvolver software de maneira sistemática e econômica, resultando em um produto confiável e eficiente." Ou seja, ela é uma disciplina que auxilia na melhoria da qualidade de software.

Nos anos 1970, os computadores pessoais, chamados PC, tornaram-se, com seus preços acessíveis, uma grande comodidade aos usuários de informática, que começaram a comprar software em grande escala. Iniciou-se, também nessa década, o desenvolvimento de software para serem acoplados em automóveis, aparelhos de microondas e outros equipamentos.

Em meados dos anos 1980, novas tecnologias tomaram o lugar das convencionais. Nesse período, surgiu a inteligência artificial, com os sistemas especialistas saindo dos laboratórios; despontou a tecnologia de programação Orientada a objetos, que parecia ser um caminho para a solução dos problemas de manutenção. Nessa década, surgiu a primeira grande necessidade identificada pelas exigências do DoD. Uma das suas maiores dificuldades era adquirir software junto aos fornecedores, visto que sucessivos problemas de erros, prazo de entrega e custos variavam muito entre si. Como eram grandes projetos, com muitos recursos investidos, arriscar com um ou outro fornecedor poderia colocar em risco vários objetivos. Para suprir essas necessidades, era preciso qualificar, de alguma maneira, o potencial dos fornecedores quanto aos produtos fornecidos.

A partir dessa demanda, foi criado, em 1984, o SEI – *Software Engineering Institute*, um centro de pesquisa e desenvolvimento, mantido com verbas do governo americano, instalado na Universidade de Carnegie Mellon, com a missão de prover liderança para o avanço do estado da prática da Engenharia de software, melhorando a qualidade dos sistemas que dependem de software.

Apesar de todos os esforços, muitos problemas continuam ocorrendo. Isso acontece principalmente por causa de alguns aspectos bastante conhecidos pelas organizações:

- a habilidade de construir software não acompanha o potencial do desenvolvimento de novo hardware, que é cada vez mais rápido e elevado;
- a capacidade de manter os projetos de software existentes está ameaçada por projetos malplanejados e por recursos inadequados; e
- os projetos de software estão ficando cada vez mais complexos.

Com isso, a quantidade de defeitos e a insatisfação dos usuários são crescentes. A visibilidade desses fatos fica por conta dos acidentes aéreos em que não se consegue a comprovação de falha humana como causa; por conta do controle, ou falta dele, sobre plantas petrolíferas e químicas; da complexidade das redes elétricas e de telecomunicações, além de inúmeros outros exemplos que poderiam ser citados.

A sofisticação do hardware ultrapassou a habilidade de construir software que explore o total potencial do hardware; a habilidade de manter os projetos de software existentes foi ameaçada por projetos pobres e por recursos inadequados; a quantidade de defeitos e a insatisfação dos usuários tornaram-se visíveis, e a possibilidade de desastres aumenta cada vez mais. Duas ocorrências foram muito comentadas pela mídia: o software dos sistemas computacionais utilizados na Olimpíada de Atlanta, EUA, em 1996, de responsabilidade da IBM, teve de ser retirado de operação, sendo que a IBM reconheceu a falha; outro fato foi que o software implantado no aeroporto de Denver, EUA, para manipulação de bagagens com mais de 100 computadores interligados, atrasou a inauguração do aeroporto em mais de sete meses.

Além dessa evolução rápida e desses problemas, o produto de software é diferente de outros tipos de produtos, como os manufaturados. Algumas características de um produto de software serão apresentadas no item 2.2.2 e mostram que grande parte das dificuldades para obter um produto de qualidade decorre da necessidade de que essas características sejam satisfatoriamente atendidas.

A importância de sistemas de software no mundo moderno não pode ser subestimada. Setores econômicos, como manufatura, finanças, comunicações, saúde, energia, transportes, educação e gestão pública dependem do uso de software para a condução de suas operações diárias, utilizando desde aplicações em computadores pessoais até grandes sistemas em redes de complexidade gigantesca. Não é exagero dizer que o software se tornou um recurso mundial, vital para o bem-estar e a competitividade.

A importância e os problemas relacionados a software deixam clara a relevância da busca pela melhoria de sua qualidade. Na próxima seção, o software será analisado em suas características particulares e diferenças em relação ao produto manufaturado.

### 2.2.2 Software e suas características como produto

Existem diferenças importantes entre produtos de software e produtos manufaturados que não podem deixar de ser notadas. As características inerentes à essência do software e diferenças em relação aos produtos manufaturados são (CAPOVILLA, 1999):

- Complexidade: normalmente, um produto de software tem muitas regras a serem cumpridas; muitas linhas de código a serem implementadas; e, frequentemente, diversos desenvolvedores envolvidos, que não só têm ideias diferentes, e algumas vezes divergentes, mas que podem levar à mesma solução;
- Invisibilidade e intangibilidade: o software é invisível para o usuário. O que se vê são as consequências da execução do software, diferentemente de um produto manufaturado. Os próprios desenvolvedores necessitam utilizar modelos para representar o sistema de software. Essa intangibilidade causa grandes dificuldades de comunicação, tanto entre os elementos da equipe de desenvolvimento como entre a equipe e o cliente, podendo acarretar problemas no produto de software;
- Conformidade e modificabilidade: o software é a interface entre diversas entidades do meio no qual será utilizado: equipamentos, outros produtos de software, usuários e cultura organizacional, entre outras. Sendo o componente mais maleável e adaptável do sistema, frequentes adaptações são realizadas no software, para adequá-lo a essas entidades;
- Produção sob medida: para software não existe produção em série, cada usuário é um cliente, que usa o software à sua maneira, com ênfase em partes diferentes;
- Não se desgasta com o uso: em software os componentes lógicos são duráveis. A falha de software resulta de erros de projeto ou de implementação, e os defeitos permanecem no sistema até serem percebidos devido à ocorrência de um erro quando uma determinada entrada acontece. Os defeitos de projeto e “fabricação” provocam um grande número de falhas logo no início, mas depois se comportam de maneira estável até sua obsolescência. O não desgaste diferencia o software da quase totalidade dos produtos modernos. Apenas a música e o cinema, por exemplo, aproximam-se do software sob esse aspecto.
- Não tem prazo de validade: o software não é sensível a problemas ambientais e nem sofre qualquer tipo de defeito devido a efeito cumulativo de seu uso. O software se torna obsoleto devido à evolução do hardware e, conseqüentemente, da tecnologia. Empresas produtoras de software proprietário, com uma base grande de usuários costumam planejar a obsolescência de seus produtos, para aumentarem as suas vendas.
- O custo final do software é basicamente o custo do projeto e do desenvolvimento: cópias do software podem ser reproduzidas em segundos e distribuídas a vários clientes, com o custo unitário do projeto e do desenvolvimento.

- Software é o único produto que, quando apresenta erro, o cliente paga para corrigir: no caso de uma nova versão licenciada, esta pode ter apenas os erros corrigidos.

Diante dessas diferenças, é difícil imaginar, de forma direta no desenvolvimento de software, o aproveitamento de toda a experiência e maturidade existentes no processo de fabricação de um produto manufaturado. A Engenharia de software, no entanto, foi criada com o objetivo de estabelecer o uso de princípios básicos da engenharia clássica, ou seja, tornar um produto invisível, intangível e complexo em um produto confiável e eficiente. As organizações de software que se preocupam com a qualidade vêm utilizando a Engenharia de software.

A Secretaria de Política de Informática do Ministério da Ciência e Tecnologia – SEPIN vem conduzindo periodicamente pesquisas diretas com organizações de software, visando acompanhar a evolução da gestão da qualidade nesse setor. Conforme a Tabela 2.1, reproduzida com base nos dados das três últimas edições desse tipo de pesquisa, é possível constatar a utilização de métodos de Engenharia de software, o que indica a credibilidade em seu potencial. Observe-se, também, que já em 2001 os dados eram favoráveis à adoção dessas metodologias. Nessa tabela pode-se observar a variabilidade da utilização de métodos de Engenharia de software.

Tabela 2.1 – Pesquisa de utilização de métodos de Engenharia de Software (MCT, 2001)

Métodos de Engenharia de Software	Resultados (Nº de Empresas / %)		
	1995 (445 empresas)	1997 (589 empresas)	1999 (426 empresas)
Empresas que adotam métodos de prevenção de defeitos			
Auditorias	...	102 / 17%	88 / 21%
Gerência de configuração	...	40 / 7%	63 / 15%
<i>Joint Application Design</i>	...	46 / 8%	36 / 9%
Medições da qualidade (medidas)	45 / 10%	48 / 8%	52 / 12%
Prototipação	207 / 46%	259 / 44%	187 / 44%
Reúso	166 / 37%	110 / 19%	104 / 24%
Verificação independente	...	81 / 14%	155 / 36%
Empresas que adotam métodos de detecção/remoção de defeitos			
Inspeções formais	45 / 10%	100 / 17%	87 / 20%
Revisões estruturadas	...	113 / 19%	66 / 16%
Testes de aceitação	212 / 48%	278 / 47%	205 / 48%
Testes de sistema	277 / 62%	392 / 67%	199 / 47%
Testes de unidade	105 / 24%	137 / 23%	130 / 31%
Validação	...	250 / 42%	192 / 45%



Empresas que adotam outras práticas de Engenharia de Software			
Gestão de mudança	...	32 / 5%	31 / 7%
Métodos estruturados	...	210 / 36%	146 / 34%
Métodos orientados a objetos	193 / 43%	216 / 37%	186 / 44%
Projetos interface com o usuário	...	207 / 35%	215 / 51%

**Observação:** O símbolo “...” indica dado indisponível.

Infelizmente, a realidade das empresas, tanto as mundiais como as nacionais, está distante do ideal. A realidade das empresas de software do país, sendo 36% do tipo microempresa e 33% do tipo pequena empresa, segundo a força de trabalho, com equipe e faturamento reduzidos, explica uma situação e indica a necessidade que as empresas têm de receber auxílio externo para superar seus altos índices de desconhecimento e capacidade de aplicação dessas normas e modelos.

Apesar de todos os esforços da Engenharia de software, os problemas de qualidade nos produtos persistem. Na próxima seção serão analisadas outras iniciativas que vêm sendo realizadas na busca pela qualidade do produto software.

### 2.2.3 Iniciativas para a qualidade de produto de software

Diante do que foi apresentado nas seções anteriores, o produto de software, há algum tempo, necessita e busca soluções para a melhoria na sua qualidade – tanto em função de seu grau de importância e integração na sociedade quanto pelas falhas frequentes e com possibilidade de consequências desastrosas. Tudo isso sem deixar de mencionar, também, o alto custo e elevado tempo de desenvolvimento e manutenção.

Qualidade de software pode ser vista, e é definida nas normas relacionadas a essa área, como um conjunto de características que devem ser alcançadas em um determinado grau, para que o produto atenda às necessidades de seu usuário, cuja participação é fundamental no processo. É por meio desse conjunto de características que a qualidade de um produto de software pode ser descrita e avaliada.

Analisando essa afirmação, percebe-se a necessidade de respostas para algumas questões básicas sobre como atingir a qualidade num produto de software a ser desenvolvido:

- A determinação do conjunto de características que atende às necessidades de seus usuários.
- A forma de avaliar se tais características foram alcançadas num grau que satisfaça seus usuários.

Diversas iniciativas surgiram para buscar as respostas a essas questões. De acordo com o livro *Qualidade de Software – Teoria e Prática* (ROCHA, 2001), diversos modelos foram desenvolvidos, detalhando as características de qualidade de um produto de software em subcaracterísticas, e estas em

atributos. Alguns desses trabalhos são encontrados nos livros: *Factors in software quality* (1977) de McCALL, *Characteristics of software quality* (1978) de B. Boehm e *Cornering the chimera* (1996) de R. G. Dromey.

O Subcomitê de Software – SC7 do Comitê Técnico Conjunto – JTC1 da ISO e IEC vêm trabalhando desde a década de 1990, na elaboração de normas e relatórios técnicos que permitam especificar e avaliar a qualidade de produtos de software, consolidando as diferentes visões de qualidade. Para auxiliar no processo de avaliação da qualidade de produtos de software, a ISO e a IEC estabeleceram o seguinte conjunto de normas na série 14598<sup>3</sup>, que já foram publicadas pela ABNT, com os seguintes assuntos:

- Visão geral do processo de avaliação;
- Processo de avaliação para desenvolvedores;
- Processo para adquirentes;
- Processo para avaliadores;
- Planejamento e gestão do processo;
- Documentação de módulos de avaliação.

A norma brasileira NBR ISO/IEC 9126-1 é uma tradução da norma ISO/IEC 9126-1, contendo modelo de qualidade com as características e subcaracterísticas de um produto de software.

Atualmente, a série de normas ISO/IEC 25000, que trata da definição da qualidade de produto de software, é composta por um conjunto de documentos contendo abordagens sobre o modelo de qualidade, medidas externas, medidas internas e medidas da qualidade em uso. O assunto será tratado com detalhes no capítulo 4.

Para se obter um produto de software de qualidade, é preciso verificar seu processo de desenvolvimento e também as características do produto em produção. Na próxima seção, serão abordados aspectos gerais de um processo de software, iniciativas e modelos para sua qualidade.

Weber cita, em *Qualidade e Produtividade em Software*, publicado em 2001, que a base para o reconhecimento da qualidade do software brasileiro e a chave para alcançar crescentes níveis de competitividade nas empresas brasileiras de software encontram-se:

- na adoção da norma ISO 9001 e do CMMI como modelos complementares para a melhoria dos processos de software;

---

<sup>3</sup> Atualmente em revisão na série 25000.

- na adoção da série ISO/IEC 9126, série ISO/IEC 14598 e ISO/IEC 12119<sup>4</sup> para avaliação dos produtos de software.

Essas abordagens podem ser adotadas separadamente ou em conjunto, como foi tratado, por exemplo, na proposta para qualidade de software com a aplicação integrada de modelo de capacidade de processo e norma de produto, em Sant'ana (2002). Procurou-se analisar atividades, metas e produtos de trabalho das áreas de processo, verificando instâncias em que as definições, diretrizes e características de qualidade de produto de software, possam ser utilizadas. Também essas abordagens podem ser escolhidas em conformidade com os objetivos de negócio das empresas e da capacitação de seu pessoal.

### **2.3 Qualidade de processo de software**

O motivador deste assunto é a busca pela qualidade de software, ou seja, qualidade tanto do produto como do processo de software. A ideia atual de que se deve atuar na qualidade do processo, para se atingir a qualidade do produto de software, não pode ser vista isoladamente, mas em composição com outras abordagens.

Processo de software está definido como um conjunto de atividades, métodos, práticas e transformações que as pessoas empregam para desenvolver e manter software e os produtos associados. Especificação de requisitos, gerência de configuração, desenvolvimento de software são exemplos de processos que podem ser formalizados e documentados para desenvolver um produto. A Figura 2.2 representa a definição de processo de software.

---

<sup>4</sup> O autor, na época se referia à norma ISO/IEC 12119

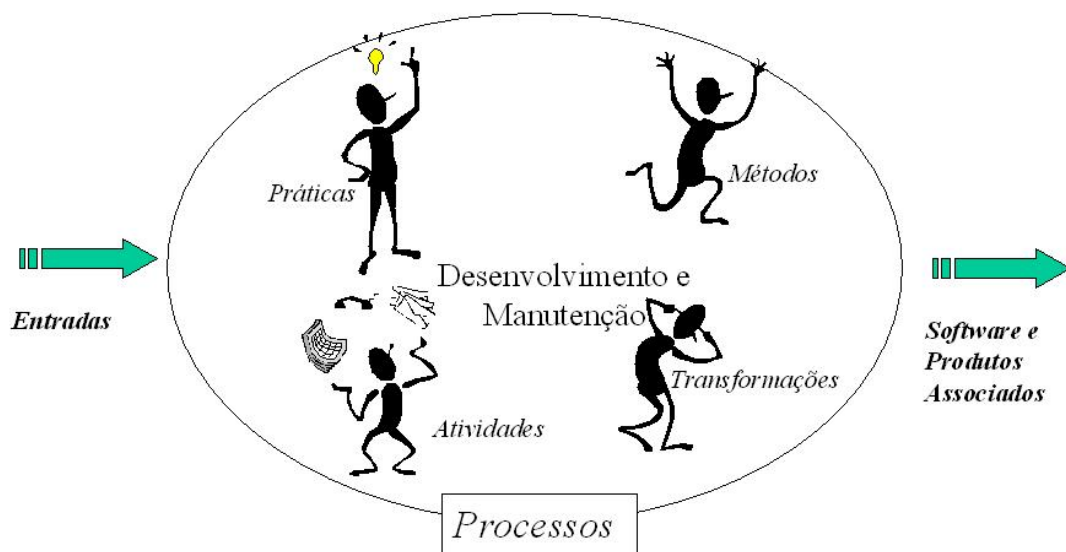


Figura 2.2 – Definição de processo de software. Fonte: CMMI (2002).

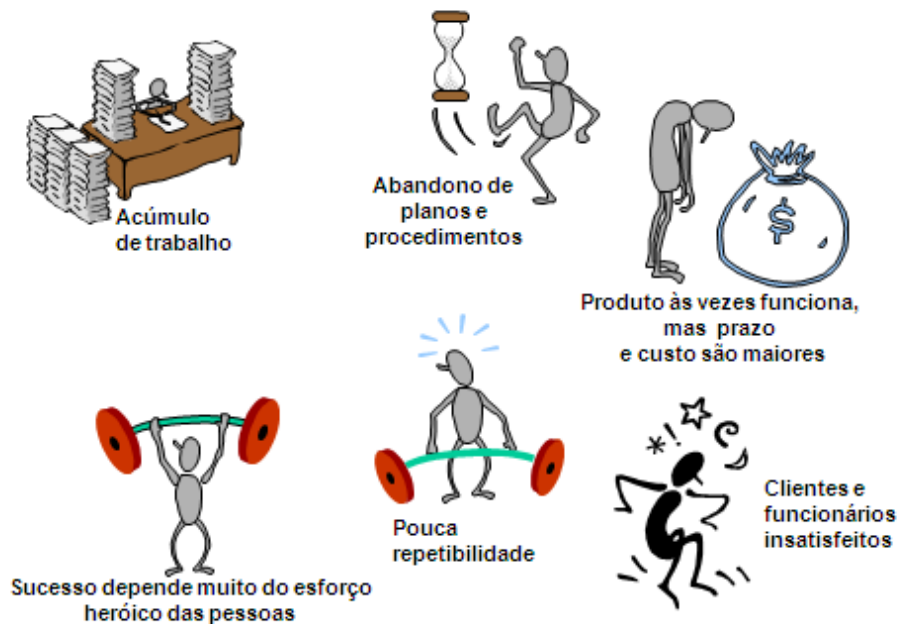
### 2.3.1 Aspectos gerais

A flexibilidade na adaptação e definição de processos é equilibrada assegurando a consistência apropriada dos processos em cada organização. Essa flexibilidade é necessária para tratar as variáveis de contexto, por exemplo: domínio de atuação, tipo de cliente, limitações de custo, cronograma e qualidade, dificuldades técnicas do trabalho e experiência das pessoas que estão implementando o processo.

Os padrões dentro da organização têm que atingir padrões organizacionais, objetivos e estratégias que sejam apropriadamente tratados, além de dados e lições aprendidas dos processos, para que possam ser compartilhados. Essa afirmação relaciona processo a ações e acontecimentos realizados por pessoas, para o desenvolvimento e para a manutenção tanto do software como de outros produtos associados a ele, por exemplo: os manuais de usuário.

Tanto o produto de software como o produto manufaturado têm como requisito, para um processo de qualidade, a necessidade de ser uma atividade sistemática e passível de repetição, independentemente de quem a execute.

A Figura 2.3 mostra a situação de muitas organizações de software. Existe um grande acúmulo de trabalho, que acarreta o abandono de planos e procedimentos, quando estes estão definidos. As consequências são produtos de software que, às vezes, funcionam, mas os prazos e custos são frequentemente maiores do que os previstos.



**Figura 2.3 – Situação de muitas organizações de software. Fonte: MAGNANI (1998).**

As organizações percebem a situação caótica, mas encontram dificuldades em mudar esse quadro em função de uma série de fatores, dentre os quais:

- As organizações estão reagindo a crises constantes;
- A ausência de guias, da mesma forma que existe nas demais engenharias, de métodos testados e comprovados sobre como desenvolver e fazer manutenção de software. Boas práticas estão sendo utilizadas, mas são pouco disseminadas;
- A maioria dos problemas nas organizações de software é de natureza gerencial, e não técnica. A tecnologia e a capacitação dos profissionais que essas organizações já possuem, poderiam gerar melhor qualidade;
- A alta velocidade de mudanças tecnológicas envolvidas, como o próprio hardware;
- O baixo estímulo à participação do usuário no processo de software, o que conduz a uma especificação incompleta de suas necessidades;
- A crescente demanda por novos processos de automação e pela manutenção dos já desenvolvidos acarreta problemas, como mostrado na Figura 2.3;

- A falta de conhecimento sobre como mudar essa realidade e a dificuldade em aplicar os princípios da qualidade ao software.

Apesar da demanda por qualidade, os avanços da Engenharia de software não conseguem acompanhar e solucionar os problemas, sendo necessários outros esforços nesse sentido. Tais esforços serão tratados nas próximas seções.

Tendo como referência a Figura 2.3, fica clara a dependência existente entre a qualidade do produto de software e seu processo, mas só trabalhar essa dependência não garante um produto de qualidade. A avaliação e a melhoria da qualidade podem ser tratadas em duas abordagens diferentes: processo e produto. A visão de processo procura estabelecer práticas que propiciem uma redução na variabilidade dos resultados e maior confiança na obtenção do produto final. Procura-se uma maturidade do processo pela sua definição explícita, gerenciamento, controle e efetividade.

A visão de produto propõe a avaliação de produtos intermediários e do produto final, verificando o atendimento às necessidades a que o produto se destina.

As duas abordagens são necessárias e complementares, pois, se o processo gera uma expectativa de redução na variabilidade na qualidade dos produtos gerados, não se tem, como decorrência direta, a garantia da qualidade do produto, porque sempre há fatores imponderáveis, que escapam ao controle do processo de produção e podem afetar o resultado final. Mais ainda, sendo o desenvolvimento de software concentrado em atividades de projeto, está mais sujeito a erros e desvios imprevisíveis. Dessa forma, qualquer esforço no sentido de melhorar a qualidade de produto de software, deve estar engajado na melhoria da qualidade de processo da organização e focado intrinsecamente no produto.

### **2.3.2 Iniciativas para a qualidade de processo de software**

Na década de 1990, houve uma grande preocupação com a modelagem e a melhoria do processo para a produção de software de qualidade, dentro do prazo e com orçamentos confiáveis.

Algumas normas e modelos desenvolvidos na busca pela qualidade de processo de software são:

- NBR ISO 9000-3<sup>5</sup> – aplicada ao desenvolvimento, fornecimento e manutenção de software;
- NBR ISO/IEC 12207 – que define os processos de ciclo de vida do software;

---

<sup>5</sup> Esta norma foi cancelada, e ainda não possui substituição na ABNT.

Na ISO existe a ISO/IEC 90003:2004 Software engineering -- Guidelines for the application of ISO 9001:2000 to computer software.

- NBR ISO/IEC 15504 – sobre avaliação de processo de software, em que um modelo para avaliação é descrito na norma ISO/IEC 15504-5, e um modelo-referência de processo é apresentado na norma ISO/IEC 15504-2;
- CMMI (CHRISISS, 2003) – desenvolvido nos EUA, substituiu o CMM;
- TRILLIUM – modelo desenvolvido no Canadá, pela Bell Canada (UHCL, 2007), em parceria com a *Bell Northern Research* e *Northern Telecom*, com o objetivo de avaliar o desenvolvimento de produtos e a capacidade de produção de fornecedores de produtos para telecomunicações. Pode ser usado também para melhoria de processos de desenvolvimento de software, e inclui orientações para sua implementação e referências cruzadas com outros modelos, como o CMM, no qual foi baseado o documento de descrição do Trillium.
- MPS – modelo de processo do software, baseado no CMMI, nas normas ISO/IEC 12207 e ISO/IEC 15504 e na realidade do mercado brasileiro ([www.softex.br](http://www.softex.br)).

No Brasil, desde 2003, existe um modelo de referência para melhoria de processo de software denominado MPS, o qual, entre outros propósitos, propõe-se a aumentar a competitividade da indústria brasileira de software, por meio de programas de qualificação de profissionais na área e de melhoria e avaliação de processos e produtos de software. Apenas a série de normas NBR ISO/IEC 15504, o modelo CMMI e o MPS terão um breve resumo, a seguir.

A melhoria da qualidade de processo de software deve ser o dia a dia de uma organização de software. Com o sucesso dessa empreitada, a previsão para o início da próxima década é um esforço concentrado na redução substancial dos prazos para a entrega de produtos pelas organizações e a exportação do software brasileiro. A pressão dos concorrentes será intensa. Organizações que forem capazes de integrar, harmonizar e acelerar seus processos de desenvolvimento e manutenção de software terão a primazia do mercado.

### **2.3.2.1 NBR ISO/IEC 15504**

A série ISO/IEC 15504 - *Information Technology – Process Assessment* define um modelo para avaliação de processo e pode ser utilizada, também, como uma referência para a melhoria de processo de software. A versão nacional publicada em 2008 é denominada norma NBR ISO/IEC 15504 Tecnologia da Informação – Avaliação de Processos, e é composta por:

- NBR ISO/IEC 15504-1 - Conceitos e vocabulário.
- NBR ISO/IEC 15504-2 - Executando uma avaliação.
- NBR ISO/IEC 15504-3 - Guia executando uma avaliação.
- NBR ISO/IEC 15504-4 - Guia sobre a utilização de resultados de avaliação.

- NBR ISO/IEC 15504-5 - Exemplo de modelo de avaliação.

Observação: As futuras normas NBR ISO/IEC 15504-6 Tecnologia da informação - Avaliação de processo – Parte 6: Exemplo de modelo de avaliação de processo de ciclo de vida de sistema e NBR ISO/IEC 15504-7 - Tecnologia da informação - Avaliação de processo – Parte 7: Avaliação de Maturidade Organizacional estão em estudo na ABNT.

O modelo da norma NBR ISO/IEC 15504 possui uma arquitetura denominada contínua, com duas dimensões: 1-dimensão de processos, na qual são definidos processos com as melhores práticas específicas de cada um; esses processos são definidos por nome, propósito, objetivos, resultados esperados, práticas e artefatos. 2- dimensão de capacidade de processo, em que são definidos os níveis de capacidade genéricos. Em cada nível de capacidade, os processos podem estar em diferentes níveis de maturidade. O conjunto de áreas de processo, com um nível de capacidade para cada processo, compõe um sistema.

A dimensão de capacidade de processo é composta por seis níveis sequenciais e cumulativos. Cada nível de capacidade é descrito na norma NBR ISO/IEC 15504, conforme é apresentado a seguir:

- Nível 0 – Incompleto: os produtos de trabalho ou os resultados dos processos são dificilmente identificados ou não são produzidos adequadamente. O propósito do processo não é satisfeito ou não é implementado.
- Nível 1 – Executado: o propósito do processo é normalmente atingido, embora isso não tenha, necessariamente, sido planejado e acompanhado rigorosamente. Existe uma consciência geral e informal de que uma ação deve ser executada e quando isso deve acontecer. As pessoas da organização reconhecem que ações devem ser executadas. Os produtos de trabalho são gerados, e existem evidências do propósito do processo.
- Nível 2 – Gerenciado: a execução do processo é planejada e acompanhada; os produtos de trabalho são gerados conforme os procedimentos e requisitos especificados. A principal distinção em relação ao nível Executado é que a execução do processo é planejada e gerenciada, e os produtos de trabalho satisfazem os requisitos de qualidade especificados, os prazos e recursos necessários.
- Nível 3 – Estabelecido: o processo é executado e gerenciado com base em um processo-padrão da organização, baseado em princípios da Engenharia de software. As práticas básicas são executadas utilizando-se uma versão adaptada e aprovada de um processo-padrão documentado. A principal distinção deste nível relativamente ao nível Gerenciado é que o processo utiliza um processo-padrão capaz de satisfazer os resultados definidos.
- Nível 4 – Previsível: neste nível o processo é executado dentro de limites de controle definidos, de forma consistente, para atingir os objetivos. A execução é gerenciada quantitativamente. Medições



detalhadas de desempenho são coletadas e analisadas, levando a um entendimento quantitativo da capacidade do processo. A principal distinção deste nível em relação ao nível Estabelecido é que o processo passa a ser executado consistentemente, dentro de limites definidos, para atingir seus resultados.

- **Nível 5 – Em Otimização:** são estabelecidas metas de eficácia e eficiência para o desempenho do processo, baseadas nos objetivos de negócio da organização. Melhorias contínuas são implementadas para satisfazer os objetivos de negócio, requerendo experiências de ideias e novas tecnologias para satisfazer objetivos e metas definidos. A principal distinção deste nível relativamente ao nível Previsível é que o processo padrão e o processo definido e padrão passa a ser alterado e adaptado, para atingir os objetivos de negócio da organização. A capacidade de um processo é definida em uma escala ordinária de seis pontos, que permite a avaliação da capacidade desde o fundo da escala (Incompleto) até seu extremo (em otimização).

A medida da capacidade dos processos está baseada em um conjunto de Atributos de Processos (PA – *Process Attributes*), os quais são utilizados para determinar se um processo atingiu uma dada capacidade. Com exceção do Nível 0, que não possui atributos de processo, todos os demais possuem.

Os níveis de capacidade e seus atributos de processo estão apresentados na Tabela 2.2.

Tabela 2.2 – Níveis de capacidade e atributos de processo

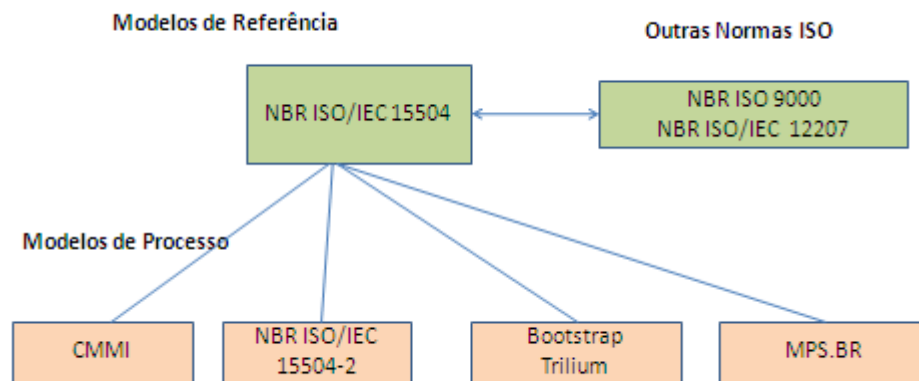
<b>Nível 0</b>	Processo Incompleto (não possui atributos)
<b>Nível 1</b>	Processo Executado PA 1.1: Atributo de Execução de Processo
<b>Nível 2</b>	Processo Gerenciado PA 2.1: Atributo de Gerência de Execução PA 2.2: Atributo de Gerência de Produto de Trabalho
<b>Nível 3</b>	Processo Estabelecido PA 3.1: Atributo de Definição de Processo PA 3.2: Atributo de Implementação de Processo
<b>Nível 4</b>	Processo Previsível PA 4.1: Atributo de Medição de Processo PA 4.2: Atributo de Controle de Processo
<b>Nível 5</b>	Processo em Otimização PA 5.1: Atributo de Inovação de Processo PA 5.2: Atributo de Otimização de Processo

Durante uma avaliação, as práticas genéricas, agrupadas de acordo com as características comuns e com os níveis de capacidade, são utilizadas para determinar a capacidade de um processo.

Para atribuir o nível ao processo, um valor tem de ser fornecido a cada atributo do processo, cujos valores estão representados na escala de medição a seguir:

- “N” – *Not adequate* (não atingido): o atributo não foi satisfeito pelo processo.
- “P” – *Partially adequate* (parcialmente): o atributo foi parcialmente satisfeito pelo processo.
- “L” – *Largely adequate* (largamente): o atributo foi amplamente satisfeito pelo processo.
- “F” – *Fully adequate* (completamente): o atributo foi completamente satisfeito.

A série NBR ISO/IEC 15504 vem sendo aplicada com resultados proveitosos para as organizações que a utilizaram para avaliação e melhoria dos seus processos. A relação dessa norma com os modelos de processo pode ser verificada na Figura 2.4, em que está representada sua paridade com as demais normas e sua predominância em relação aos modelos, que devem ser compatíveis com os requisitos dessa norma.



**Figura 2.4 – Relação da norma NBR ISO/IEC 15504 com os modelos de processo.**

A avaliação e a melhoria dos processos fundamentais são orientadas pelos objetivos de negócio da organização e envolvem o planejar, gerenciar, executar, controlar e melhorar os aspectos de software quanto a adquirir, fornecer, desenvolver, operar, efetuar manutenção e fornecer suporte.

O modelo de referência da série NBR ISO/IEC 15504 estabelece um conjunto universal de 40 processos fundamentais para a Engenharia de software, divididos em categorias, conforme mostra a, e um roteiro racional para a determinação da capacidade, por meio de uma avaliação, e também para a melhoria de cada processo.

**Tabela 2.3 – Categorias e processos da série NBR ISO/IEC 15504 (WEBER, 2001)**

Categorias	Siglas	Processos fundamentais
Cliente-fornecedor	CUS.1	Aquisição
	CUS.1.1	Preparação da aquisição

	CUS.1.2 CUS.1.3 CUS.1.4 CUS.2 CUS.3 CUS.4 CUS.4.1 CUS.4.2	Seleção de fornecedor Acompanhamento do fornecimento Aceitação pelo cliente Fornecimento Elicitação de requisitos Operação Uso operacional Suporte ao cliente
Engenharia	ENG.1 ENG.1.1 ENG.1.2 ENG.1.3 ENG.1.4 ENG.1.5 ENG.1.6 ENG.1.7 ENG.2	Desenvolvimento Análise de requisitos e projeto de sistema Análise de requisitos de software Projeto de software Construção de software Integração de software Teste de software Integração e teste de sistema Manutenção de sistema e software
Gerência	MAN.1 MAN.2 MAN.3 MAN.4	Gerência Gerência de projeto Gerência da qualidade Gerência de riscos
Organização	ORG.1 ORG.2 ORG.2.1 ORG.2.2 ORG.2.3 ORG.3 ORG.4 ORG.5 ORG.6	Alinhamento organizacional Melhoria Estabelecimento do processo Avaliação de processo Melhoria de processo Gerência de recursos humanos Infraestrutura Medição Reúso
Suporte	SUP.1 SUP.2 SUP.3 SUP.4 SUP.5 SUP.6 SUP.7 SUP.8	Documentação Gerência de configuração Garantia da qualidade Verificação Validação Revisões conjuntas Auditorias Resolução de problema

### 2.3.2.2 Modelo CMMI

Numa decisão entre amadurecer a versão do modelo CMM ou idealizar outro modelo adaptado, o SEI publicou, em 2001, o CMMI – *Capability Maturity Model Integration*, modelo que visou harmonizar alguns componentes da família CMM e da série ISO/IEC 15504, que são:

- SW-CMM ou CMM (*Capability Maturity Model for Software*) – Descreve os elementos importantes de um processo de desenvolvimento de software para melhoria contínua nas empresas.
- SE-CMM (*Systems Engineering Capability Maturity Model*) – Descreve os elementos essenciais dos processos de uma organização, que devem existir para garantir uma boa prática de Engenharia de Sistemas.
- IPD-CMM (*Integrated Product Development*) – Provê um guia sistemático para o desenvolvimento de produtos ao longo do ciclo de vida, para melhor satisfazer as necessidades dos clientes.

A versão 1.0 do CMMI, patrocinada pelo DoD e pela NDIA – *National Defense Industrial Association*, foi lançada em agosto de 2000. Seu desenvolvimento foi uma integração da indústria, do governo americano e do SEI. As metas para o desenvolvimento do CMMI eram, além de integrar os modelos, eliminar inconsistências, reduzir o custo de implementação do modelo, melhorar o entendimento do modelo e assegurar a consistência com a série ISO/IEC 15504. O modelo está em melhoria contínua, sendo as mais recentes versões:

- "CMMI for Development" (CMMI-DEV) – publicado em 26/08/2006, em <http://www.sei.cmu.edu/publications/documents/06.reports/06tr008.html>
- "CMMI for Acquisition" (CMMI-ACQ) – publicado em novembro de 2007, em <http://www.sei.cmu.edu/publications/documents/07.reports/07tr017.html>
- "CMMI for Services" (CMMI-SER) – disponibilizada em fevereiro de 2009, em <http://www.sei.cmu.edu/cmmi/models/CMMI-Services-status.html>

Com o modelo CMMI, a organização pode promover a melhoria do processo de software, por meio do aprimoramento da capacidade dos processos ou da maturidade da organização. O CMMI suporta essas duas abordagens por meio, respectivamente, das representações “contínua”, da mesma forma que a norma ISO/IEC 15504, e “por estágio”, da mesma forma que o modelo CMM.

A representação “por estágio” classifica a organização em cinco “níveis de maturidade” quanto ao seu processo de software, ao passo que a representação “contínua” classifica os processos, e não a organização, em seis “níveis de capacidade”, a saber:

- 0 – Incompleto
- 1 – Realizado
- 2 – Gerenciado
- 3 – Definido
- 4 – Gerenciado quantitativamente

- 5 – Otimizando (sempre em otimização)

As principais diferenças entre os modelos CMM e CMMI são:

- O CMM estabelece 18 áreas-chave de processo, ou KPA – *Key Process Areas* (áreas chave), e a representação é “por estágio”.
- O CMMI estabelece 22 áreas de processo, ou PA – *Process Areas* (áreas de processo), que são classificadas em quatro categorias, listadas na Tabela 2.4.

Tabela 2.4 – Categorias e áreas de processo do modelo CMMI

Categorias	Áreas de Processo – PA
Gerenciamento de processo	Foco no processo organizacional Definição do processo organizacional Treinamento organizacional Desempenho do processo organizacional Inovação e melhoria organizacional
Gerenciamento de projeto	Planejamento de projeto Acompanhamento e controle de projeto Gerenciamento de aceite do fornecedor Gerenciamento integrado de projeto Gerenciamento de risco Gerenciamento quantitativo de projeto
Engenharia	Gerenciamento de requisitos Desenvolvimento de requisitos Soluções técnicas Integração de produto Verificação Validação
Suporte	Gerenciamento de configuração Garantia da qualidade de processo e de produto Medições e análise Análise de causa e resolução Análise de decisão e resolução

A principal contribuição do modelo CMMI e da norma ISO/IEC 15504 é a liberdade que se adquire, com a representação “contínua”, de escolher quais processos devem ser trabalhados naquele momento, respeitando as necessidades prioritárias e particularidades existentes em cada organização.

### **2.3.2.3 Programa MPS.BR**

No Brasil, uma iniciativa da Softex ([www.softex.br](http://www.softex.br)), desde 2003, vem obtendo resultados positivos, no sentido de melhorar os processos de software nas organizações. O objetivo do programa MPS.BR é a Melhoria de Processo do Software Brasileiro, com duas metas a alcançar no médio e longo prazos: i) meta técnica visando à criação e aprimoramento do modelo MPS; ii) meta de mercado visando à disseminação e adoção do modelo MPS, em todas as regiões do país, em um intervalo de tempo justo, a um custo razoável, tanto em PME – Pequenas e Médias Empresas (foco principal) quanto em grandes organizações públicas e privadas. O modelo MPS possui três componentes:

1. Modelo de Referência para Melhoria de Processo de Software – MR-MPS, que fornece uma visão geral sobre os demais guias que apoiam os processos de avaliação e de aquisição.
2. Método de Avaliação para Melhoria de Processo de Software – MA-MPS, cujo propósito é verificar a maturidade da unidade organizacional na execução de seus processos de software.
3. Modelo de Negócio para Melhoria de Processo de Software – MN-MPS, um resumo executivo que estabelece as regras de negócio e relacionamentos jurídicos dos envolvidos.

O Modelo MPS é descrito nos seguintes guias, disponíveis no portal [www.softex.br/mpsbr](http://www.softex.br/mpsbr):

- Guia de Aquisição – descreve um processo de aquisição de software e serviços correlatos, baseado na norma internacional ISO/IEC 12207: 2008, e também aborda relacionamentos desse processo com o Modelo de Referência MR-MPS.
- Guia Geral – contém a descrição geral do MPS e detalha o Modelo de Referência MR-MPS e as definições comuns necessárias para seu entendimento e aplicação.
- Guia de Implementação – contém orientações para a implementação dos sete níveis do Modelo de Referência MR-MPS.
- Guia de Avaliação – descreve o processo e o Método de Avaliação MA-MPS, tendo como base a norma internacional ISO/IEC 15504.

O Guia Geral tem como base as normas ISO/IEC 12207:2008 e ISO/IEC 15504, e como complemento o Modelo CMMI. O MR-MPS define sete níveis de maturidade:

- A (Em Otimização)
- B (Gerenciado Quantitativamente)
- C (Definido)
- D (Largamente Definido)

- E (Parcialmente Definido)
- F (Gerenciado)
- G (Parcialmente Gerenciado)

Esses níveis com menor conteúdo de implementação dão ao modelo maior facilidade para as empresas que desejam melhorar seus processos. A escala de maturidade se inicia no nível G e progride até o nível A. Para cada um desses sete níveis de maturidade é atribuído um perfil de processos os quais indicam em que lugar a organização deve colocar o esforço para melhoria. Obtêm-se o progresso e o alcance de um determinado nível de maturidade MPS, quando são atendidos os propósitos e todos os resultados esperados dos respectivos processos e dos atributos de processo estabelecidos para aquele nível. Esse material é público, e a ele têm acesso todas as pessoas interessadas no assunto.

A motivação da Softex foi a de incluir, agregar valor ao processo das empresas, pois as empresas brasileiras, depois dessa iniciativa, ainda estão sendo impedidas de aderir aos padrões internacionais, em função dos altos custos praticados. Uma empresa avaliada MPS tem maiores possibilidades de obter o nível CMMI correspondente. O modelo MPS possui um maior número de níveis, o que contribui para implementações mais rápidas, avaliações com menor periodicidade, resultados em menores prazos e de modo incremental, e custos mais baixos.

A avaliação MPS tem validade de três anos, o que garante segurança adicional a quem compra software, pois a empresa tem de se manter atenta à qualidade; caso contrário, perde o nível de maturidade que havia conseguido.

Os custos da adesão ao modelo MPS são mais baixos e ainda podem ser parcialmente subsidiados. Esse modelo está desenvolvendo não somente as empresas, mas também o capital humano, formando implementadores e avaliadores, ministrando cursos com provas de capacitação, o que aumenta o nível das discussões sobre Engenharia de software no país. Os resultados desse programa estão muito acima do esperado, como pode ser visto em “Resultados de desempenho de organizações que adotaram o modelo MPS” cujos autores Guilherme H. Travassos e Marcos Kalinowski em 2008 ([www.softex.br](http://www.softex.br)) relatam nesse documento.

### 2.3.3 Comparação entre os modelos

Quanto à abordagem da qualidade de software, considerando os processos de software, isto é, processos de produzir software, existem normas e modelos para definição, avaliação e melhoria de processos de software. A série NBR ISO/IEC 15504 trata da avaliação de processos de software. O modelo CMMI – *Capability Maturity Model Integration*, desenvolvido nos Estados Unidos, define um modelo de maturidade para empresas desenvolvedoras de software e possui seu próprio método de avaliação *Standard CMMI Appraisal Method for Process Improvement* – SCAMPI. A norma NBR ISO/IEC 12207 define processos do ciclo de vida de software.

O modelo MPS, assim como o *Capability Maturity Model Integration*, possui seu próprio método de avaliação: MA-MPS – Método de Avaliação para Melhoria do Processo de Software. O guia de avaliação descreve o processo e o método de avaliação MA-MPS, com base na norma internacional ISO/IEC 15504-2: 2003. Esses modelos disponibilizam os materiais para utilização dos interessados e propõem a avaliação da capacidade e maturidade de uma organização, além de indicarem diretrizes para a melhoria. As organizações de software são enquadradas em níveis de maturidade definidos pelos modelos.

A norma NBR ISO/IEC 12207 estabelece processos, atividades e tarefas a serem aplicados durante a aquisição, o fornecimento, o desenvolvimento, a operação e a manutenção de software. Ela apresenta uma definição abrangente em relação aos processos e orienta a adaptação para sua utilização nos projetos de software implementados em uma organização.

Tabela 2.5 – Comparação entre os modelos para processo de software

Aspectos abordados	NBR ISO/IEC 12207	CMMI	NBR ISO/IEC 15504	MPS
Objetivo	Estabelecer uma terminologia e um entendimento comum para os processos entre todos os envolvidos com software	Determinar a capacitação da organização e apoiar sua evolução de acordo com os níveis estabelecidos	Conhecer e avaliar os processos da organização, determinar a capacitação e promover a melhoria	Melhorar os processos de software nas micro, pequenas e médias empresas (PMEs), a um custo acessível, em diversos locais do país
Abordagem	Definição dos processos para aquisição, fornecimento, desenvolvimento, operação e manutenção de software	Avaliação dos processos e enquadramento da organização em um dos níveis de maturidade	Avaliação dos processos da organização em relação a níveis de capacitação	Avaliação dos processos da organização e
Organizações-alvo	Organizações em geral	Organizações que necessitam de comprovação formal de sua capacidade	Organizações em geral	Organizações em geral
Definição de processos	Estabelece 43 processos, organizados em 7 áreas de processo	Estabelece 22 áreas de processos, organizados em 5 níveis crescentes de maturidade	Estabelece 35 processos, organizados em 5 categorias	Estabelece 22 processos, organizados em 7 níveis crescentes de maturidade
Flexibilidade nos aspectos definidos pelo modelo	Classificação de processos pode ser utilizada conforme os objetivos da organização	Níveis e áreas-chave de processo são a base do modelo e não podem ser alterados	Permite a definição de perfis de processo e práticas de acordo com os objetivos da organização	Níveis com menor conteúdo de implementação para facilitar as empresas e motivar a adesão ao modelo para as empresas que desejam melhorar seus processos

A série NBR ISO/IEC 15504 pode ser utilizada por organizações envolvidas em planejar, gerenciar, monitorar, controlar e melhorar a aquisição, o fornecimento, o desenvolvimento, a operação, a evolução e o suporte de software.

Uma análise comparativa desses modelos e normas é apresentada na Tabela 2.5. Essa análise considera o objetivo de cada modelo e norma, o resumo de suas abordagens, o tipo de organização a



que melhor se adaptam os processos de software considerados e a flexibilidade de aplicação de cada modelo e norma.

O foco em qualidade de produto de software reporta-se ao próximo capítulo, que trata das categorias de produtos de software mais conhecidas, que ajudam na especificação e na avaliação da qualidade do software.

## CAPÍTULO 3

### **Categorias de produtos de software e avaliação**

A evolução da tecnologia, a abrangência das aplicações e a complexidade existente na produção de produtos de software tornam a tarefa de classificar software em categorias um desafio. É possível classificar pelo domínio de aplicação, pelo modelo de negócio, pelas mesmas características construtivas, mesma disponibilidade de dados, entre outros critérios. Aqui estão descritas: a classificação tradicional e classificação teórica de Pressman (2002), a classificação da norma IEEE 1062 (1998), esta última adotada para explicar como avaliar esse tipo de produto, e uma classificação utilizada na prática, a qual pode ser utilizada em eventos - por exemplo, numa premiação.

#### **3.1 Categorias segundo Pressman**

Numa pesquisa sobre classificação de produto de software, a visão de Roger Pressman (2002) ainda nos oferece uma classificação que atende as necessidades atuais. Assim sendo, sete categorias são apresentadas para aplicações de software.

- Software básico – caracteriza um conjunto de programas necessários para dar apoio a outros programas. São suas principais características: forte interação com o hardware e compartilhamento de recursos; uso constante de processamento concorrente, que exige o escalonamento; e estruturas de dados muito complexas. Exemplos: compiladores, editores de texto, sistemas operacionais.
- Software de tempo real – caracterizado por monitorar, analisar e controlar eventos do mundo real. Coleta dados de um ambiente externo e transforma as informações, dependendo do tipo de aplicação e de acordo com a necessidade do sistema; muitas vezes, ainda devolve ao ambiente externo controles e saídas. Um software de tempo real caracteriza-se por responder dentro de pequenos espaços de tempo. Quando esse tempo não é obedecido, pode causar péssimos resultados. São exemplos desse tipo de software: controle de navegação, controle de voo, sistema de injeção eletrônica, sistema de direção, jogos de computador e sinalizações.
- Software comercial – desenvolvido por empresas com o objetivo de lucrar com sua comercialização. É a categoria que tem a maior demanda de mercado, pois realiza a maior parte das automações da sociedade, oferecendo soluções para as mais diversificadas práticas. Neste software, dados são organizados de forma a facilitar as operações comerciais e as decisões administrativas, utilizando também técnicas de computação interativa. São exemplos: controle de estoque, folha de pagamento, contas a pagar e a receber.

- Software científico e de engenharia – são caracterizados por algoritmos de processamento numérico. São exemplos: sistemas de astronomia, controle da dinâmica orbital de naves espaciais, sistemas de manufatura automatizada.
- Software embarcado – produto usado para controlar outros produtos e sistemas, não podendo ser tratado ou utilizado separadamente. Caracteriza-se por utilizar uma memória de leitura e usar rotinas limitadas e particulares, sendo responsável por toda a inteligência instalada em alguns equipamentos. São exemplos: controle de teclados em microondas, controle de sistemas digitais em automóveis, como painel de controle de combustível ou sistemas de freio.
- Software de computador pessoal – são os programas de software utilizados em computadores de uso pessoal. Exemplos: editores de texto, planilhas eletrônicas, gerenciamento de dados, acesso a banco de dados, entre outros.
- Software de inteligência artificial – utilizado para uso de algoritmos não numéricos, para resolver problemas complexos. Outra questão interessante da Inteligência artificial são os sistemas baseados no conhecimento e também os sistemas de reconhecimento de padrões, tais como imagem ou voz. São exemplos: sistemas com entrada pelo reconhecimento de voz do usuário, sistemas de reconhecimento de imagens, como: digitais, foto, entre outras.

Nos termos dessas características, considera-se que os produtos tratados neste livro são os que Pressman chamou de software comercial, o software desenvolvido por uma empresa com o objetivo de lucrar com sua utilização.

### 3.2 Caracterizações da norma IEEE 1062

Essa norma é perfeitamente aplicada para a categoria de software comercial de Pressman. É uma norma específica para a aquisição de S&SC – Software e Serviços Correlatos, e está em conformidade com a norma NBR ISO/IEC 12207. Determinar a caracterização ou a classificação do software, entre outros aspectos, serve para construção do modelo de qualidade e para determinar os respectivos itens de uma avaliação.

A classificação adotada pela IEEE STD 1062 (1998), para os produtos de software, é definida conforme o grau de liberdade que tem o usuário para definir e especificar suas funcionalidades. Segundo a norma, a Tabela 3.1 mostra que há três tipos de produtos de software:

- COTS (*Commercial Off-The-Shelf-Software*). Esse tipo de software é desenvolvido pelo fabricante, com os requisitos pelos quais muitos usuários podem ser beneficiados. Não apresenta nada de particular ou específico para um cliente: todos os clientes fazem uso do mesmo produto.
- MOTS (*Modified Off-The-Shelf-Software*). Existe um produto-padrão e, a partir desse padrão, serão desenvolvidas particularidades para clientes, diferenciando o produto ao longo do tempo.

- **CUSTOMIZADO** ou **FD** (*Fully Developed Software*). Neste tipo, o software é totalmente desenvolvido para o cliente. Há necessidade de projeto, especificação de requisitos, entre outras etapas do ciclo de vida de um software. Evidentemente, é o produto que precisa de mais tempo para ser desenvolvido, sendo o mais complexo em termos de aquisição, controle e aceitação pelo cliente.

Tabela 3.1 – Características de produtos de software segundo a IEEE 1062

Características	COTS	MOTS	CUSTOMIZADO ou FD <sup>[1]</sup>
Escopo	Fixo	Parcialmente customizado	Totalmente customizado
Adequação ao uso	Demonstrado	Demonstrado em aplicações similares	Sem precedentes
Manutenção	Sem controle	Controle parcial	Controle total
Prazo de entrega	Imediato	Pequeno – grande	Grande
Custo de aquisição	Baixo – médio	Médio – alto	Alto
Qualidade (ISO/IEC 9126-1)	Não controlada	Parcialmente controlada	Controlada em sua maior parte
<sup>[1]</sup> Parcialmente ou completamente terceirizado			

Cada um desses grupos, como descrito na Tabela 3.1, tem as características de qualidade bem determinadas. Assim, um produto chamado COTS pode ter um modelo de qualidade e ser avaliado segundo os critérios genéricos baseados nas normas. Esse modelo de qualidade e o método de avaliação serão descritos nos próximos capítulos e podem ser utilizados em processos de aquisição de software.

Um aspecto do uso de software que ilustra a diferença entre COTS e MOTS é a necessidade do MOTS de estar em constante manutenção ou evolução. Toda falha de software indica um erro no projeto ou no processo por meio do qual o projeto foi traduzido em código executável. Portanto, a manutenção ou a evolução do software envolve consideravelmente mais complexidade do que a manutenção pura e simples. Sendo assim, esse tipo de produto pode também ser avaliado segundo os critérios de manutenibilidade apresentados nas normas.

Para o software desenvolvido especificamente para um cliente, como é o caso dos FDs, existe a necessidade de montar seu modelo de qualidade, para desenvolver o produto e avaliar seu desempenho segundo as características de qualidade tratadas nas normas, juntamente com todas as características específicas da aplicação e do domínio ao qual pertence.

### 3.3 Categorias para prêmios de software

Numa aplicação prática, tratando-se de software comercial do tipo COTS, pode ser mencionado o exemplo do PRÊMIO ASSESPRO (ROCHA, 2001), que foi realizado em nível nacional, de 1990 a 1999,

período em que mais de 60 empresas e produtos das mais diversas áreas de aplicação foram premiados, tendo-se utilizado a seguinte classificação:

Categoria 1. Suporte à documentação e ao planejamento: Compreende sistemas destinados à composição de documentos, organização e manipulação de dados (valores, palavras, textos, imagens, tabelas) e gerenciamento de atividades de planejamento. Alguns exemplos: editores de textos, planilhas, dicionários, gerenciadores de projeto, programas de editoração, formataadores de relatórios, agendas eletrônicas, processadores de imagens, OCR - *Optical Character Recognition* e digitalização de documentos.

Categoria 2. Software básico e de apoio ao desenvolvimento: Compreende sistemas que apresentam funções de controle básico do computador, de seus periféricos e sistemas destinados ao suporte e desenvolvimento de programas aplicativos. Exemplos: sistemas operacionais, compiladores, redes, servidores, geradores de aplicativos, ferramentas CASE - *Computer-Aided Software Engineering*, gerenciadores de atividades computacionais, gerenciadores de arquivos e banco de dados.

Categoria 3. Sistemas de engenharia e ferramentas gráficas: Sistemas que utilizam cálculos de engenharia ou cálculos complexos; sistemas que reúnem ferramentas gráficas na execução de suas funções; sistemas de automação industrial, monitoração e controle de processos. Exemplos: CAD's - *Computer-Aided Design*, geradores de desenhos e gráficos, bibliotecas de funções gráficas; programas de cálculo de edificações, pacotes estatísticos; sistemas de apoio à programação de CLP (Controlador Lógico Programável) ou CN (Controle Numérico); sistemas de controle de tráfego, sistema de controle de processo industrial.

Categoria 4. Sistemas de informação específicos: Compreende sistemas específicos destinados à realização da atividade-fim do usuário nas áreas administrativa, comercial e financeira, em organizações e diversos ramos de atividades. Exemplos: folha de pagamento, controle de estoque, vendas, contabilidade, faturamento, controle fiscal, sistemas específicos para escolas, bancos, saúde, setor jurídico, comércio.

Essa categoria pode ser subdividida em outras categorias de domínios específicos como:

Categoria 4.1. Software de Infraestrutura: Compreende sistemas integrados destinados à infraestrutura e segurança das transações e desenvolvimento de sistemas. Exemplos: segurança da informação e gestão de riscos, gestão de conteúdo, ferramentas de apoio, ferramentas de desenvolvimento de aplicativos, linguagens, sistemas colaborativos.

Categoria 4.2. Varejo: Compreende os sistemas de gestão de varejo. Exemplos: ponto de vendas, emissão de cupão fiscal, nota fiscal eletrônica, sistemas de radiofrequência, gestão de armazém/depósito.

- Categoria 4.3. Saúde: Compreendem sistemas integrados destinados à gestão hospitalar, centros de saúde e laboratórios de análise. Exemplos: prontuário eletrônico, gestão de centro cirúrgico, “administração do paciente”, “saúde do trabalho”.
- Categoria 4.4. Agropecuária: Compreende sistemas integrados destinados à gestão das atividades de agronegócio. Exemplos: gestão da plantação, gestão da colheita, gestão da pecuária, gestão de frigoríficos, controle de frotas, folha de pagamento.
- Categoria 4.5. Comércio Exterior: Compreende sistemas integrados destinados gestão do comércio exterior. Exemplos: gestão de processos de exportação e importação, gestão de pátio, sistemas logísticos, gestão de armazéns, gestão alfandegária.
- Categoria 4.6. Sistemas Destinados à Gestão Pública: Compreende sistemas integrados destinados à gestão da administração pública: Exemplos: folha de pagamento, sistema contábil, ponto eletrônico, gestão de RH.
- Categoria 4.7. Sistemas Destinados a Concessionárias de Serviços Públicos: Compreende sistemas integrados destinados à gestão dos serviços públicos. Exemplos: controle de consumo de energia elétrica, sistemas de bilhetagem, sistemas de cobrança.
- Categoria 4.8. Sistemas Destinados ao Setor de Indústria: Compreende sistemas específicos do setor de indústria. Exemplos: automação, controle da produção, modelagem e simulação, otimização de processos.
- Categoria 5. Sistemas de informação integrados: Compreende sistemas integrados destinados à realização da atividade fim do usuário nas suas áreas de atividades e que permitam a troca automatizada de dados com o ambiente. Exemplos: controle de ponto e/ou acesso integrado com outros sistemas, sistemas integrados, tais como faturamento, controle financeiro, estoque, contas a pagar.
- Categoria 6. Educação e entretenimento: Compreende sistemas que visam disseminar conhecimento, como cultura geral ou específica e programas de entretenimento, que fornecem informações de forma organizada. Exemplos: cursos, jogos, enciclopédias, programas de alfabetização e documentários.
- Categoria 7. Software específico para internet e intranet: Compreende sistemas que englobam ferramentas de desenvolvimento específicas para utilização de internet e intranet. Exemplos: software de comunicação, segurança, construção de *home page*, WEB.
- É importante categorizar produtos de software para que se possa avaliar a qualidade do produto, considerando o custo-benefício de uma avaliação.

### 3.4 Avaliação de produto de software

Avaliação pode ser vista como o exame sistemático para determinar até que ponto uma entidade é capaz de atender os requisitos especificados. Avaliar um produto de software é atribuir certo valor a esse produto, com base em requisitos pré-estabelecidos e sob demanda de um patrocinador.

Os requisitos são derivados dos modelos de qualidade definidos nas normas de produto de software, e o patrocinador pode ser um usuário, um comprador e até mesmo o próprio desenvolvedor ou fabricante. Os requisitos devem ser, sempre que possível, objetivos. Se a avaliação do software for do ponto de vista de um desenvolvedor, as características do software serão avaliadas, provavelmente, em relação a suas qualidades intrínsecas, como por exemplo: elegância do design, correção lógica, desempenho, robustez, concisão, tolerância a falhas, portabilidade. Neste caso, esses requisitos devem ser explicitados de forma objetiva.

Se a avaliação de software for do ponto de vista de um usuário, em que as características a serem avaliadas estão relacionadas à sua aplicabilidade, utilidade, facilidade de uso, etc. A tarefa de avaliação não deve tratar apenas das questões complexas relacionadas à ciência da computação e engenharia de software; deve, também, responder a questões como:

O produto satisfaz as necessidades de quem vai usá-lo?

É fácil de usar?

Como é o seu nível de desempenho e a quantidade de recursos utilizados?

Faz o que foi proposto de forma certa?

Está de acordo com as normas, leis do país onde vai ser utilizado?

É fácil de ser modificado?

Há riscos quando se fazem alterações no código?

É imune a falhas?

É capaz de recuperar dados em caso de falha?

Está de acordo com padrões de portabilidade?

Tais tipos de pergunta podem ser respondidos de forma sistemática e sem a interferência do avaliador, ou seja, eliminando a subjetividade do processo de avaliação. Para que isso aconteça, os resultados da avaliação devem ser baseados em fatos, e não influenciados pelas opiniões do avaliador.

Para software da categoria COTS, numa avaliação genérica, podem ser utilizadas as seguintes normas aplicadas à qualidade de software:

- NBR ISO/IEC 9126-1, 2003;
- NBR ISO/IEC 25051, 2008 e
- A série de normas NBR ISO/IEC 14598.

### 3.4.1 Método de avaliação genérico (MEDE-PROS®)

Um método de avaliação genérico independe da categoria do software que se deseja avaliar, tratando apenas do modelo de qualidade, explícito nas normas, de forma equalizada. Ou seja, todos os aspectos de qualidade são avaliados tendo o mesmo valor agregado.

O MEDE-PROS® – Método de Avaliação de Qualidade de Produto de Software foi desenvolvido para avaliar a qualidade de produto de software, tendo como referência as normas NBR ISO/IEC 9126 e NBR ISO/IEC 12119<sup>6</sup>. Este método não está especializado para nenhuma área de domínio, sendo um exemplo de método de avaliação genérico.

O estabelecimento de normas para a avaliação da qualidade de produto de software abre a possibilidade de existir uma base conceitual comum, que pode levar à aceitação universal de avaliações e certificações de produtos. Um método de avaliação genérico pode ser formado por três componentes: 1-Lista de Verificação, 2-Manual do Avaliador, 3-Modelo de Relatório de Avaliação.

1- A Lista de Verificação é uma ferramenta de avaliação que ajuda os avaliadores, durante o processo de avaliação da qualidade de produtos de software, a realizar uma inspeção sistemática da qualidade. A qualidade do produto de software, decomposta hierarquicamente em um modelo que contempla as características e subcaracterísticas do produto, pode ser usada como uma lista de verificação de tópicos relacionados com a qualidade. Pode ser elaborada tomando-se como base as normas de qualidade citadas anteriormente. Outras normas, tais como a série de normas ISO 9241 (1996), ANSI/IEEE 1063 (1987) e o método ERGOLIST (2008) também ajudaram na criação dessa Lista de Verificação.

Essa lista pode ser composta por um conjunto de atributos; e esses, por um conjunto de questões. Em um processo de avaliação de software, os seguintes elementos que compõem um software poderão ser avaliados: embalagem, descrição do produto, documentação do usuário, interface e software. Entretanto, nem sempre os produtos submetidos a um processo de avaliação apresentam todos esses elementos. Exemplo desse fato são os produtos fornecidos pela internet: não apresentam embalagem, uma vez que os compradores os obtêm realizando um simples *download*. Cada um desses elementos pode ser descrito conforme segue:

- A embalagem é um meio físico que acondiciona a mídia e documentos impressos.

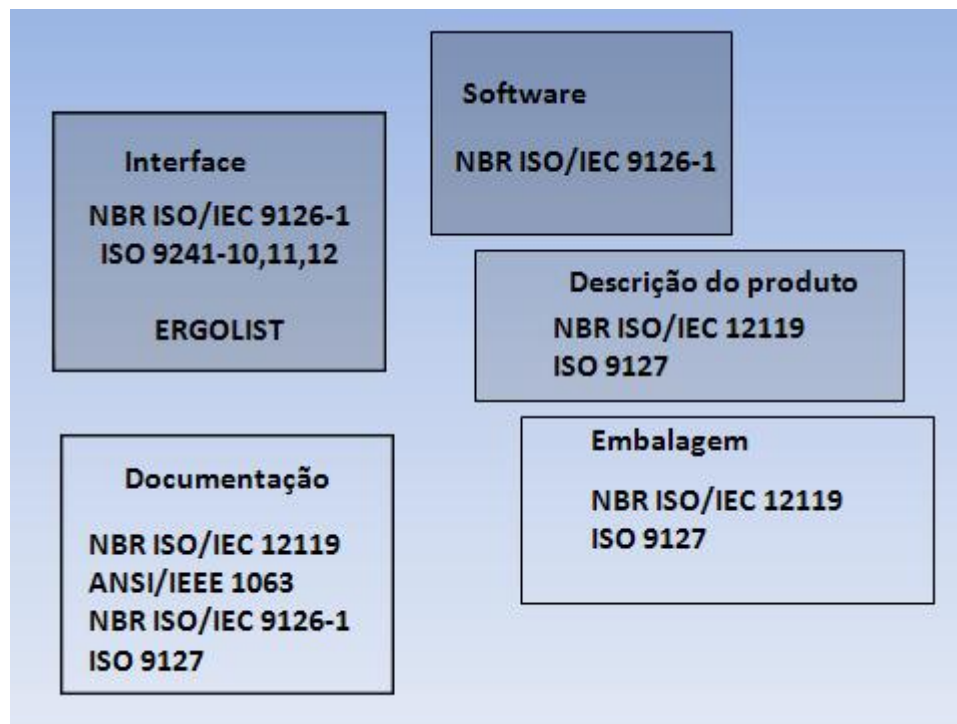
---

<sup>6</sup> O MEDE-PROS® foi concebido com a NBR ISO/IEC 12119, e sua última versão é de 2006.



- A descrição do produto é um documento exposto as propriedades do software, com o objetivo de auxiliar potenciais compradores na avaliação de adequação, antes da aquisição.
- A documentação do usuário é o conjunto completo de documentos. Está disponível ao usuário na forma impressa ou não, sendo fornecida para auxiliar na utilização dos produtos de software.
- A interface permite que as informações sejam transferidas entre o usuário e os componentes de hardware ou software de um sistema computacional.
- O software propriamente dito são as instruções (programas de computador) que, quando executadas pelo usuário, produzem a função e o resultado esperados.

A Figura 3.1 mostra a estrutura básica de uma lista de verificação e onde as normas de qualidade contribuíram para sua construção, permitindo que os elementos dos produtos de software considerados pudessem ser avaliados adequadamente. Cada um desses componentes se subdivide pelas características de qualidade, pelas subcaracterísticas, pelos atributos, até que se obtenham questões objetivas.



**Figura 3.1 – Estrutura da lista de verificação de um método de avaliação (MEDE-PROS®)**

A série de normas NBR ISO/IEC 14598 não aparece na figura porque ela fornece orientações para a realização do processo de avaliação, e esse processo será apresentado posteriormente, em um capítulo dedicado exclusivamente ao assunto. Como a avaliação do produto de software se baseia na

comparação do produto com alguns requisitos, ou ainda com necessidades explícitas e implícitas dos usuários, o trabalho realizado durante a elaboração de um método é o de transformar o modelo de qualidade, presente nas normas, em atributos; e os atributos, em um conjunto de questões, de tal forma que, avaliando o atributo por meio do conjunto de questões associadas, seja possível julgar o atendimento ou não atendimento do atributo.

2- O Manual do Avaliador deve conter um conjunto de informações para a utilização da lista de verificação, durante a avaliação da qualidade de um produto de software, e fornecer diretrizes e recomendações para a execução do processo de avaliação. Nesse manual são encontrados: explicações e exemplos de alguns atributos presentes na lista de verificação, para uma melhor compreensão do aspecto a ser avaliado; as convenções utilizadas na lista de verificação; as diretrizes para a execução da avaliação; regras e obrigações dos avaliadores se for o caso de uma avaliação profissional; informações sobre o preenchimento da lista de verificação; o material utilizado na avaliação; o procedimento da avaliação, com sugestão de uma sequência de passos adequada ao processo; orientações para a elaboração do relatório de avaliação e um glossário contendo explicações dos termos utilizados na lista de verificação.

3- Para apresentar o resultado obtido durante o processo da avaliação, pode existir um terceiro componente – o modelo de relatório da avaliação (Apêndice C). O relatório da avaliação nada mais é do que um laudo técnico sobre a qualidade do produto de software avaliado, do ponto de vista de um usuário final. Ele apresenta o resultado da avaliação, de acordo com a especificação estabelecida entre o solicitante e o responsável pela avaliação. Esse relatório destaca os aspectos do produto de software que atendem as normas de qualidade de software e os aspectos a serem revistos, originados das não conformidades encontradas durante a avaliação. Um conjunto de sugestões é fornecido, no final desse relatório, ao solicitante da avaliação, visando à adequação do produto às normas de qualidade de software, aos requisitos especificados e consequente melhoria do produto de software a ser fornecido ao mercado.

Além das informações sobre a qualidade do produto de software avaliado, outras informações podem ser apresentadas no início do relatório, com o objetivo de fornecer ao solicitante da avaliação uma visão genérica do processo realizado, tais como:

- o escopo da avaliação, que foi definido durante a fase de Especificação da avaliação, como recomendado pela norma NBR ISO/IEC 14598-5;
- a base teórica utilizada, ou seja, as normas de modelo de qualidade de produto de software;
- o processo de avaliação, executado de acordo com as fases recomendadas pela norma NBR ISO/IEC 14598-5 – Análise de requisitos da avaliação, especificação da avaliação, projeto da avaliação, execução da avaliação e conclusão da avaliação;
- os elementos do produto de software que foram avaliados;

- a quantidade de horas e de avaliadores envolvidos no processo;
- o ambiente computacional utilizado na realização da avaliação, comparado com o sugerido pelo solicitante.

Essa metodologia seguiu a abordagem de uma lista de verificação, sendo possível o desenvolvimento de outros tipos de metodologias para avaliar produtos de software. O importante é que se preserve a repetibilidade, a reprodutibilidade, a imparcialidade e a objetividade, sendo que uma das dificuldades de uma avaliação de produto de software é eliminar a subjetividade dessa avaliação.

Uma prática adotada no ciclo de evolução do MEDE- PROS® é o que se denomina meta-avaliação. Consiste basicamente na comparação das avaliações realizadas pelos pares de avaliadores, no sentido de analisar estatisticamente se as respostas da avaliação do mesmo produto são consistentes. Com essa análise estatística, é feita uma revisão da lista de verificação, com a ajuda dos avaliadores, no sentido de torná-la mais clara e objetiva. Essa prática se mostra útil, no sentido de tornar o método cada vez mais próximo das características citadas no parágrafo anterior.

### **3.4.2 Método de avaliação especialista**

Segundo Rocha, 2001, sistemas especialistas são sistemas cujo objetivo é a resolução de problemas em um domínio específico, por meio da exploração de uma base de conhecimento e de um mecanismo de raciocínio. Para sistemas especialistas ou produto de software pertencente a um domínio específico – CUSTOMIZADOS/FDs, ou seja, produtos que possuem certas características, como escopo totalmente customizável, é desenvolvido um método de avaliação a partir de suas especificações de requisitos funcionais e não-funcionais.

Para elaboração de um método dessa natureza, é necessário apresentar a base teórica das normas de qualidade existentes para produtos de software genérico e, além disso, contemplar toda a especificação de requisitos do software em questão.

Um método de avaliação é chamado de “especialista” quando tem por objetivo avaliar um produto de software desenvolvido para uma área de domínio específico. Uma das atividades realizadas durante a elaboração de um método de avaliação especialista consiste em entrevistas com os prováveis usuários do sistema e com especialistas da área. Essa atividade é uma das mais importantes e visa verificar se o sistema a ser desenvolvido, ou aquele que já está concluído, atende as necessidades explícitas e implícitas dos usuários do sistema, de tal forma que suas expectativas sejam atingidas. Esse processo de entrevistas denomina-se exploração da base de conhecimento.

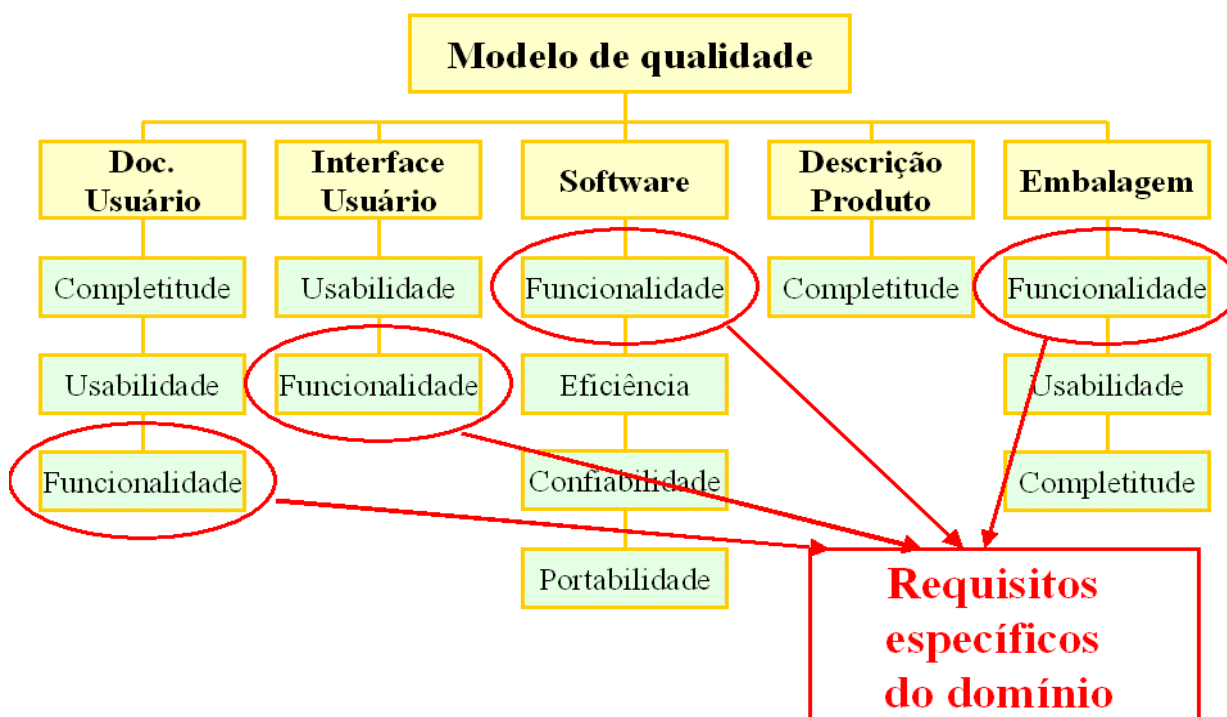
Um método de avaliação especialista pode ser desenvolvido para diversos fins, estando o momento adequado para sua elaboração diretamente vinculado ao objetivo da avaliação. Existem dois tipos de avaliação para o software: a avaliação realizada ao longo do processo de desenvolvimento do software,

em que poderão ser avaliados os artefatos gerados (ou produtos intermediários) e/ou a avaliação do produto final, sendo esta última a mais comum.

Para a criação de um método de avaliação especialista, devem ser considerados, além das características de qualidade, os aspectos inerentes ao domínio da aplicação, as tecnologias específicas utilizadas no desenvolvimento do produto e o ambiente no qual o produto de software será inserido.

Para a criação de um método de avaliação para produtos de software, seja ele genérico ou especialista, devem sempre ser considerados os requisitos funcionais e os requisitos não funcionais especificados para o desenvolvimento do produto de software.

Um método de avaliação especialista trata as características de qualidade de forma geral, ficando muitas vezes a cargo da experiência do avaliador a realização de algumas avaliações específicas, principalmente aquelas relacionadas com as funcionalidades do sistema, o que torna a avaliação bastante subjetiva, no contexto de requisitos funcionais do software. A Figura 3.2 mostra uma estrutura contendo os elementos considerados para um produto de software, as características de qualidade verificadas nesses componentes e destaca as partes a serem desenvolvidas em um método de avaliação especialista, a fim de tornar uma avaliação menos subjetiva e menos dependente da experiência do avaliador. O método aqui apresentado é derivado do método genérico; entretanto, o desenvolvimento dos detalhes da funcionalidade em todos os componentes do método é proveniente do domínio de aplicação.



**Figura 3.2 – Estrutura a ser desenvolvida numa avaliação especialista.**

O método de avaliação de software aplicativo, do tipo CUSTOMIZADO/FDs, nos diferentes domínios de aplicação, deve contar com o apoio de especialistas, seja em informática, seja no respectivo domínio, trabalhando conjuntamente. Avaliações de domínio específico apresentam uma proposta para a construção de um método de avaliação especialista, para produtos de software, baseado na especificação de requisitos encontrada, por exemplo, no conteúdo de editais. Esse tipo de método visa avaliar o produto de software na sua totalidade, considerando-se tanto os requisitos funcionais como os requisitos não funcionais.

Para o contexto do método especialista, foram utilizados os conceitos existentes para qualidade de software presentes:

Nas normas NBR ISO/IEC 9126-1, NBR ISO/IEC 12119<sup>7</sup> e NBR ISO/IEC 14598-5;

No MEDE-PROS®, considerando-se sua estrutura, base teórica e vasta utilização;

<sup>7</sup> A NBR ISO/IEC 12119 foi substituída pela NBR ISO/IEC 25051, em 2008.

Na experiência do projeto específico denominado PNAFM – Programa Nacional de Apoio à Gestão Administrativa e Fiscal dos Municípios Brasileiros, que continha um processo de avaliação regido por um edital, em conjunto com a Lei Nº 8.666 (MAITINGUER, 2004).

As contribuições do projeto citado abrangem a divulgação dos conceitos existentes para qualidade de software, a utilização desses conceitos na elaboração de um edital e de um método de avaliação especialista para produtos de software.

### **3.5 Avaliação e certificação de produto de software**

Quando se fala em qualidade de software, é importante abordar as seguintes questões: “Como avaliar a qualidade de software?” e “Existe certificado de qualidade de software?” Para esclarecer essas questões, é importante ressaltar as definições: Avaliação da qualidade – exame sistemático de quanto uma entidade é capaz de atender os requisitos especificados. Certificação de software – emissão de um certificado de conformidade de um software a certo conjunto de normas ou especificações. Para obter uma certificação é necessária a realização de uma avaliação, atendendo a conformidade de requisitos, e é emitido um laudo ou certificado.

A sociedade está começando a exigir que o software usado em sistemas críticos tenha padrões mínimos de segurança e de confiabilidade. Os fabricantes desses sistemas estão em posição não confortável, por não existir um guia a respeito do que pode ser considerado como padrões aceitáveis nessas situações. Mesmo nos lugares em que os sistemas não são de missão crítica, os produtores do software e seus clientes estão se tornando interessados nos métodos para assegurar a qualidade que possam resultar no software fornecido com as respectivas garantias. Em agosto de 2006 houve, na *McMaster University*, em Hamilton ON, Canadá, o *International Workshop on Software Certification* (Certsoft, 2006), para tratar do assunto.

No Japão, o interesse em certificação de software está aumentando, principalmente em três áreas da indústria de software: a indústria de equipamento de controle de segurança, a indústria automotriz e a indústria de instrumentos de medição. Outra dimensão carente de certificação é o DBC - desenvolvimento de software baseado em componentes. Essa tecnologia de desenvolver software tenta dinamizar suas tarefas no desenvolvimento das aplicações que contêm invariavelmente sistemas de terceira parte, chamados *Commercial Off-The-Shelf* (COTS), com funcionalidade de caixa preta. Quando integradas, as aplicações requerem a certificação de segurança. Os componentes COTS, quando individualmente certificados, ainda assim podem introduzir vulnerabilidades no sistema, se seus mecanismos de segurança estiverem mal combinados (CERTSOFT, 2006).

Para certificação de produtos de software, podem-se adotar as normas NBR ISO/IEC 14598-1 e, mais especificamente para Pacotes de Software, pode-se utilizar a norma NBR ISO/IEC 25051. Porém ainda não existe um certificado para produto de software, ou seja, não existe um órgão que seja um laboratório

de avaliação credenciado pelo INMETRO – Instituto Nacional de Metrologia, Normalização e Qualidade Industrial, ou por outras instituições semelhantes, para emitir um certificado de qualidade de produto de software. Essa situação é semelhante em outros países; pode-se encontrar certificação de software para algum domínio específico em que se exigem requisitos criteriosos específicos da área, como, por exemplo, na área aeroespacial.

Algumas iniciativas para as questões de certificação de software no Brasil já estão sendo realizadas. Como exemplo, o Painel Setorial - Acompanhamento de Mercado por Agentes Externos no Programa Nacional de Certificação de software e serviços -, organizado pelo Inmetro em 31 de maio de 2007<sup>8</sup>.

Na prática, alguns projetos foram realizados com o intuito de impulsionar a evolução do assunto. Assim, podem ser citadas as seguintes realizações:

O projeto SCOPE – *Software CertificatiOn Programme in Europe* (SCOPE, 2009), como já foi citado, e o livro de Bache e Bazzana foram os que mais influenciaram na elaboração de normas na área de qualidade de software, motivando toda a comunidade a estabelecer um padrão consensual sobre a questão da qualidade de produto de software (BACHE, 1994).

O Projeto SQUAD – *Software Quality Abroad Different regions* (SQUAD, 2009), que teve uma duração de três anos, entre 1998 e 2001, uniu-se a instituições que pesquisam e aplicam a qualidade de software na Europa e na América Latina, com o objetivo de trocar experiência e divulgar o trabalho e os resultados em cada região dos participantes. Esse projeto pôde avaliar a experiência da aplicação das normas de Qualidade de Produto de Software e do Modelo CMM em casos reais.

No Brasil, uma experiência de aplicação das normas de qualidade de produto de software foi realizada na avaliação de software no prêmio “Melhor software do ano”, pelo qual o Centro de Tecnologia da Informação Renato Archer – CTI/MCT – e a ASSESPRO – Associação das Empresas Brasileiras de Tecnologia da Informação, Software e Internet – foram os responsáveis. As informações detalhadas sobre este caso prático foram publicadas no Capítulo 13 do livro *Qualidade de Software – Teoria e Prática* (ROCHA, 2001). Esse trabalho teve cinco edições, nos anos de 1993, 1994, 1995, 1996 e 1998, com aproximadamente um total de 180 avaliações de produtos de software, os quais foram distribuídos em seis categorias de software: Sistemas de suporte à documentação e ao planejamento, Software básico e de apoio ao desenvolvimento, Sistemas de engenharia e ferramentas gráficas, Sistemas de informação específicos, Sistemas de informação integrados e Educação e entretenimento. Tal trabalho também propiciou o desenvolvimento e a evolução do método de avaliação da qualidade de produtos de software – MEDE-PROS® – e também uma grande difusão, entre as empresas de software participantes do evento, dos conceitos e requisitos definidos nas normas de qualidade de software.

---

<sup>8</sup> <http://www.inmetro.gov.br/noticias/conteudo/programaSoftware.htm>

Com essa experiência prática de avaliação de produto, o assunto evoluiu para que os processos de produção de software fossem trabalhados com maior sistematização, mostrando a interdependência da qualidade do produto durante sua produção.

A avaliação do processo de desenvolvimento de software consiste no exame dos procedimentos operacionais e gerenciais, métodos e técnicas utilizados nas fases de desenvolvimento de um produto de software, com o objetivo de identificar práticas que possam provocar problemas na qualidade do produto e de estabelecer novas práticas que evitem esses problemas. Procura-se avaliar como está sendo desenvolvido o software, por meio das seguintes disposições: verificação de gerenciamento de configuração e controle de versões (manutenibilidade); adequada organização de testes e procedimentos para controle de mudanças (confiabilidade); cuidadosa e correta elaboração de testes internos, que possam auxiliar no processo de revisão das funções (funcionalidade).

A avaliação do processo proporciona uma expectativa de geração de produtos melhores; entretanto, não garante a qualidade do produto final. Os dois tipos de avaliação são necessários e complementares e, embora distintos, com técnicas e métodos específicos, objetivam garantir a qualidade do produto final.

As propostas de avaliação do processo de desenvolvimento e dos produtos finais e intermediários têm o objetivo final de melhorar a qualidade do produto em uso, ou seja, o grau em que o produto pode ser usado por usuários específicos, para alcançar objetivos especificados com eficácia, eficiência e satisfação, em um contexto específico de uso. A qualidade em uso pode ser medida por meio da operação do produto final em condições de uso normal ou simuladas, com o objetivo de verificar a existência e o nível das características de qualidade de um produto de software, conforme definidas na norma NBR ISO/IEC 9126-1.

A avaliação do produto de software consiste no exame de um produto final resultante de um processo de desenvolvimento de software, ou de produtos resultantes de atividades de fases intermediárias desse processo. Dentre muitas razões, deve-se avaliar a qualidade do produto final para:

- a) identificar e entender as razões técnicas para as deficiências e limitações do produto, que podem manifestar-se na ocorrência de problemas operacionais ou de manutenção;
- b) comparar um produto com outro, mesmo que indiretamente;
- c) formular um plano de ação sobre como fazer o produto de software evoluir.

O processo de avaliação da qualidade de um produto de software contém os procedimentos para medições em uso, que incluem métodos de análise sobre artefatos estáticos – por exemplo, verificação da documentação do usuário – e de análise de produtos em operação – na qual a avaliação é obtida por meio de teste de integração, teste de desempenho, teste de aceitação e a execução dos programas, por exemplo.



Dessa forma, a qualidade de um produto de software é resultante das atividades realizadas no processo de seu desenvolvimento. Avaliar a qualidade de um produto de software é verificar, por meio de técnicas e atividades operacionais, o quanto os requisitos são atendidos. Tais requisitos, de maneira geral, são a expressão das necessidades, explicitadas em termos quantitativos ou qualitativos, e têm por objetivo definir as características de um software, a fim de permitir o exame de seu atendimento.

O tema tem sido tratado de forma mais específica, motivando várias iniciativas para o desenvolvimento de métodos e ferramentas para sua avaliação, dentro das respectivas visões do usuário, do desenvolvedor e do comprador. É consenso que tais visões são consideradas complementares tanto quanto a visão da Qualidade de Processo e Qualidade de Produto de Software.

A qualidade de software deve ser avaliada durante seu processo de desenvolvimento. Em seguida, deve ser avaliado o produto gerado e, finalmente, o produto em uso, pois a qualidade do processo influencia a qualidade do produto e, da mesma forma, a qualidade do processo pode ser aprimorada a partir da medição da qualidade do produto, como mostra a Figura 3.3.



**Figura 3.3 – Qualidade de software durante seu desenvolvimento (Fonte:NBR ISO/IEC 9126-1)**

Medidas no processo de desenvolvimento devem ser obtidas de acordo com requisitos exigidos na norma NBR ISO 9001, ou nos processos definidos no modelo CMMI, na norma NBR ISO/IEC12207, na norma NBR ISO/IEC 15504 e no modelo MPS.

Medidas no produto de software podem ser classificadas como medidas internas, externas ou em uso:

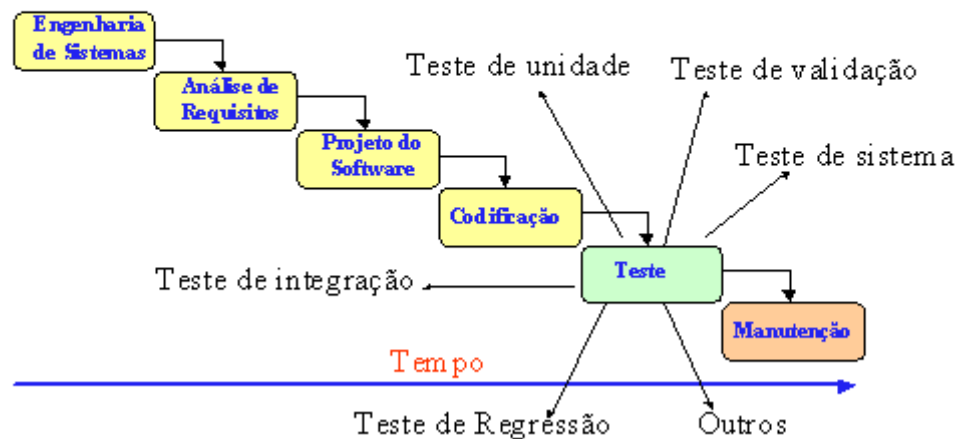
- Medidas internas – Medem atributos internos pela análise das propriedades estáticas dos produtos de software intermediários ou preparados para entrega. Medidas internas podem ser aplicadas a um produto de software não-executável, tais como uma especificação ou código-fonte, respectivamente, durante o projeto e a codificação – alguns exemplos estão descritos na norma ISO/IEC 9126-3. Exemplo: o número de linhas de código, as medidas de complexidade e o número de defeitos encontrados em uma revisão, são todas medidas de qualidade interna de software executadas no próprio produto.

- Medidas externas – Medem o comportamento do sistema do qual o software é uma das partes, pela realização de teste, operação e observação do software executável ou do sistema. Alguns exemplos dessas medidas estão descritos na norma ISO/IEC 9126-2. Esse documento define exemplos de medidas externas que se associam a atributos de qualidade e que podem ser uma referência inicial, facilitando a tarefa de definir atributos.
- Medidas de qualidade em uso – Medem quanto um produto atende às necessidades de usuários especificados, para que estes atinjam metas especificadas com eficácia, produtividade, segurança e satisfação, em um contexto de uso especificado. A avaliação de qualidade em uso valida a qualidade do produto de software em cenários de uso específicos. Mostram a visão do usuário sobre qualidade, em um ambiente contendo software, e são medidas por meio dos resultados do uso do software em ambientes, e não pelas propriedades do software. Exemplos de medidas de qualidade em uso são apresentados no relatório técnico ISO/IEC 9126-4.

### 3.6 Avaliação e teste de software

Durante a produção de um software, é necessário saber se o produto está de acordo com os requisitos especificados e também se obedece às normas sobre qualidade. Teste de software é a atividade de executar um software com o objetivo de encontrar diferenças entre os resultados obtidos e os resultados esperados. O ciclo de vida clássico de um software tem as fases representadas na Figura 3.4, e uma dessas fases é Teste.

#### Ciclo de vida para desenvolvimento de Software



### **Figura 3.4 – Ciclo de vida clássico para desenvolvimento de software.**

O processo de testes de software apresenta uma estruturação em etapas, atividades, artefatos, papéis e responsabilidades que buscam a sistematização dos procedimentos e controle dos projetos de testes. Existem vários tipos de testes, entre eles: Teste de Unidade - testa os módulos do software; Teste de Integração - testa a integração dos módulos; Teste de Validação - testa a validação do produto em relação aos requisitos; Teste de Sistema - testa o sistema como um todo; Teste de Regressão - testa o sistema após modificações. Não será tratado em detalhes aqui, pois existem muitos autores trabalhando nesse assunto. Teste é feito durante o desenvolvimento do produto.

Avaliação tem outro propósito, que é observar, com a visão do usuário, o quanto aquele produto de software atende suas expectativas.

As atividades de teste e de avaliação de software devem ser aplicadas nos projetos de desenvolvimento de software, cada uma com sua especificidade e momento adequado.

Modelos de qualidade – teóricos e práticos - e as normas de qualidade de produto serão assuntos do próximo capítulo.

## CAPÍTULO 4

### Modelos de qualidade

Este capítulo apresenta modelos de qualidade para produtos de software advindos de pesquisa teórica e da própria aplicação desses modelos em projetos e objetivos específicos originários da demanda do mercado.

Em relação aos modelos teóricos, um dos primeiros é o modelo da qualidade apresentado por Jim McCall et al, conhecido também como o modelo da *General Electric*, de 1977.

Outro modelo específico surgiu da elaboração da norma ISO/IEC 9126-1, que envolveu cerca de duas dezenas de profissionais especializados de empresas, universidades e centros de pesquisa do mundo todo, que já aplicavam os conceitos em suas atividades profissionais, antes mesmo de o texto final ser aprovado como projeto de norma.

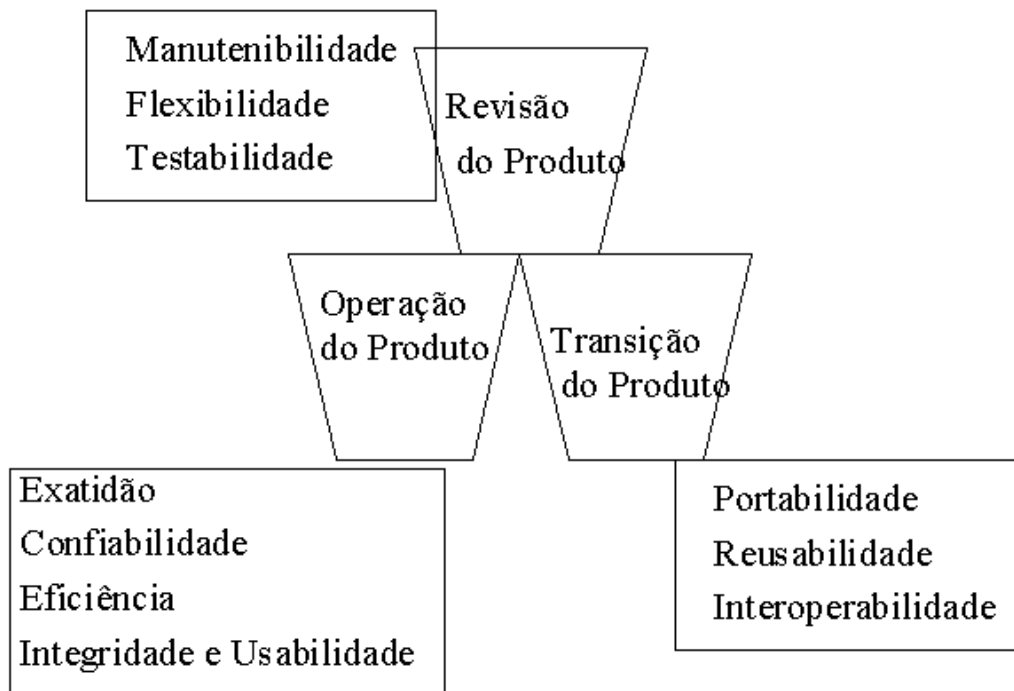
A integração crescente da economia mundial está uniformizando os conceitos de qualidade e, em consequência, tem exigido a utilização de textos normativos comuns a todos os países. Esse foi um dos motivos que levaram a Comissão de Estudo de Qualidade de Software a estabelecer como norma NBR ISO/IEC 9126-1, 2001 a tradução do texto-base da norma ISO/IEC 9126-1, 1996 *Information technology – Software product quality*, em vez de criar uma nova norma.

Uma revisão e uma nova abordagem das normas para qualidade de produto de software estão sendo trabalhadas no projeto SQuaRE – *Software Quality Requirements and Evaluation* (Requisitos e Avaliação da Qualidade do Software). O trabalho ocorre no âmbito do núcleo e extensão das normas, de 1999 a 2002, com a proposta à aprovação na ISO, que em 2005 teve a primeira norma da série publicada: a norma ISO/IEC 25000 – *Guide to SQUARE: 2005* e no Brasil a respectiva NBR ISO/IEC 25000 – Engenharia de Software - Requisitos e Avaliação da Qualidade de Produtos de Software (SQuaRE) – Guia do SQuaRE, em 2008.

Em relação aos modelos advindos das aplicações práticas desenvolvidas no âmbito do CTI, que foram baseados nos modelos teóricos citados, serão apresentados o modelo de qualidade do MEDE-PROS® e outros como sua derivação: modelo de qualidade do PNAFM e modelo de qualidade de componentes de software.

#### 4.1 Modelo de McCall

Este modelo, assim como outros modelos contemporâneos, teve origem nos meios militares americanos e foi utilizado, primeiramente, pelos seus desenvolvedores, para o processo do desenvolvimento de seus sistemas. O modelo McCall da qualidade tenta estabelecer uma ligação entre usuários e desenvolvedores, focalizando nos fatores de qualidade do software que refletem as opiniões dos usuários e as prioridades dos desenvolvedores. O modelo da qualidade de McCall tem, como mostra a Figura 4.1, três perspectivas principais para definir e identificar a qualidade de um produto de software: revisão do produto (habilidade de se submeter a mudanças), transição do produto (adaptabilidade a novos ambientes) e operação do produto (suas características da operação).



**Figura 4.1 – Modelo de qualidade de McCall, organizado em função de três tipos de características da qualidade.**

MacCall define um modelo de qualidade de forma hierárquica, com 11 fatores de qualidade. Cada um dos fatores se subdivide em 23 critérios de qualidade, que ainda podem ser subdivididos em medidas de controle, as quais não estão definidas nesse modelo. Os fatores descrevem a visão externa do software, com a visão do usuário.

Para revisão do produto:

- Manutenibilidade – esforço exigido para localizar e reparar erros em um programa. O software é fácil de ser corrigido?

- Flexibilidade – esforço exigido para modificar um programa já em utilização. O software é facilmente alterável?
- Testabilidade – esforço exigido para testar um programa frente a suas pretendidas funções. O software pode ser testado facilmente? (a facilidade de testar o software, para se assegurar de que seja isento de erros e de acordo com sua especificação).

Para transição do produto para outra plataforma:

- Portabilidade – esforço exigido para transferir um software de um ambiente (software e hardware ou arquitetura tecnológica) para outro. O software pode ser utilizado facilmente em outro ambiente?
- Reusabilidade – medida na qual um programa é facilmente reusado em um contexto diferente. O software pode ser reaproveitado facilmente?
- Interoperabilidade – esforço requerido para acoplar o software a outro sistema. O software é facilmente acoplável?

Para operação do produto:

- Exatidão ou Correção – medida na qual o software cumpre sua especificação e objetivos previstos pelo cliente. O software faz aquilo que o usuário deseja?
- Confiabilidade – medida na qual o programa executa a função pretendida com a precisão exigida. O software é preciso?
- Eficiência – quantidade de recursos de hardware e código exigidos para que um programa execute sua função (eficiência da execução e do armazenamento, e geralmente em fazer uso dos recursos; por exemplo: tempo de processador, armazenamento). Ele roda bem em seu ambiente? Ele funciona rapidamente nas especificações recomendadas?
- Integridade – medida na qual o acesso ao software ou aos dados por pessoas não autorizadas pode ser controlado. Existe proteção do software para acesso sem autorização? Os dados podem ser modificados por pessoas não autorizadas?
- Usabilidade – esforço do usuário para aprender a operar, preparar a entrada e interpretar a saída de um programa de software. O software foi projetado para o usuário?

Os fatores de qualidade descrevem tipos diferentes de características comportamentais do software, e os critérios de qualidade são atributos a um ou mais dos fatores de qualidade. A ideia do modelo de qualidade de McCall é que os fatores de qualidade sintetizados devem fornecer um retrato completo da qualidade do software.

Os 11 fatores de qualidade resumidamente são: exatidão, confiabilidade, eficiência, integridade, usabilidade, manutenibilidade, testabilidade, flexibilidade, portabilidade, reusabilidade, interoperabilidade.

Os fatores de qualidade são subdivididos nos seguintes critérios de qualidade de software segundo McCall:

- Auditabilidade – medida da facilidade com que se pode verificar a conformidade a padrões.
- Acurácia – medida da precisão dos tratamentos e do controle.
- Padrões de comunicação – medida em que padrões de interfaces de máquina, protocolos e larguras de banda são usados.
- Inteiraça – medida de quanto a implementação total da função pretendida foi conseguida.
- Concisão – medida da compactação do programa, em termos de linhas de código por função.
- Consistência – medida do uso de técnicas de projeto e documentação uniformes ao longo do ciclo de desenvolvimento do software.
- Padrões de dados – medida do uso de padrões de estruturas e tipos de dados.
- Tolerância a erros – medida dos danos que ocorrem quando o programa executa um erro.
- Eficiência de execução – medida do desempenho de um programa, em tempo de operação.
- Expansibilidade – medida da possibilidade de o projeto ser estendido.
- Generalidade – medida da amplitude da aplicabilidade dos componentes de um programa.
- Independência de hardware – medida de quanto o software é desvinculado do hardware em que opera.
- Instrumentação – medida de quanto o programa monitora sua própria operação e identifica os erros que venham a ocorrer.
- Modularidade – medida da independência funcional dos componentes de um programa.
- Operabilidade – medida da facilidade de operação de um programa.
- Segurança – medida da disponibilidade de mecanismos que controlem ou protejam programas e dados.
- Autodocumentação – medida de quanto o código-fonte apresenta documentação significativa.
- Simplicidade – medida de quão facilmente o programa é entendido.
- Independência do software básico – medida de quanto o programa é independente de particularidades não padronizadas da linguagem de programação, de sistemas operacionais e de ambientes.

- Rastreabilidade – medida da possibilidade de rastrear as decisões de projeto, desde sua análise como requisito até sua implementação como componente.
- Treinamento – medida da capacidade do software de auxiliar novos usuários na utilização do sistema.
- Eficiência de armazenamento – medida relacionada ao armazenamento dos dados.
- Controle de acesso – barreiras de segurança apropriadas.

Outro modelo, similar ao de McCall, surgiu em 1978, com o nome de Modelo de Boehm. Apresenta um modelo hierárquico de qualidade, estruturado em torno de características de alto nível, de características de nível intermediário e de características primitivas - cada uma dessas características contribuindo para a qualidade do produto de software.

Embora os modelos de McCall e de Boehm possam parecer muito similares, existem diferenças com relação à hierarquia das características. Entretanto, pode-se afirmar que foram esses modelos que inspiraram os atuais modelos de qualidade apresentados na norma e foram o início dos estudos aqui apresentados.

#### **4.2 O modelo da norma NBR ISO/IEC 9126-1**

O objetivo do *Total Quality Management* - TQM, que é o de prover as necessidades dos clientes agora e no futuro, e com a definição da norma NBR ISO/IEC 9126-1 sobre qualidade de software como sendo a “totalidade das características de um produto de software, que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas”, é possível visualizar a relação existente entre o TQM e a norma NBR ISO/IEC 9126-1.

Se, por um lado, desde o início do modelo CMM, a partir dos conceitos do TQM, procura garantir a qualidade do insumo, ou entradas, e do processo de software, por outro, a norma NBR ISO/IEC 9126-1 está preocupada em garantir que as necessidades dos clientes, em relação ao produto de software, sejam providas. Esta norma define quais são as características que um produto de software deve ter e fornece um modelo para ser utilizado em uma avaliação de verificação da presença de tais características. Isso significa que, a partir de uma entidade de software, disponível a um usuário, é possível utilizar a norma para verificar a qualidade dessa entidade.

As séries de normas NBR ISO/IEC 9126 NBR ISO/IEC 14598 e a norma NBR ISO/IEC 25051 tratam de três assuntos importantes: modelo de qualidade, processo de avaliação, avaliação de pacotes de software.



A seguir, a norma que trata modelo de qualidade será apresentada por meio de sua própria estrutura, ou seja, serão apresentadas suas diretrizes de uso, as características de qualidade de produto de software e o modelo sugerido para avaliação.

#### 4.2.1 Diretrizes para uso da norma NBR ISO/IEC 9126-1

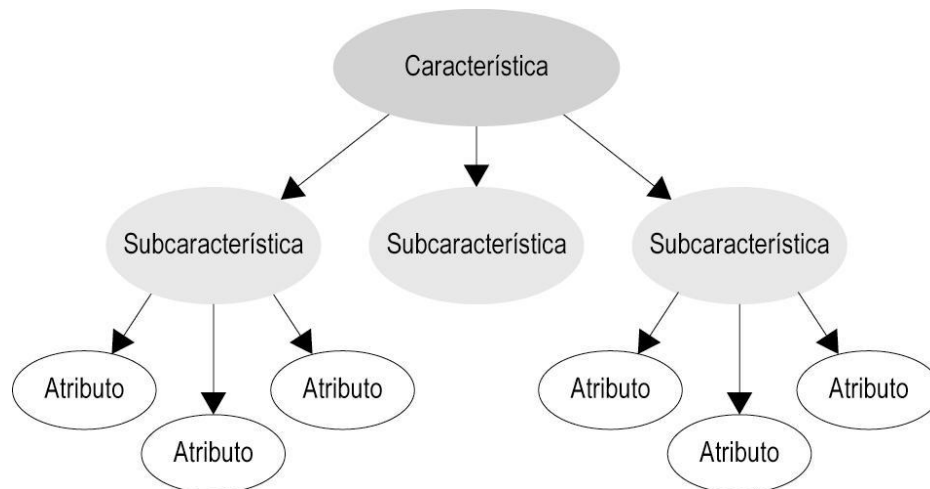
Esta norma pode ser aplicada nos seguintes momentos:

- Definição dos requisitos de qualidade de um produto de software.
- Avaliação da especificação de software para verificar se ele irá satisfazer aos requisitos de qualidade durante o desenvolvimento.
- Descrição de particularidades e atributos do software implementado, por exemplo, em manuais de usuário.
- Avaliação do software desenvolvido, antes da entrega.
- Avaliação do software desenvolvido, antes da aceitação.

A seguir, são apresentadas as características de qualidade de software que devem ser consideradas em cada um dos momentos citados anteriormente.

#### 4.2.2 Características e subcaracterísticas de qualidade de software

O modelo de qualidade apresenta uma estrutura hierárquica, definindo características, subcaracterísticas de qualidade e atributos, como mostra a Figura 4.2, de uma forma genérica.



**Figura 4.2 – Modelo básico de qualidade.**

Sabe-se, no entanto, que as características da qualidade definidas por essa norma não são diretamente mensuráveis. É necessário um desdobramento das características em níveis mais específicos, até chegar a um ponto em que se consiga obter uma medida objetiva.

O modelo de qualidade apresentado na Figura 4.2 pode ser aplicável na fase de definição de requisitos de qualidade de um software a ser desenvolvido, e também na fase de aceitação de produtos avaliando produtos intermediários e o produto final.

Os modelos de qualidade geralmente representam a totalidade dos atributos do software classificados em uma estrutura de árvore hierárquica de características e subcaracterísticas. O nível mais alto dessa estrutura é composto pelas características de qualidade, e o nível mais baixo é composto pelos atributos de qualidade do software. Esta norma fornece um modelo de propósito geral, o qual define seis amplas categorias de características de qualidade de software: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Estas podem ser subdivididas em subcaracterísticas que possuem atributos mensuráveis. O efeito combinado das características de qualidade em uma situação particular de uso é definido como qualidade em uso; também devem ser consideradas a qualidade externa e a qualidade interna:

- Qualidade em uso – “satisfazer as necessidades reais de usuários ao utilizar o produto de software, para atingir metas especificadas em contextos de uso especificados”, ou seja, o efeito da utilização do produto, medido com relação às necessidades dos usuários.
- Qualidade externa – “influenciar o comportamento de um sistema, para satisfazer necessidades explícitas e implícitas, quando o sistema que o inclui for utilizado sob condições especificadas”, ou seja, o efeito da execução das funções, medido com relação aos requisitos externos.
- Qualidade interna – “atributos estáticos do produto de software para satisfazer necessidades explícitas e implícitas, quando o software for utilizado em condições especificadas”, ou seja, o efeito das propriedades de produtos intermediários, medidos com relação aos requisitos internos – projeto e código.

A norma NBR ISO/IEC 9126-1, publicada em 2001, estabeleceu dois modelos de qualidade. Um modelo para qualidade externa e interna contendo um conjunto de seis características da qualidade de produto de software e suas respectivas subcaracterísticas, como mostra a Figura 4.3. O outro modelo é para qualidade em uso e contém quatro características de qualidade: Eficácia, Produtividade, Segurança e Satisfação, como mostra a Tabela 4.1. Esse modelo ainda requer estudos para sua aplicação.

Tabela 4.1 – Características de qualidade em uso (NBR ISO/IEC 9126-1, 2001)

Características	Definições
Eficácia	O software deve permitir que os usuários especificados atinjam metas especificadas com acurácia e

	completude, no contexto de uso especificado.
Produtividade	O software deve permitir que seus usuários diretos e indiretos empreguem quantidade apropriada de recursos em relação à eficácia obtida, no contexto de uso especificado.
Segurança	O software deve apresentar níveis aceitáveis de riscos de danos a pessoas, negócios, software, propriedades ou ao ambiente, no contexto de uso especificado
Satisfação	O software deve satisfazer usuários, no contexto de uso especificado.

Os requisitos para a escolha dessas características foram os seguintes:

- cobrir conjuntamente todos os aspectos de qualidade de software resultantes da definição de qualidade da ISO;
- descrever a qualidade do produto com um mínimo de sobreposição;
- ficar o mais próximo possível da terminologia estabelecida;
- formar um conjunto entre seis e oito características, por questões de clareza e manuseio; e
- identificar áreas de atributos de produtos de software para posterior refinamento.



**Figura 4.3 – Características e subcaracterísticas de qualidade.**

O modelo de qualidade de produto de software foi definido para apoiar o objetivo de aplicar as seis características de qualidade em uma avaliação de produto de software. A utilização do modelo

apresentado não é obrigatória; outros modelos podem ser utilizados, mas é importante perceber a necessidade de seguir um modelo tanto para desenvolvimento do software quanto para uma avaliação. O modelo apresentado é baseado na definição das características de qualidade de software e das subcaracterísticas.

As seis características de qualidade de software estão apresentadas na Tabela 4.2. Uma preocupação da norma foi definir características com um mínimo de sobreposição de conceitos entre elas.

Tabela 4.2 – Características de qualidade interna e externa (NBR ISO/IEC 9126-1, 2001)

Características	Definições
Funcionalidade	Conjunto de atributos que evidencia a existência de um conjunto de funções e suas propriedades especificadas. As funções são as que satisfazem às necessidades explícitas ou implícitas.
Confiabilidade	Conjunto de atributos que evidencia a capacidade do software de manter seu nível de desempenho sob condições estabelecidas durante um período de tempo definido.
Usabilidade	Conjunto de atributos que evidencia o esforço necessário para se poder utilizar o software, bem como o julgamento individual desse uso por um conjunto explícito ou implícito de usuários. Entende-se por usuários aqueles que utilizam software interativo, ou seja, operadores, usuário final e usuários indiretos, que estão sob influência ou dependência do uso do software.
Eficiência	Conjunto de atributos que evidencia o relacionamento entre o nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas.
Manutenibilidade	Conjunto de atributos que evidencia o esforço necessário para fazer modificações especificadas no software.
Portabilidade	Conjunto de atributos que evidencia a capacidade do software de ser transferido de um ambiente para outro.

Além de definir as características de qualidade de software e orientar quando devem ser utilizadas, a norma se preocupou, também, em apresentar um modelo de qualidade de produto de software. Sua publicação teve o grande mérito de estabelecer um modelo básico de qualidade de produto de software, transformado em referência conhecida por grande parte da comunidade.

Desde sua publicação, desenvolve-se uma opinião generalizada de que, apesar do grande benefício trazido por essa norma, ao definir um modelo de qualidade propiciando um vocabulário comum, há uma grande dificuldade em adequar sua aplicação para a avaliação prática de produtos de software. A série de normas NBR ISO/IEC 14598, complementares à norma NBR ISO/IEC 9126, procura oferecer diretrizes para essa questão.

Na Tabela 4.3, são apresentadas, para cada característica, as subcaracterísticas e suas definições.

Tabela 4.3 – Subcaracterísticas de qualidade de software

Características	Subcaracterísticas	Descrições
-----------------	--------------------	------------

Funcionalidade	Adequação	Presença de um conjunto de funções e sua apropriação para as tarefas especificadas
	Acurácia	Geração de resultados ou efeitos corretos
	Interoperabilidade	Capacidade de interagir com outros sistemas especificados
	Conformidade	Estar de acordo com normas, convenções e regulamentações
	Segurança de acesso	Capacidade de evitar acesso não autorizado a programas e dados
Confiabilidade	Maturidade	Frequência de falhas
	Tolerância a falhas	Capacidade de manter o nível de desempenho em caso de falha ou violação nas interfaces
	Recuperabilidade	Capacidade de restabelecer seu desempenho e restaurar dados após falha
	Conformidade	Estar de acordo com normas, convenções e regulamentações
Usabilidade	Inteligibilidade	Atributos do software que evidenciam o esforço do usuário para reconhecer o conceito lógico e sua aplicabilidade
	Apreensibilidade	Atributos do software que evidenciam o esforço do usuário para apreender sua aplicação
	Operacionalidade	Atributos do software que evidenciam o esforço do usuário para sua operação e controle desta
	Atratividade	A capacidade do software de ser atrativo ao usuário
	Conformidade	Estar de acordo com normas, convenções e regulamentações
Eficiência	Comportamento em relação ao tempo	Tempo de resposta, de processamento e velocidade na execução de funções
	Comportamento em relação aos recursos	Quantidade de recursos utilizados
	Conformidade	Estar de acordo com normas, convenções e regulamentações
Manutenibilidade	Analísabilidade	Esforço necessário para diagnosticar deficiência e causas de falhas
	Modificabilidade	Esforço necessário para realizar modificações e remoção de defeitos
	Estabilidade	Ausência de riscos de efeitos inesperados ocasionados por modificações
	Conformidade	Estar de acordo com normas, convenções e regulamentações
	Testabilidade	Facilidade de ser testado
Portabilidade	Adaptabilidade	Capacidade de ser adaptado a ambientes diferentes
	Capacidade para ser instalado	Esforço necessário para a instalação
	Coexistência	Capacidade do software de coexistir com outros produtos independentes, em um ambiente comum, compartilhando os mesmos recursos
	Conformidade	Consonância com padrões ou convenções de portabilidade
	Capacidade para substituir	Capacidade e esforço necessário para substituir outro software

A elaboração dessa série de normas consolidou-se na estrutura mostrada na Tabela 4.4. Estudos e revisões continuam sendo desenvolvidos em termos dessa estrutura.

Tabela 4.4 – Normas da série ISO/IEC 9126: “Status” em 2009

Norma	Título	Assunto	Estado Internacional	Estado Nacional
9126-1	Engenharia de software – Qualidade de Produto – Modelo de Qualidade	Definição das características e subcaracterísticas da qualidade	Norma, publicada em 2001	Norma, publicada em 2003
9126-2	Engenharia de software – Qualidade de Produto – Medidas externas	Exemplos de medidas externas	Relatório técnico, publicado em 2003	Relatório técnico a ser publicado
9126-3	Engenharia de software – Qualidade de Produto – Medidas internas	Exemplos de medidas internas	Relatório técnico, publicado em 2003	Relatório técnico a ser publicado
9126-4	Engenharia de software – Qualidade de Produto – Medidas qualidade em uso	Exemplos de medidas de qualidade em uso	Relatório técnico, publicado em 2004	Relatório técnico a ser publicado

No Apêndice D, estão resumidos os conceitos de qualidade para produtos de software.

Esta série de normas está sendo reestruturada numa nova série, que será apresentada na próxima seção.

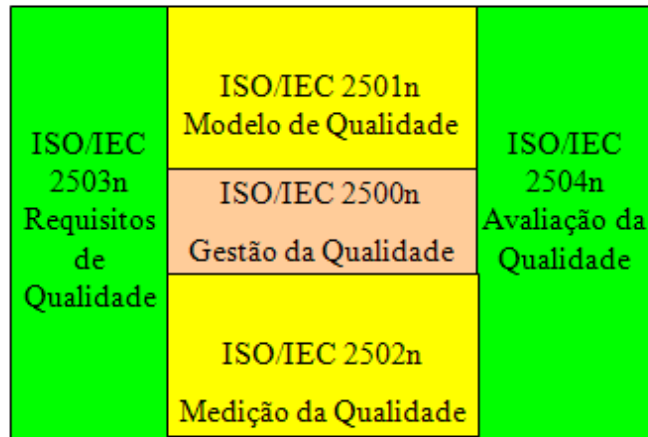
#### 4.3 A série ISO/IEC 25000

O processo de transição na ISO entre as normas ISO/IEC 9126, ISO/IEC 14598 foi longo e muito trabalhado para construção da nova série ISO/IEC 25000 – SQuaRE – *Software Quality Requirements and Evaluation* (Requisitos e Avaliação da Qualidade de Software).

A necessidade de um conjunto harmônico de documentos foi verificada quando especialistas do mundo todo concordaram que faltava clareza na utilização das normas de qualidade de produto. Assim, as séries 9126 e 14598 existentes levaram a uma lista de melhorias que foram implementadas na nova série.

Foi construída uma nova estrutura, um primeiro ajuste e a construção da estrutura necessária foram aprovados pela ISO/IEC, a versão foi revisada, e um índice detalhado foi definido. Seguiu-se, então, a escolha dos novos números que foram atribuídos aos documentos SQuaRE. Em maio de 2002, a numeração final da série foi aprovada e aplicada. A arquitetura SQuaRE é mostrada na Figura 4.4.

## Arquitetura da Série SQuaRE



Fonte: ISO/IEC 25000 SQuaRE SERIES

**Figura 4.4 – Arquitetura atual da série ISO/IEC 25000.(2009)**

A série ISO/IEC 25000 é composta pelas seguintes divisões: Gestão da Qualidade, Modelo de Qualidade, Medição, Requisitos e Avaliação. Uma representação dessas divisões é mostrada nas Figuras 4.5 a 4.9, conforme as seguintes denominações: ISO/IEC 2500n – Divisão Gestão da Qualidade; ISO/IEC 2501n – Divisão Modelo de Qualidade; ISO/IEC 2502n – Divisão Medição da Qualidade; ISO/IEC 2503n – Divisão Requisitos de Qualidade; ISO/IEC 2504n – Divisão Avaliação da Qualidade.

O objetivo das normas SQuaRE é obter uma série logicamente organizada, unificada com abrangência de dois processos principais: especificação de requisitos e avaliação da qualidade de software, apoiados por um processo de medição. Essas normas podem auxiliar desenvolvedores e adquirentes de produtos de software durante os processos de especificação de requisitos e avaliação da qualidade, estabelecendo critérios de especificação dos requisitos de qualidade, para medição e avaliação.

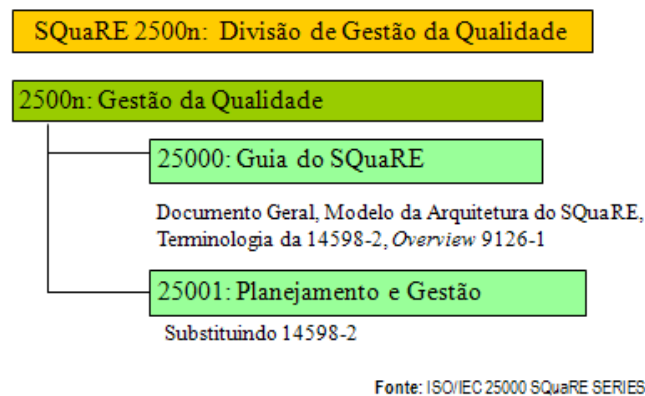
Os computadores são utilizados para automatizar uma variedade de áreas de aplicação, e seu uso é crítico para o sucesso dos negócios. A especificação e a avaliação da qualidade de produto de software são fatores para assegurar a qualidade adequada, muitas vezes difícil de ser estabelecida. Isso pode ser alcançado, definindo-se características de qualidade apropriadas ao uso pretendido do produto de software. É essencial que todas as características de qualidade importantes do produto de software sejam especificadas e avaliadas, utilizando medidas validadas ou internacionalmente aceitas.

As características de qualidade e suas medidas associadas podem ser utilizadas tanto para avaliar um produto de software quanto para definir requisitos de qualidade. A nova série estabelece critérios para a especificação dos requisitos de qualidade de produto de software, para medição e avaliação.

Está incluído nessa nova série um modelo de qualidade composto por duas partes, para alinhar as definições de qualidade do usuário aos atributos relacionados ao processo de desenvolvimento. Além disso, a série recomenda medidas para atributos de qualidade que podem ser utilizadas por desenvolvedores, adquirentes e avaliadores.

A divisão SQuaRE ISO/IEC 2500n – Gestão da Qualidade – fornece orientações sobre o uso da série SQuaRE, dando uma visão geral do seu conteúdo, de seus modelos de referência, definições, o relacionamento entre todos os documentos da série, como também orientações para planejamento e gestão para especificação de requisitos e avaliação de produto. Ela é apresentada na Figura 4.5.

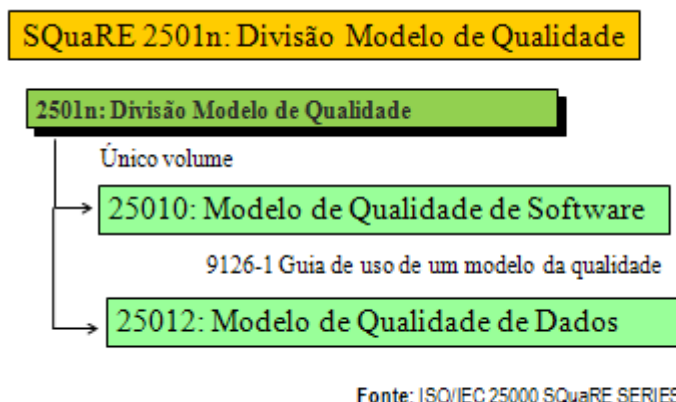
As normas que compõem a divisão Gestão da Qualidade definem todos os modelos e termos referidos por todas as outras normas da série 25000. Tal divisão fornece requisitos e orientações para uma função de apoio, que é responsável pela gestão da especificação de requisitos e avaliação de produto de software.



**Figura 4.5 – Divisão de Gestão da Qualidade da série ISO/IEC 25000.**

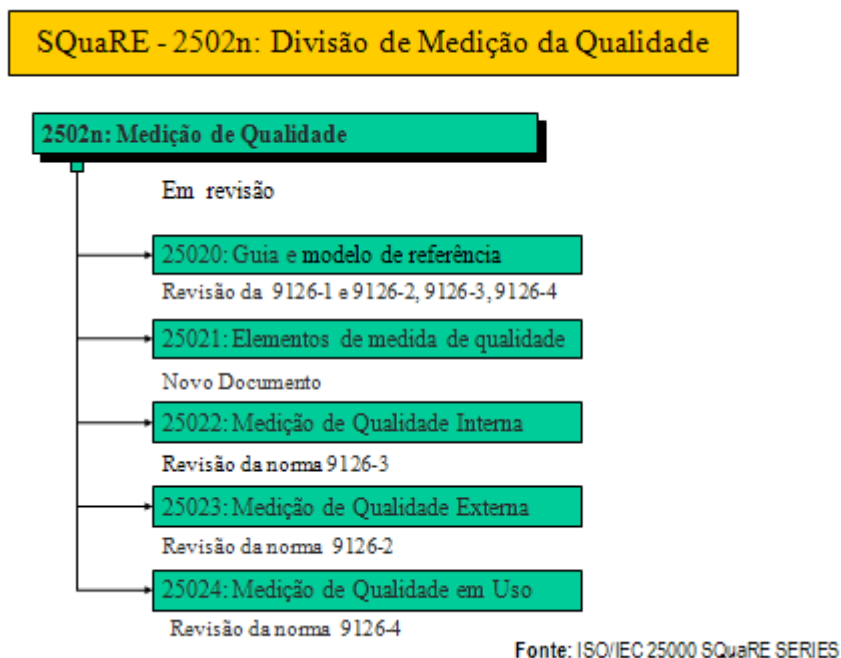
A norma que compõe a divisão Modelo de Qualidade, apresentada na Figura 4.6, propõe dois modelos de Qualidade. Um modelo que inclui características para qualidade interna e externa de software e qualidade em uso, além disso, as características internas e externas de software são decompostas em subcaracterísticas. O outro modelo define qualidade para os dados pertencentes a um sistema computacional, num formato estruturado. Também são fornecidas orientações práticas para o uso de modelos de qualidade.





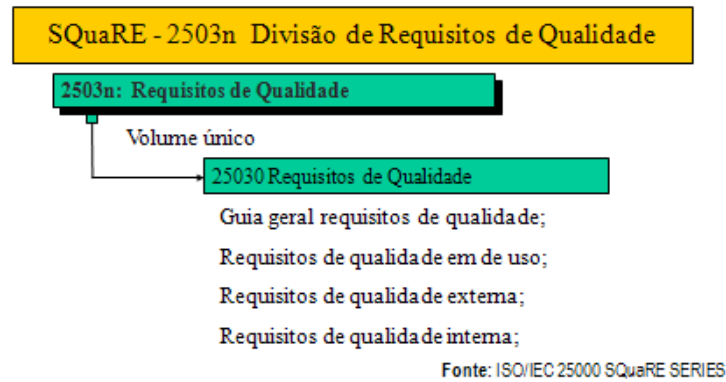
**Figura 4.6 – Divisão Modelo de Qualidade da série ISO/IEC 25000.**

As normas que compõem a divisão Medição da Qualidade contêm um modelo de referência para medição da qualidade do produto de software, algumas definições analíticas para medidas da qualidade de software e orientações práticas para aplicação. A Figura 4.7 apresenta a estrutura que contém elementos de medidas de qualidade, medição de qualidade interna, medição de qualidade externa de software e medição de qualidade em uso de software.



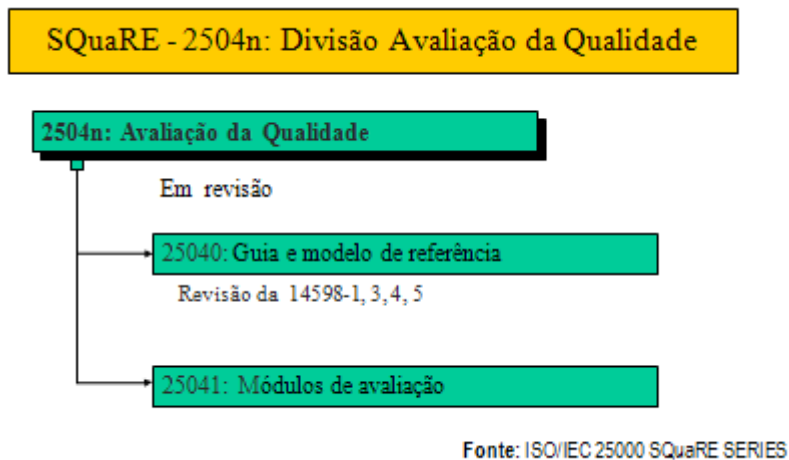
**Figura 4.7 – Divisão de Medição de Qualidade da série ISO/IEC 25000.**

A norma que compõe a divisão Requisitos de Qualidade auxilia na especificação de requisitos de qualidade, os quais podem ser utilizados no processo de elicitação de requisitos para um produto que será desenvolvido, isto é, no início do seu ciclo de vida ou posteriormente, como entrada para um processo de avaliação. Tal divisão está representada esquematicamente na Figura 4.8.



**Figura 4.8 – Divisão Requisitos de Qualidade da série ISO/IEC 25000.**

As normas que compõem a divisão de Avaliação da Qualidade, representada na Figura 4.9, fornecem requisitos, recomendações e orientações para o processo de avaliação de produto de software.



**Figura 4.9 – Divisão de Avaliação da Qualidade da série ISO/IEC 25000.**

Apresenta também uma maneira formal de documentar uma medida, utilizando um módulo de avaliação. Além disso, ela apresenta uma estrutura para a avaliação da qualidade de produto de

software. Essas estruturas são provenientes das normas ISO/IEC 9126-1 e 14598-1, 14598-3, 14598-4, 14598-5 e 14598-6.

As normas anteriores que tratam de qualidade de produto continuam sendo utilizadas até que as da nova série SQuaRE sejam publicadas. As principais diferenças entre as séries ISO/IEC 9126, ISO/IEC 14598 e SQuaRE são:

introdução do novo modelo de referência geral;

introdução de guias detalhados e direcionados para cada divisão da norma;

introdução de elementos de medidas de qualidade dentro da divisão Medição da Qualidade;

introdução da divisão Requisitos de Qualidade;

incorporação e revisão dos processos de avaliação;

introdução de orientações para uso prático em forma de exemplos;

coordenação e harmonização do conteúdo com a norma NBR ISO/IEC 15939 – Engenharia de Sistemas e de Software – Processo de Medição, publicada em janeiro de 2009.

Além dessas divisões, existe uma extensão dessa série; a sequência ISO/IEC 25050 até 25099 está reservada para ser utilizada por normas ou relatórios técnicos que estendem o SQuaRE.

A norma 25051 diz respeito a: requisitos para COTS – *Commercial Off-The-Shelf*, produtos de software de prateleira disponíveis comercialmente; requisitos para a documentação de teste, quando em desenvolvimento desses produtos, incluindo requisitos de teste, casos de teste e relatório de teste; e também instruções para avaliação de conformidade. Essa norma substitui a ISO/IEC 12119.

A norma 25062 se refere à Usabilidade. Fornece instruções para relatórios de teste de usabilidade, especificando como descrever os resultados de um teste de usabilidade num contexto de uso específico.

A Tabela 4.5 mostra a nomenclatura das siglas utilizadas unicamente na ISO, onde foram mantidas as palavras em Inglês para facilitar o entendimento das siglas; a sequência de etapas de projetos por meio dos quais o trabalho é desenvolvido; o nome do documento associado a cada etapa.

Tabela 4.5 – Etapas de projeto e documentos associados

Etapas de projeto	Documentos associados	
	Nome	Abreviatura
0 – Estágio preliminar	Preliminary work item	PWI
1 – Estágio de Proposta	New work item proposal	NP
2 – Estágio preparatório	Working draft(s)	WD
3 – Estágio Comitê	Committee draft(s)	CD

4 – Estágio de consulta	Draft International Standard	DIS
5 – Estágio de aprovação	Final Draft International Standard	FDIS
6 – Estágio de publicação	International Standard	ISO/IEC

Os trabalhos na ISO são organizados de acordo com as áreas de especialização; então são criados os Grupos de Trabalho, numerados sequencialmente na ordem em que foram estabelecidos. Os trabalhos dos grupos são chamados de projetos e são classificados conforme a etapa de desenvolvimento em que se encontram. Por isso, a Tabela 4.5 contém algumas siglas em inglês que se referem à etapa do projeto em que a norma se encontra.

A Tabela 4.6 apresenta o *status* das normas de qualidade de produtos de software até 2009.

Tabela 4.6 – SQuaRE Status: 2009

Norma	Título resumido	Estado internacional	Estado nacional
2500n	Divisão Gestão da Qualidade		
25000	Guia para SQuaRE	Publicada em 2005	Publicada em 2008
25001 (ex - 14598-2)	Planejamento e Gestão	Publicada em 2007	Publicada em 2009
2501n	Divisão Modelo de Qualidade		
25010 (ex - 9126-1)	Modelo de Qualidade	Em revisão	
25012	Modelo de Qualidade de Dados	Em revisão	
2502n	Divisão Medição da Qualidade		
25020	Guia e Modelo de Referência	Publicada em 2007	Publicada em 2009
25021	Elementos de Medida de Qualidade	Publicada em 2007	
25022 (ex - 9126-3)	Medição de Qualidade Interna	Em revisão	
25023 (ex - 9126-2)	Medição de Qualidade Externa	Em revisão	
25024 (ex - 9126-4)	Medição de Qualidade em Uso	Em revisão	
2503n	Divisão Requisitos de Qualidade		
25030	Requisitos de Qualidade	Publicada em 2007	Publicada em 2008
2504n	Divisão Avaliação de Qualidade		
25040 (ex - 14598-1)	Guia e Modelo de Referência	Em revisão	
25041 (ex - 14598-6)	Módulos de Avaliação	Em revisão	
	Extensão da série		
25051	Requisitos de Qualidade para COTS	Publicada em 2006	Publicada em 2008
25062	Formato Comum da Indústria para Relatórios de Usabilidade	Publicada em 2006	

A seguir serão apresentados modelos de qualidade baseados nas normas e advindos de aplicações práticas.

#### **4.4 O modelo de qualidade do MEDE-PROS®**

Este modelo de qualidade foi baseado na norma NBR ISO/IEC 9126-1, que padronizou uma linguagem universal para um modelo de qualidade de produto de software e é um exemplo de como aplicar os conceitos definidos nesta norma. Este modelo foi amplamente utilizado nas avaliações de qualidade de produto de software, em diversas ocasiões em que o CTI desenvolveu atividades sob demanda.

Este modelo é adequado para avaliar produto de software, pode-se dizer que é um modelo genérico para avaliar software de qualquer domínio de aplicação.

Os requisitos de qualidade para software do tipo pacote, podem ser baseados, principalmente, na norma NBR ISO/IEC 12119<sup>9</sup>, que define requisitos para essa categoria de software, a qual trouxe uma definição universal do que um pacote de software deve possuir para ter um mínimo de qualidade e profissionalismo.

Adicionalmente, foram utilizados outros requisitos baseados nas normas de documentação de usuário e de requisitos de usabilidade na interface homem-máquina, IEEE e ISO, respectivamente. Esse assunto será detalhado no próximo capítulo.

Uma estrutura inicial de um modelo de qualidade de software pode ser desdobrada a partir dos seus componentes até os seus atributos a serem avaliados.

O modelo de qualidade sugerido neste livro tem início com os três componentes definidos na norma NBR ISO/IEC 12119, que são: Descrição do Produto, Documentação do usuário, Programas e dados. Além desses componentes, serão acrescentados outros componentes, considerados importantes numa avaliação dessa natureza. A seguir, algumas considerações são expostas, com o objetivo de desdobrar esses componentes, para obter uma avaliação mais objetiva.

- Pode-se considerar “Programas e dados” como sendo o software propriamente dito, ou seja, na linguagem de Engenharia de software, é o que é executável em uma máquina tipo computador; assim, este componente passa a ser chamado simplesmente de software.
- O software, quando requer interatividade, requer dados de entrada e opções a serem escolhidas pelo usuário; apresenta uma interface, que geralmente é composta por janelas e campos a serem preenchidos pelo usuário. Dessa maneira, é importante estar avaliando a interface como um componente específico em produtos de software.

---

<sup>9</sup> O MEDE-PROS® foi desenvolvido com a norma NBR ISO/IEC 12119 e sua última versão data de 2006.

- No caso de pacote de software, este pode ser comercializado em algum tipo de embalagem; atualmente tem sido comercializado também pela internet. Nesse caso, existe uma página na internet onde o software é apresentado. Em ambos os casos, é interessante avaliar sua embalagem ou informações a ele relacionadas, localizadas no site do fornecedor do produto.

Considerando a norma NBR ISO/IEC 25051 e a forma como um software normalmente se apresenta, os componentes de um software propostos neste modelo podem ser:

- Descrição do Produto
- Embalagem
- Documentação de usuário
- Software e Interface

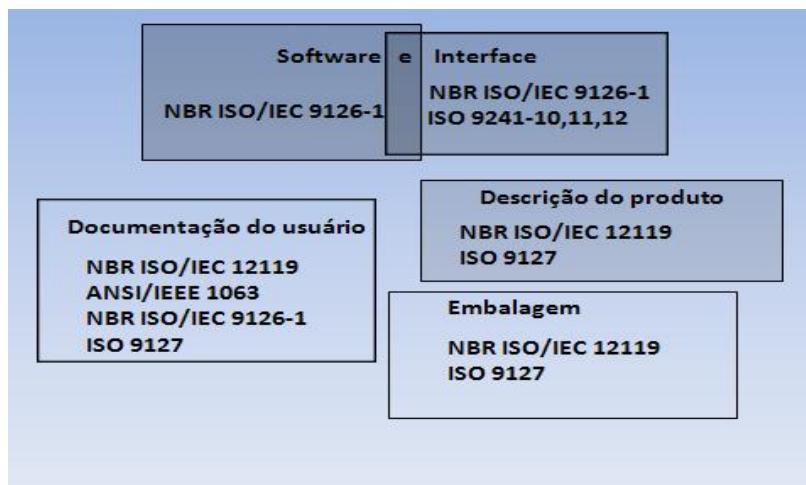
Identificando cada um desses componentes, tem-se, segundo a norma NBR ISO/IEC 25051, a *Descrição do Produto*, que é “um documento expondo as propriedades de um pacote de software, com o objetivo de auxiliar os potenciais compradores, na avaliação da adequação do produto antes da compra”. Essa descrição pode estar disponível em um catálogo próprio, na embalagem, em mídia digital de apresentação ou qualquer outro meio disponível ao usuário, independentemente da aquisição do produto. O produto deve possuir um documento único e de fácil localização.

Por *Embalagem*, entende-se o meio físico que acondiciona a mídia e documentos impressos. Exemplos de embalagem: caixa de papelão tipo cartolina, caixa de papelão resistente e com abas tipo encaixe, caixa plástica para CD, caixa de papelão com capa dura e com encaixe sobreposto, se for o caso.

*Documentação de usuário* é o conjunto completo de documentos, disponível na forma impressa, ou não; é fornecida para utilização de um produto, sendo também uma parte integrante do produto. Por exemplo: manual, complementos de manuais, atualizações, mídia digital de demonstração, selo de identificação. Documentação on-line é toda documentação em mídia e pode ser consultada por meio de qualquer meio eletrônico.

As *Funções do software* são representadas na *Interface* por meio de menus, botões, caixas de diálogo, janelas, teclas aceleradoras e outros meios.

Esses componentes de software podem ser visualizados na Figura 4.10, em que se apresentam os cinco componentes conceituais do modelo de qualidade e a indicação de normas que serão utilizadas. Os componentes software e interface são provenientes de normas diferentes, mas fisicamente é um único componente.

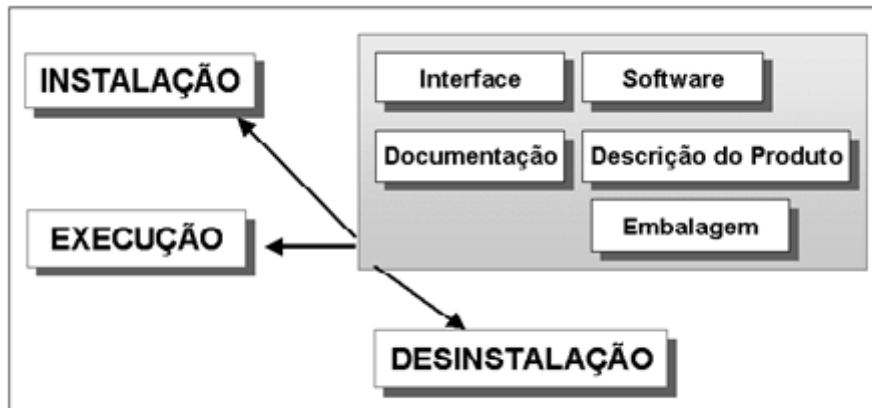


**Figura 4.10– Componentes de software.**

Para avaliar a Descrição do Produto e a Embalagem, podem ser considerados os requisitos de qualidade especificados nas normas NBR ISO/IEC 25051 e ISO 9127 (ISO 9127, 1988). Para avaliar a Documentação de Usuário, foram considerados os requisitos de qualidade especificados nas normas NBR ISO/IEC 25051, ANSI/IEEE 1063, NBR ISO/IEC 9126-1 e ISO 9127. Para avaliar o Software, foram considerados os requisitos de qualidade especificados na norma NBR ISO/IEC 9126-1. Para avaliar a Interface, foram considerados os requisitos de qualidade especificados nas normas NBR ISO/IEC 9126-1 e ISO 9241 partes 10, 11 e 12.

Além das considerações expostas anteriormente, mais uma constatação que deve ser referenciada é que, para utilizar um software, deve-se instalar o software em um sistema computacional; em seguida, executar o software por meio de sua interface; finalmente, em algum momento, deverá ser possível, ou mesmo necessário, fazer sua desinstalação do ambiente computacional.

Assim, é sugerido que sejam avaliados os componentes de um software nestas três etapas de utilização do software: Instalação, Execução e Desinstalação, como mostra a Figura 4.11.



**Figura 4.11 – Etapas de avaliação.**

Durante a instalação, a execução e mesmo na desinstalação do produto, o avaliador poderá estar medindo alguns atributos dos componentes do software – Software, Interface, Documentação, Descrição do produto e Embalagem.

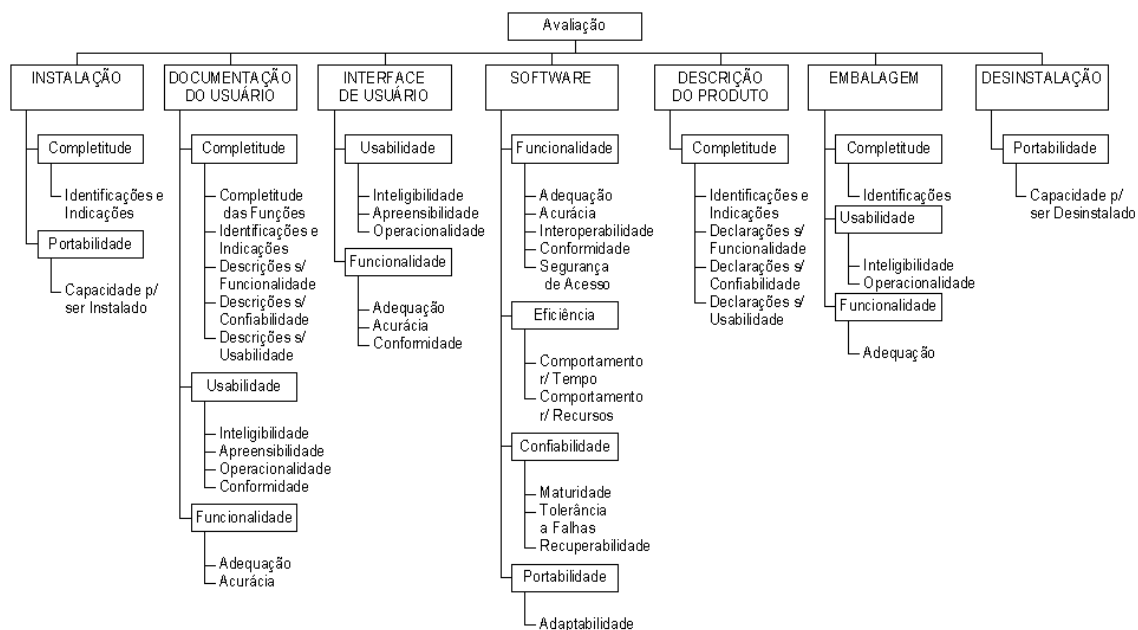
Finalmente, sugere-se a definição do modelo de qualidade para avaliação de software, estruturado pelos seus sete componentes, e cada um desses componentes com suas respectivas características de qualidade.

Cada um dos componentes é avaliado segundo as características definidas pela norma NBR ISO/IEC 9126-1. As que estão sendo utilizadas aqui são: Portabilidade, Usabilidade, Funcionalidade, Eficiência e Confiabilidade, segundo o requisito Completitude<sup>10</sup>, definido pela norma NBR ISO/IEC 12119. A característica Manutenibilidade não se aplica neste modelo, pois esta avaliação não tem o objetivo de avaliar o esforço necessário para fazer modificações no software, assim como não há necessidade de acesso a programas-fontes e produtos intermediários gerados durante o desenvolvimento do produto de software.

Uma visão detalhada desta estrutura do modelo de qualidade é mostrada na Figura 4.12. Nela estão incluídas, também, as subcaracterísticas de qualidade da norma NBR ISO/IEC 9126-1, que é o desdobramento de cada uma das características, mais os requisitos do conceito completitude, definido na norma NBR ISO/IEC 12119.

<sup>10</sup> O requisito Completitude na norma NBR ISO/IEC 25051, tem o nome de Completeza.





**Figura 4.12 – Modelo de qualidade numa visão mais detalhada.**

Este modelo de qualidade pode ser visto como um modelo genérico para avaliar qualquer categoria de software, independente da sua área de aplicação.

#### 4.5 Modelo de qualidade do PNAFM

Este modelo de qualidade foi baseado no modelo de qualidade do MEDE-PROS®, que já foi aplicado a centenas (aproximadamente 430) de avaliações de produtos de software. O método foi desenvolvido pela Divisão de Qualificação em Software do CTI, para atender ao Programa Nacional de Apoio à Gestão Administrativa e Fiscal dos Municípios Brasileiros - PNAFM, coordenado pela UCP/MF - Unidade de Coordenação de Programas do Ministério da Fazenda, com apoio técnico do PNUD - Programa das Nações Unidas para o Desenvolvimento. O financiamento foi feito pelo BID - Banco Interamericano de Desenvolvimento, no processo de qualificação de empresas e instituições e seus Conjuntos de Sistemas Aplicativos – CSA, que compõem a solução de tecnologia da informação para os projetos simplificados deste programa.

Cada CSA é constituído por 8 (oito) sistemas aplicativos, como mostra a Figura 4.13: Tributário, Financeiro, Compras, Recursos Humanos, Informações gerenciais, Legislativo, Protocolo e Ouvidoria. O objetivo do método de avaliação PNAFM é verificar se cada sistema aplicativo do CSA atende os requisitos obrigatórios, desejáveis e recomendados, mencionados no Edital 01/2001 e adendo agosto/2002, convocando empresas e instituições para pré-qualificação, com base na visão do usuário final. Neste método são avaliados comandos, integração entre sistemas, estabilidade do sistema e, como

foco principal, a verificação da conformidade dos requisitos funcionais, isto é, o atendimento preestabelecido no Edital.

Editais são de uso obrigatório para qualquer tipo de aquisição realizada por órgãos do governo, nos âmbitos federais, estaduais e municipais, tanto na aquisição de bens como para serviços. São instrumentos que dão ao processo de aquisição o caráter legal, tornando-o transparente e, conseqüentemente, imparcial.

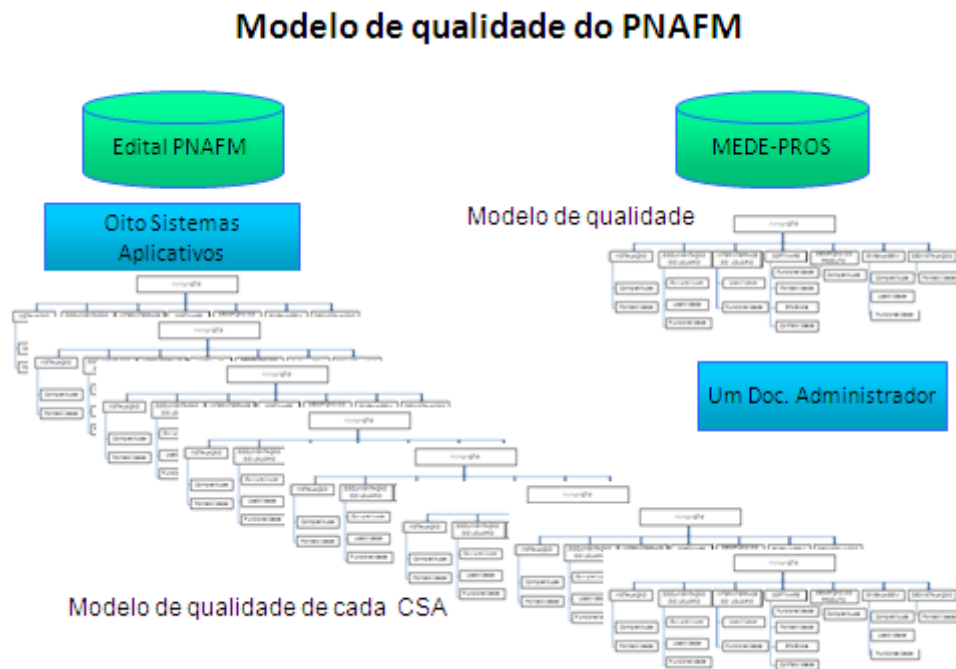


**Figura 4.13 – Conjunto de Sistemas Aplicativos do PNAFM**

Este método de avaliação, criado para atender ao PNAFM, trouxe características inovadoras, podendo derivar novas iniciativas, como o estabelecimento de um padrão mínimo (básico) da qualidade dos sistemas aplicativos utilizados na Gestão Pública Municipal Brasileira. Propondo-se a ser uma iniciativa precursora na busca da qualidade nas soluções de tecnologia da informação pública.

Quanto a sua representatividade no meio científico como pesquisa inovadora, o método se caracteriza por ter sido desenvolvido utilizando como base teórica as normas de qualidade de software que tratam de modelo de qualidade de produto e medidas, de requisitos de usabilidade de interfaces, documentação e testes e requisitos de qualidade para pacote de software entre outros, e considerando os requisitos funcionais pré-estabelecidos. Este é o primeiro método de avaliação de aplicativos desenvolvido, testado e aplicado tanto no contexto nacional.

Assim o modelo de qualidade pode ser representado, esquematicamente, como na Figura 4.14, onde se vê o modelo de qualidade do MEDE-PROS®, aplicado e adaptado para cada um dos sistemas aplicativos, com as exigências estabelecidas no Edital.



**Figura 4.14 – Modelo de Qualidade do PNAFM**

O modelo de qualidade de cada CSA dos sistemas aplicativos contempla os requisitos específicos contidos e publicados no Edital.

#### 4.6 Modelo de qualidade de componentes de software

Nas seções anteriores, foram apresentados desde o modelo de qualidade de McCall, proposto em 1977, até o modelo prático e genérico proposto na Figura 4.12. Esta seção apresenta a proposta de um modelo de qualidade para componentes de software como exemplo da utilidade do modelo de qualidade para software. Esse tipo de software pode ser utilizado no Desenvolvimento Baseado em Componentes – DBC e obedece a uma definição do que se entende por “componente de software”.

- Componentes de software – “São artefatos autocontidos, claramente identificáveis, que descrevem ou realizam uma função específica e têm interfaces claras, em conformidade com um dado modelo de arquitetura de software, documentação apropriada e um grau de reutilização definido” (VILLELA, 2000).

Para melhor entendimento da definição apresentada por Villela, alguns termos foram detalhados por Sametinger (SAMETINGER, 1997) de forma mais profunda:

- Autocontido – característica dos componentes de poderem ser reusáveis sem a necessidade de incluir/depender de outros componentes. Caso exista alguma dependência, todo o conjunto deve ser visto como o componente reutilizável.
- Identificação – componentes devem ser facilmente identificados, ou seja, devem estar contidos em um único lugar, ao invés de espalhados e misturados com outros artefatos de software ou documentação.
- Funcionalidade – componentes têm uma funcionalidade clara e específica que realizam e/ou descrevem. Componentes podem realizar funções ou podem ser simplesmente descrições de funcionalidades, por exemplo, artefatos do ciclo de vida.
- Interface – componentes devem ter interfaces claras, que indicam como podem ser reusados e conectados a outros componentes, e devem ocultar os detalhes que não são necessários para o reúso.
- Documentação – a existência de documentação é indispensável para o reúso. O tipo de componente e sua complexidade irão indicar a conveniência do tipo de documentação.
- Condição de reúso – componentes devem ser mantidos de modo a preservar o reúso sistemático, e a condição de reúso compreende diferentes informações: por exemplo, quem é o proprietário, quem deve ser contatado em caso de problema, qual é a situação de qualidade, entre outras.

Propõe-se uma estrutura de qualidade de componentes que assegure um rigoroso fundamento teórico para derivação e definição das características, subcaracterísticas, atributos e medições da qualidade, considerando, assim, diferentes abordagens (GUERRA, 2007). Dessa forma é apresentado o modelo de qualidade externa para componentes de software, na

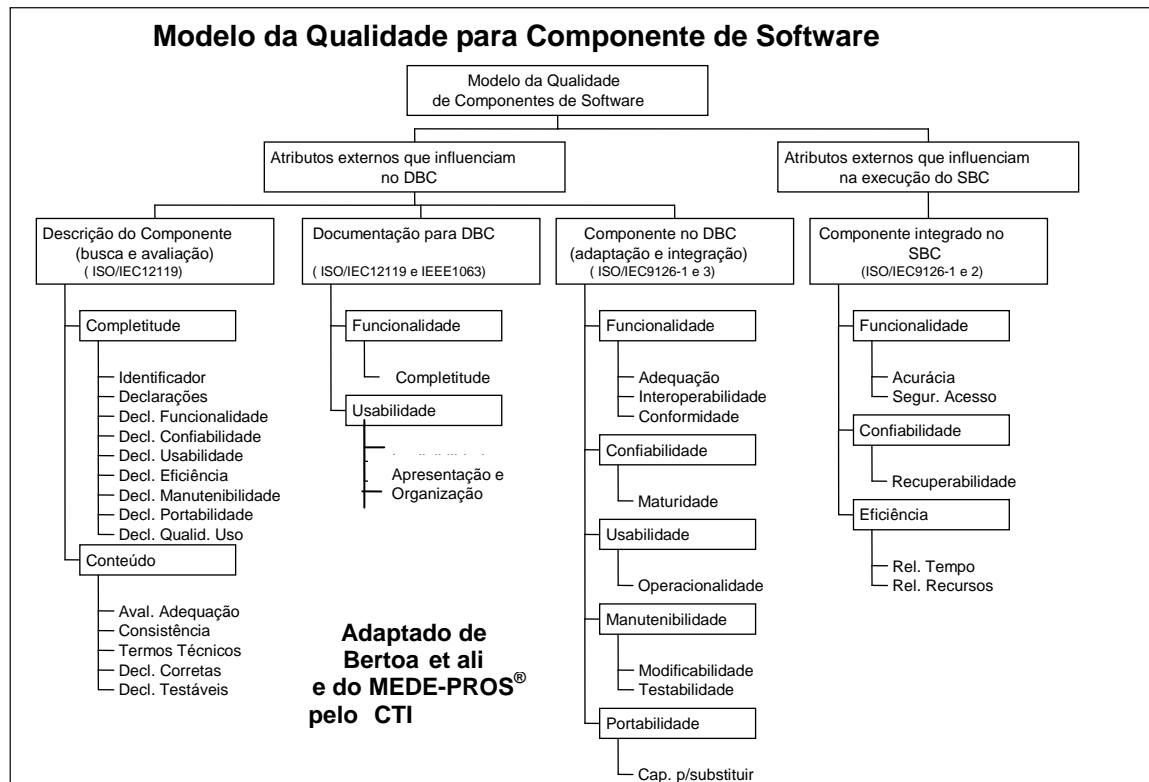
Figura 4.15. Os atributos externos são divididos em dois grupos: Aqueles que influenciam no DBC e Aqueles que influenciam na execução do Sistema Baseado em Componentes - SBC.

Para os atributos externos do DBC, considera-se que o componente deve apresentar:

Um documento de descrição de componente, o qual deve conter as informações que permitam avaliar a adequação do componente. Esse documento deve seguir uma adaptação dos requisitos do documento de descrição de produto de software, da norma NBR ISO/IEC 25051.

Uma documentação para o DBC, que deve conter todas as informações necessárias, e com o grau de detalhamento adequado. Foco principal dessa documentação devem ser as informações para a adaptação e integração do componente no sistema, nas quais são descritas suas características e funcionalidades, por exemplo: interfaces de gerência, interfaces providas e requeridas, eventos recebidos e enviados, bem como as propriedades e as interfaces de acesso às propriedades, quando estas sejam internas. Caso o componente tenha funções de interface com o usuário, estas também devem estar descritas. A documentação será avaliada com base nos requisitos de qualidade das normas NBR ISO/IEC 25051 e IEEE 1063.

A qualidade externa do componente no DBC, que contém informações de qualidade do componente que são relevantes para sua utilização no DBC. Essas informações englobam tanto características funcionais quanto não funcionais, mas que são observáveis ou medidas durante o ciclo de vida do desenvolvimento do componente.



**Figura 4.15 – Modelo de qualidade para componentes de software COTS.**

Para os atributos externos no Sistema SBC, considera-se que o componente deve apresentar atributos da qualidade externa influenciando no SBC, que também contém informações de qualidade do componente, mas que influenciam na execução do sistema.

A seguir serão apresentados os atributos de qualidade a serem avaliados neste modelo.

#### **4.6.1 Descrição do componente**

Para descrição do componente, o modelo de qualidade se decompõe em: Completitude e Conteúdo.

Como Completitude, entende-se a capacidade de verificar quão completa é a descrição do componente. O atributo Completitude se decompõe nos seguintes requisitos:

- Identificador – capacidade de verificar se a descrição do componente contém identificações importantes do componente.

As seguintes identificações devem ser declaradas:

- Identificação do Componente - por exemplo, nome do componente, versão e data;
- Descrição do modelo de componente adotado;
- Produtor/fornecedor do componente;
- Descrição dos serviços do componente, ou seja, interfaces fornecidas;
- Especificação das dependências, ou seja, Interfaces requeridas;
- Artefatos que fazem parte do componente (caso o componente possua).
- Declarações – capacidade de verificar se a descrição do componente contém declarações sobre suporte e manutenção.

As declarações devem ser seguintes:

- Suporte – declarações de suporte do componente;
- Manutenção – declarações de manutenção do componente.

- Declaração da Funcionalidade – capacidade de verificar se, na descrição do componente, existem declarações sobre as funções, valores-limites, grau de precisão, interoperabilidade e segurança de acesso do componente.

Os atributos avaliados são:

- Declaração dos serviços – declarações dos serviços que o componente oferece por meio de suas interfaces;
  - Grau de precisão – declarações do grau de precisão dos resultados obtidos por meio da execução do componente;
  - Interoperabilidade – declarações sobre a interoperabilidade do componente, ou seja, da compatibilidade dos dados tratados pelo componente;
  - Conformidade – declarações de conformidade das funcionalidades do componente com padrões, normas e convenções;
  - Valores-limites – declarações de valores limites, caso o uso do componente seja limitado por eles;
  - Segurança de acesso – declarações de maneiras (se fornecidas) de evitar o acesso não autorizado ao componente e a seus dados.
- Declaração da Confiabilidade – capacidade de verificar se a descrição do componente contém informações sobre procedimentos para a preservação dos dados.

Os atributos avaliados são:

- Cópia de segurança – declarações sobre a realização de cópias de segurança;
  - Verificação de entradas – declarações sobre entradas aceitáveis;
  - Recuperação de erros – declarações sobre a capacidade do componente de se recuperar de erros ocorridos.
- Declaração da Usabilidade – capacidade de verificar se, na descrição do componente, existem declarações sobre o conhecimento específico requerido do usuário do componente, adaptações que podem ser feitas pelo usuário e se proteções contra infrações a direitos autorais podem interferir na usabilidade.

Os atributos avaliados são:

- Interfaces – declaração dos tipos de interfaces do componente;

- Conhecimento necessário – declaração sobre conhecimentos requeridos pelo usuário do componente;
- Adaptação – declarações sobre ferramentas, caso existam, interfaces de gerências ou propriedades do componente para configurações do componente às necessidades do usuário;
- Proteção técnica – declarações sobre proteção contra infrações a direitos autorais, caso estes dificultem a usabilidade.
- Declaração da Eficiência – capacidade de verificar se a descrição do componente inclui dados sobre o comportamento do componente em relação ao tempo. Por exemplo: tempo de resposta.

Os atributos avaliados são:

- Tempo – declarações sobre o comportamento do componente em relação ao tempo;
- Recursos – declarações sobre o comportamento do componente em relação aos recursos.
- Declaração da Manutenibilidade – capacidade de verificar se, na descrição do componente, existem declarações sobre manutenibilidade.

Os atributos avaliados são:

- Análises – declarações sobre a analisabilidade do componente, ou seja, declarações sobre possíveis diagnósticos que podem ser realizados no componente;
- Modificações – declarações sobre a capacidade de modificar o componente;
- Testes – declarações da capacidade de realizar testes para validar o componente após modificações.
- Declaração da Portabilidade – capacidade de verificar se, na descrição do componente, existem declarações sobre portabilidade.

Os atributos avaliados são:

- Adaptações – declarações sobre a capacidade de realizar adaptações no componente para diferentes ambientes;
- Reúso – declarações sobre a capacidade de o componente ser utilizado em diferentes contextos e domínios.
- Declaração da Qualidade em Uso – capacidade de verificar se a descrição do componente inclui declarações de qualidade sob a perspectiva do desenvolvedor, projetista, arquiteto de



software. Ou seja, se há uma declaração de usos anteriores do componente com respeito a eficácia, produtividade, segurança e satisfação.

Os atributos avaliados são:

- Eficácia no DBC – declarações sobre a eficácia atingida com a utilização do componente em contextos de usos especificados;
- Produtividade no DBC – declarações sobre a produtividade atingida com a utilização do componente em contextos de usos especificados;
- Segurança no DBC – declarações sobre níveis aceitáveis de riscos com a utilização do componente em contextos de usos especificados;
- Satisfação no DBC – declarações sobre a satisfação dos usuários (desenvolvedores e integradores) com a utilização do componente em contextos de usos especificados.

Como Conteúdo, entende-se a capacidade de verificar se o conteúdo da descrição do componente é suficientemente adequado, livre de inconsistências internas, e se cada declaração da descrição do componente está correta e passível de teste. O atributo Conteúdo se decompõe nos seguintes requisitos:

- Avaliação da adequação – avaliar se o conteúdo da descrição do componente é inteligível, completa e possui boa organização e apresentação;
- Consistência – avaliar se o conteúdo da descrição do componente é livre de inconsistências internas;
- Termos técnicos – avaliar se os termos técnicos possuem significados únicos;
- Declarações corretas – avaliar se a descrição do componente possui declarações corretas;
- Declarações testáveis – avaliar se a descrição do componente possui declarações passíveis de serem testadas.

#### **4.6.2 Documentação do usuário**

Para a documentação do usuário do Desenvolvimento Baseado em Componentes - DBC, o modelo de qualidade se decompõe nas seguintes características e suas respectivas subcaracterísticas:

- Funcionalidade – essa característica diz respeito às descrições das funcionalidades do componente na documentação para o DBC. Exemplo de funcionalidades do componente: serviços oferecidos e requeridos, documento de implantação, valores-limites etc.

- Completitude – capacidade de verificar quão completa é a documentação das funcionalidades do componente. Para isso, é avaliada a descrição dos serviços oferecidos e requeridos do componente, é verificado se os valores-limites citados na descrição do componente são informados também na documentação e, também, se está incluído na documentação o manual de implantação do componente.

Os atributos avaliados são:

- Descrição dos serviços – verificar a completitude da descrição dos serviços oferecidos e requeridos do componente por meio de suas interfaces providas e requeridas;
  - Valores-limites – verificar se os valores-limites citados na descrição do componente estão declarados também na documentação;
  - Manual de adaptação e integração – verificar se está incluído na documentação o manual de implantação do componente.
- Usabilidade – essa característica diz respeito à usabilidade das informações contidas na documentação. Para isso, é verificado se as informações são corretas, consistentes, de fácil compreensão pela classe de usuários da documentação. Também são verificadas a apresentação e a organização das informações.
  - Correção – capacidade de verificar se as informações na documentação para DBC estão corretas, sem ambiguidades nem erros. Para isso é verificado, nas informações do documento, se há termos ambíguos ou incorretos dentro do contexto indicado.

O atributo avaliado é:

- Informações corretas – verificar se, nas informações da documentação, há termos ambíguos ou incorretos dentro do contexto indicado.
- Consistência – capacidade de verificar se o documento para DBC não apresenta contradições internas ou e com a descrição do componente. Para isso, é avaliada a consistência dos termos que devem ter um significado único em toda a documentação, e também a consistência entre as informações da documentação e as da descrição do componente.

Os atributos avaliados são:

- Consistência dos termos – verificar se os termos apresentam um significado único em toda a documentação;
- Consistência com a descrição do componente – verificar a consistência entre as informações da documentação e as da descrição do componente.

- Inteligibilidade – capacidade de verificar se o documento para DBC é de boa compreensão pela classe de usuários que normalmente utilizam o componente em suas tarefas, como projetistas, desenvolvedores, arquitetos de software etc. Para verificar a inteligibilidade, utiliza-se uma seleção apropriada de termos, exibições gráficas e explicações detalhadas.

Os atributos avaliados são:

- Termos adequados – verificar se os termos utilizados são apropriados para o contexto do componente;
- Exibições gráficas – verificar se o documento possui exibições gráficas, como ilustrações, figuras, para facilitar a compreensão;
- Explicações detalhadas – verificar se o documento possui explicações em um nível de detalhes que facilite a compreensão;
- Clareza – verificar se as informações apresentadas na documentação são claras, de fácil entendimento.
- Apresentação e Organização – capacidade de verificar se a documentação possui uma boa apresentação e organização, de tal modo que quaisquer relacionamentos sejam facilmente identificados. Para isso, é verificado se há presença de índices analíticos e remissivos e se possui uma boa apresentação visual e de layout.

Os atributos avaliados são:

- Presença de índices – verificar se há presença de índices analíticos e remissivos na documentação;
- Apresentação visual – verificar se há uma boa apresentação visual; por exemplo, tamanhos e tipos de fontes, cores e layout.

#### **4.6.3 Componentes no DBC**

Para o Componente em si, o modelo de qualidade se decompõe nas seguintes características e suas respectivas subcaracterísticas:

- Funcionalidade – capacidade do componente de prover serviços que satisfaçam as necessidades especificadas, quando o componente for usado sob condições específicas.
- Adequação – capacidade do componente de prover um conjunto apropriado de serviços para tarefas e objetivos do usuário especificados. Para isso, é verificada sua adequação com uma determinada arquitetura, se a implementação cobriu os serviços oferecidos pelas interfaces do

componente, se há excesso de interfaces para desempenhar os serviços oferecidos pelo componente e se o componente é autocontido.

Os atributos avaliados são:

- Adequação arquitetural – verificar a adequação do componente com a arquitetura especificada;
- Cobertura – verificar se a implementação do componente cobriu os serviços oferecidos pelas suas interfaces;
- Excesso – verificar se há excesso de interfaces no componente, para desempenhar os serviços oferecidos por ele;
- Autocontido – verificar se o componente realiza, de forma completa, a função que ele desempenha, ou seja, as funcionalidades devem estar adequadas para o componente atender seu próprio propósito.
- Interoperabilidade – capacidade do componente de interagir com um ou mais componentes ou sistemas especificados. Para isso, é verificada a compatibilidade dos dados controlados pelo componente: se está em conformidade com padrões ou convenções (ex.: ASNI, XML etc.).

O atributo avaliado é:

- Compatibilidade de dados – verificar se os dados controlados pelo componente estão em conformidade com padrões ou convenções.
- Conformidade – capacidade do componente de estar de acordo com padrões, convenções ou regulamentações previstas em leis e prescrições similares relacionadas à sua funcionalidade. Para avaliar a conformidade, é verificado se o componente está de acordo com padrões internacionais ou regulamentações previstas em leis.

O atributo avaliado é:

- Padronização – verificar se o componente está de acordo com padrões internacionais ou regulamentações previstas em leis.
- Confiabilidade – capacidade do componente de manter um nível de desempenho especificado, quando usado em condições especificadas.
- Maturidade – capacidade do componente de evitar falhas decorrentes de defeitos no software. Para avaliar a maturidade, é verificada a volatilidade das versões, ou seja, o tempo de vida de uma versão no mercado, a evolução do componente de acordo com o número de suas versões que têm sido comercializadas e pelo número de remoção de falhas de uma versão para outra.

Os atributos avaliados são:

- Volatilidade – verificar qual o tempo de vida de uma versão no mercado;
  - Evolução entre versões – verificar a evolução do componente de acordo com o número de suas versões que têm sido comercializadas;
  - Remoção de falhas – verificar o número de remoção de falhas de uma versão para outra.
- Usabilidade – capacidade do componente de ser compreendido, aprendido, utilizado e atrativo para o usuário, quando usado sob condições especificadas.
  - Operacionalidade – capacidade do componente de possibilitar ao usuário operá-lo e controlá-lo. Para isso, é avaliada a clareza com que as interfaces são declaradas, a taxa de complexidade para utilizar suas interfaces e o esforço necessário para operar o componente por meio da configuração de seus parâmetros. Os componentes são operados e controlados pelas suas interfaces (por exemplo, interfaces de gerência, providas, requeridas e interfaces de controle).

Os atributos avaliados são:

- Clareza das interfaces – verificar o nível de clareza com que as interfaces são declaradas;
  - Complexidade – verificar a taxa de complexidade para utilizar as interfaces do componente;
  - Esforço de operação – esforço necessário para operar o componente por meio da configuração de seus parâmetros.
- Manutenibilidade – capacidade do componente de ser modificado. As modificações podem incluir correções, melhorias ou adaptações do componente, devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais.
  - Modificabilidade – capacidade do componente de permitir que uma modificação especificada seja implementada. Para isso, é verificada nesta subcaracterística a adaptabilidade do componente, ou seja, o número de parâmetros customizáveis que o componente oferece, esforço de customização, ou seja, a quantidade de parâmetros que o componente oferece para customizá-lo com relação às suas interfaces oferecidas; também a habilidade de controle de mudança, ou seja, a facilidade do usuário de identificar a versão atual do componente.

Os atributos avaliados são:

- Adaptabilidade – verificar o número de parâmetros customizáveis que o componente oferece;
- Taxa de adaptabilidade – verificar a quantidade de parâmetros que o componente oferece para customizá-lo com relação às suas interfaces oferecidas;

- Capacidade de controle de mudança – verificar a facilidade com que o usuário identifica a versão atual do componente e suas versões compatíveis.
- Testabilidade – capacidade do componente de permitir que ele possa ser validado quando modificado. Para avaliar a testabilidade, é verificado se o componente permite a execução de testes de inicialização, e também se o componente possui conjunto de casos de testes para validar sua funcionalidade.

Os atributos avaliados são:

- Teste de inicialização – verificar se o componente permite a execução de testes de inicialização;
- Pacotes de teste – verificar se o componente possui um conjunto de casos de testes para validar sua funcionalidade.
- Portabilidade – capacidade do componente de ser transferido de um ambiente para outro. O ambiente pode ser organizacional, de hardware ou de software.
- Capacidade para substituir outro componente – capacidade do componente para ser usado em substituição a outro componente especificado, com o mesmo propósito e mesmo ambiente. A compatibilidade entre as versões do componente é o atributo verificado para avaliar esta subcaracterística.

O atributo avaliado é:

- Compatibilidade entre versões – verificar se o componente possui versões compatíveis com sua versão original.

#### **4.6.4 Componente integrado no SBC**

Para o Componente integrado no Sistema Baseado em Componentes - SBC, o modelo de qualidade se decompõe nas seguintes características e suas respectivas subcaracterísticas:

- Funcionalidade – capacidade do componente de prover serviços que satisfaçam as necessidades especificadas, quando o componente for usado sob condições específicas.
- Acurácia – capacidade do componente de prover, com o grau de precisão necessário, resultados ou efeitos corretos, ou conforme acordados. Para verificar a acurácia, é avaliada, nos componentes, a precisão dos resultados obtidos com o que foi especificado pelo requisito do usuário.

O atributo avaliado é:

- Precisão – verificar a precisão dos resultados obtidos com o que foi especificado pelo requisito do usuário.
- Segurança de acesso – capacidade do componente de proteger informações e dados, de forma que pessoas, componentes ou sistemas não autorizados não possam lê-los nem modificá-los, e que não seja negado o acesso às pessoas, ou componentes, ou sistemas autorizados. Para isso, é avaliada a capacidade do componente de criptografar suas informações, controlar o acesso aos seus serviços providos e verificar se o componente implementa algum mecanismo de auditoria, com capacidade para registrar os acessos dos usuários em suas funcionalidades e em seus dados.

Os atributos avaliados são:

- Criptografia dos dados – verificar a capacidade do componente de criptografar suas informações;
  - Capacidade de controle – verificar se o componente é capaz de controlar o acesso aos seus serviços providos;
  - Capacidade para auditar – verificar se o componente implementa algum mecanismo de auditoria, com capacidade para registrar os acessos dos usuários em suas funcionalidades e em seus dados.
- Confiabilidade – capacidade do componente de manter um nível de desempenho especificado, quando usado em condições especificadas.
  - Recuperabilidade – capacidade do componente de restabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados, no caso de uma falha. Para isso, é avaliada a habilidade do componente em serializar seus dados e estado; assim, ele pode ser transferido para uma máquina diferente ou armazenado para persistência. É avaliada, também, a capacidade do componente de armazenar seu estado para uma futura recuperação; também se o componente provê qualquer interface para implementar transações com suas operações, e se o componente pode controlar situações de erros, por meio de mecanismos implementados, como, por exemplo: exceções.

Os atributos avaliados são:

- Serializabilidade – verificar se o componente é capaz de serializar seus dados e estado de modo que ele possa ser transferido para diferentes máquinas ou armazenado para persistência;
- Persistência – verificar se o componente possui a capacidade de armazenar seu estado para uma futura recuperação;

- Transacionalidade – verificar se o componente provê qualquer interface para implementar transações com suas operações;
- Tratamento de erros – verificar se o componente pode controlar situações de erros por meio de mecanismos implementados, por exemplo: exceções.
- Eficiência – capacidade do componente de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob condições especificadas (recurso diz respeito à configuração de hardware e software).
- Em relação ao tempo – capacidade do componente de fornecer tempos de resposta e de processamento, além de taxas de transferência, apropriados, quando o componente executa suas funções, sob condições estabelecidas. Para isso, é avaliado o tempo de resposta que o componente gasta, desde a requisição que é recebida até a resposta que ele envia. Também é avaliada a saída que pode ser produzida com sucesso, em um dado período de tempo, e também é medida a soma de entrada de informação que pode ser processada com sucesso, pelo componente, em um dado período de tempo.

Os atributos avaliados são:

- Tempo de resposta – verificar o tempo que o componente gasta, desde a requisição que é recebida até a resposta que ele envia;
- Capacidade de emissão – verificar a saída que pode ser produzida com sucesso, pelo componente, em um dado período de tempo;
- Capacidade de recepção – verificar a soma de entrada de informação que pode ser processada com sucesso, pelo componente, em um dado período de tempo.
- Em relação à utilização de recursos – capacidade do componente de usar tipos e quantidades apropriados de recursos, quando o componente executa suas funções sob condições estabelecidas. Para isso, é avaliada a soma de memória necessária para o componente operar, e o espaço em disco utilizado pelo componente, incluindo o espaço utilizado para armazenar seu código e partes constituintes e o espaço utilizado, temporária ou permanentemente, durante a execução.

Os atributos avaliados são:

- Memória requerida – verificar a soma de memória necessária para o componente operar;
- Utilização de disco – verificar o espaço em disco utilizado pelo componente, incluindo o espaço utilizado para armazenar seu código e partes constituintes e o espaço utilizado, temporariamente ou permanentemente, durante a execução.



O modelo de qualidade para componente de software foi objeto de um projeto FINEP - Financiadora de Estudos e Projetos -, desenvolvido pela equipe do CTI, e deverá ser aplicado em ambiente produtivo para consolidação da sua adequação.

Requisitos de qualidade e o processo de avaliação da qualidade serão assuntos do próximo capítulo. Os requisitos de qualidade serão abordados para COTS, usabilidade na interface e documentação de usuário.

## CAPÍTULO 5

### **Requisitos e avaliação da qualidade de software**

Com estudos e pesquisas bibliográficas sobre o tema qualidade de produto de software e a identificação e exploração da necessidade de mercado, surge uma visão abrangente e profunda sobre o que deve ser a avaliação da qualidade de produto de software.

Assim, em busca de experiência de organizações nacionais e internacionais e de contatos com entidades nacionais representativas de desenvolvedores de software, tais como: ASSESPRO – Associação das Empresas Brasileiras de Tecnologia da Informação, Software e Internet e SOFTEX – Associação para Promoção da Excelência do Software Brasileiro, surge uma visão de como implementar e aplicar um processo de avaliação da qualidade de produto de software, com o objetivo de verificar a qualidade deste em relação a requisitos de padrões reconhecidos internacionalmente.

Este capítulo apresenta normas específicas para executar uma avaliação. Essas normas são relativamente recentes; a mais antiga, internacionalmente, é de 1998, e as nacionais estão disponíveis desde 2002. Apresenta, também, normas específicas para ajudar na especificação de requisitos de qualidade, sendo elas para documentação de usuário, pacotes de software e usabilidade na interface.

Existem ainda poucos trabalhos práticos de avaliação de produto de software de âmbito internacional, mas indicam uma tendência na área e mostram a utilização das normas abordadas neste trabalho. É uma área de atuação muito recente, tanto da pesquisa como da aplicação.

### **5.1 Processo de avaliação da qualidade de produto de software**

Um exame sistemático exige um processo de avaliação que seja responsável por fornecer passos a serem seguidos por quem for avaliar a qualidade do produto de software. Um processo de avaliação para esse exame sistemático pode ser encontrado na série NBR ISO/IEC 14598, que está inclusa na série 25000 e em revisão na parte 2504n – Divisão de Avaliação da Qualidade.

Essa série de normas oferece uma visão geral dos processos de avaliação de produtos de software e fornece guias e requisitos para avaliação. Nessa série existe um processo de avaliação genérico, definido na parte 1, e pode ser particularizado em outras três situações diferentes, para a avaliação da qualidade de produto, focando os processos para desenvolvedores, compradores e avaliadores, respectivamente as partes 3, 4 e 5 dessa série.

O público-alvo da série NBR ISO/IEC 14598 são desenvolvedores, adquirentes e avaliadores independentes, particularmente aqueles que são responsáveis por avaliações de produtos de software.

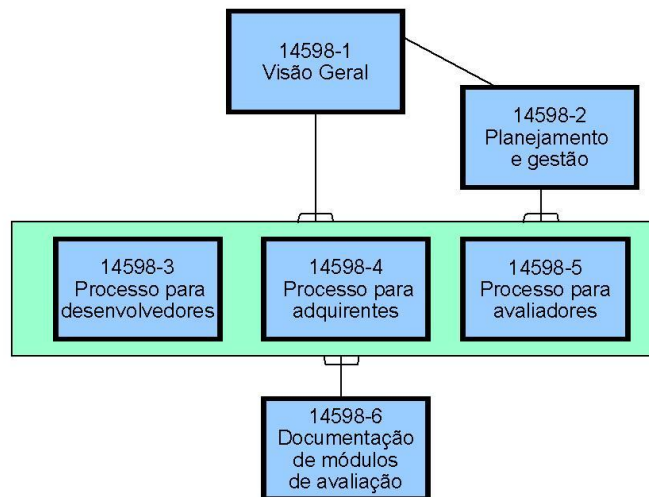
Os resultados de avaliações obtidos a partir da aplicação da série NBR ISO/IEC 14598 podem ser utilizados por gerentes, desenvolvedores e mantenedores, para medir a aderência aos requisitos e para fazer melhorias onde necessárias. Esses resultados também podem ser utilizados por analistas, para estabelecer o relacionamento entre medidas internas e externas.

Resumidamente, a série 14598 está organizada da seguinte forma:

- Parte 1 – Visão geral do processo de avaliação de produto em geral. Também explica o relacionamento entre a série NBR ISO/IEC 14598 e o modelo de qualidade apresentado na norma NBR ISO/IEC 9126-1. Esta parte define os termos técnicos utilizados nas demais partes, contém requisitos gerais para especificação e avaliação de qualidade de software e esclarece os conceitos gerais. Adicionalmente, ela também fornece uma estrutura para avaliar a qualidade de quaisquer produtos de software e estabelece os requisitos para métodos de medição e avaliação de produtos de software.
- Partes 2 e 6 – Apoio ao processo de avaliação de produto. Estão relacionadas ao suporte e gestão da avaliação corporativa ou departamental. Fornecem exemplos de medidas que correspondem a uma subcaracterística. É difícil utilizar essas medidas de maneira consistente em uma organização, podendo ser necessário desenvolver novas medidas para uso específico. Entretanto, é necessário que uma função de apoio na organização especifique cada medida para o uso correto e consistente dentro da organização. Convém que se estabeleça o formato para documentar uma medida e métodos associados, assim como guias para uso. O conceito para um módulo de avaliação fornece uma solução para essa necessidade. A parte 6 trata da documentação de módulos de avaliação.
- Partes 3, 4 e 5 – Requisitos e orientações para o processo de avaliação de projeto, em três situações: desenvolvimento; aquisição e avaliação independente, incluindo avaliação de terceira parte.

A relação entre as normas dessa série pode ser visualizada na Figura 5.1. Cada processo de avaliação (partes 3, 4 e 5) pode ser usado em conjunto com o suporte à avaliação (partes 2 e 6).

Esta figura se refere à série 14598, que está em processo de revisão para a nova série. Todas as partes estarão compondo a divisão 2504n, sendo que especificamente a norma NBR ISO/IEC 14598-2 – Planejamento e Gestão da Avaliação irá compor a divisão 2500n – Divisão de Gestão da Qualidade.



**Figura 5.1 – Relacionamento entre as partes da ISO/IEC 14598.**

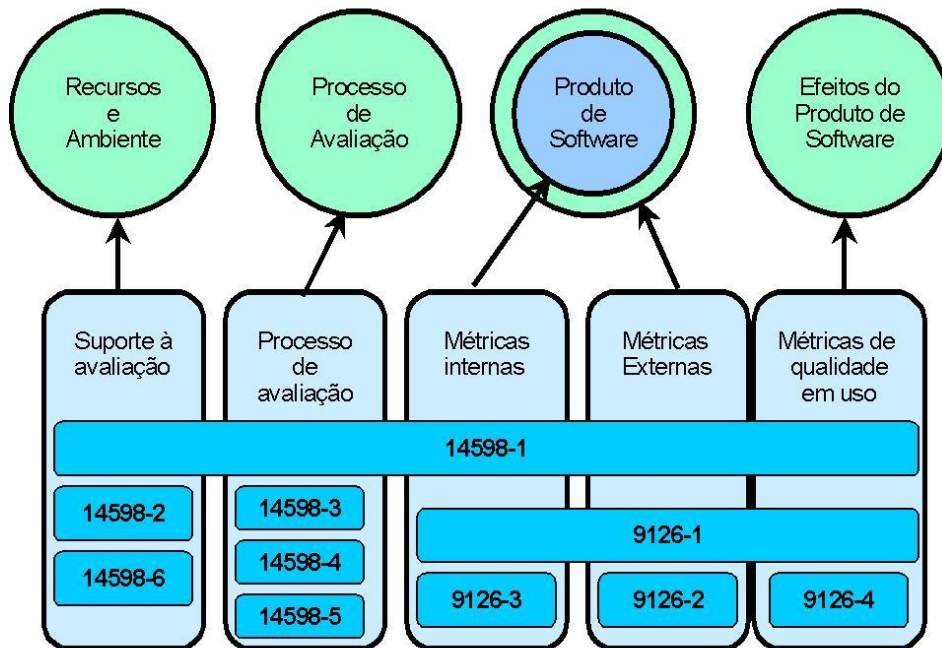
A elaboração da série de normas NBR ISO/IEC14598 consolidou-se na estrutura mostrada na Tabela 5.1. Estudos e revisões continuam sendo desenvolvidos de forma sistemática, e as normas são revisadas e evoluídas.

**Tabela 5.1– Normas da série ISO/IEC 14598: “Status” em 2009.**

<b>Norma</b>	<b>Título resumido</b>	<b>Assunto</b>	<b>Estado Internacional</b>	<b>Estado Nacional</b>
14598-1	Avaliação de Produto de Software – Parte 1: Visão Geral	Visão geral da estruturação dessa série de Normas e dos processos de avaliação	Norma, publicada em 1999	Norma, publicada em 2002
14598-2	Avaliação de Produto de Software – Parte 2: Planejamento e Gerenciamento	Atividades de planejamento e gerenciamento do processo de avaliação.	Norma, publicada em 2000	Norma, publicada em 2003
14598-3	Avaliação de Produto de Software – Parte 3: Processo para a equipe de desenvolvimento	Atividades de avaliação durante o processo de desenvolvimento de software.	Norma, publicada em 2000	Norma, publicada em 2002
14598-4	Avaliação de Produto de Software – Parte 4: Processo para o comprador	Atividades de avaliação no processo de seleção para aquisição de software.	Norma, publicada em 1999	Norma, publicada em 2003
14598-5	Avaliação de Produto de Software – Parte 5: Processo para o Avaliador	Processo de avaliação, com definição das atividades, incluindo relações entre avaliador e requisitante.	Norma, publicada em 1998	Norma, publicada em 2002
14598-6	Avaliação de Produto de Software – Parte 6: Módulos de Avaliação	Definição da estrutura de Módulos de Avaliação	Norma, publicada em 2001	Norma, publicada em 2004

As normas das séries 9126 e 14598 podem ser utilizadas em complementação, umas às outras, de acordo com o objetivo da avaliação.

A norma NBR ISO/IEC 14598-1 contém conceitos de como avaliar a qualidade de software e define um modelo de processo de avaliação genérico; junto com as normas NBR ISO/IEC 14598-2 e NBR ISO/IEC 14598-6, estabelecem itens necessários para o suporte à avaliação; e junto com as normas NBR ISO/IEC 14598-3, NBR ISO/IEC 14598-4 e NBR ISO/IEC 14598-5, estabelecem processo de avaliação específico para desenvolvedores, adquirentes e avaliadores de software, respectivamente. O relacionamento entre elas pode ser visto na Figura 5.2.



**Figura 5.2 – Relacionamento entre as séries 9126 e 14598.**

A norma NBR ISO/IEC 14598-3, que trata do processo de avaliação para desenvolvedores de software, aplica-se a todo software e em todas as fases do ciclo de vida de desenvolvimento. Enfoca a seleção de indicadores, para prever a qualidade do produto final pela medição da qualidade de produtos intermediários. Também enfoca a medição da qualidade do produto final. Uma organização deve rever todos os requisitos e recomendações indicados na norma:

- Requisitos organizacionais – infraestrutura que permita a obtenção de dados e modificações nos processos baseados na análise dos dados.
- Requisitos de projeto – processo de desenvolvimento disciplinado, que permita o planejamento e a condução das medições do software e sua avaliação; coordenar as atividades de avaliação com

suporte; modelo de desenvolvimento similar a projetos anteriores, na sua organização de desenvolvimento com mesmo conjunto de atributos, deve também ser aplicado nos projetos, para permitir a análise dos dados.

- Retroalimentação para a organização – muitos métodos de análise de dados requerem dados de projetos desenvolvidos anteriormente, sob condições similares e com requisitos de qualidade comparáveis. A utilização de dados obtidos em outras experiências é recurso para melhorar o processo de avaliação e os módulos de avaliação.

O desenvolvedor deve:

- tornar os dados coletados disponíveis para a organização, para seu uso em outros projetos de desenvolvimento;
- rever os resultados da avaliação e a validade do processo de avaliação, indicadores e medidas aplicadas;
- melhorar os módulos de avaliação, incluir a coleta de dados sobre indicadores extras, de maneira a validá-los para uso posterior, quando necessário.

Convém que seja aplicado um modelo de desenvolvimento similar ao que foi utilizado em projetos anteriores, na sua organização; o mesmo conjunto de atributos seja aplicado nos projetos, de maneira a permitir a análise dos dados.

A norma NBR ISO/IEC 14598-4, que trata do processo de avaliação para Adquirentes de software, deve ser usada na aceitação ou seleção de um produto. A norma contém requisitos e orientações para avaliação da qualidade de produto durante a aquisição de produtos de prateleira, produtos de software de uso geral ou modificações em produtos de software existentes.

O processo de avaliação nela descrito também auxilia a estabelecer regras de decisão sobre a aceitação de um único produto, ou selecionar um produto entre produtos alternativos.

A seguir será abordado especificamente o processo de avaliação de qualidade para produto de software genérico; em seguida, o processo de avaliação contendo diretrizes sugeridas para avaliadores de qualidade de produto de software.

#### **5.1.1 Norma NBR ISO/IEC 14598-1 e NBR ISO/IEC 14598-5**

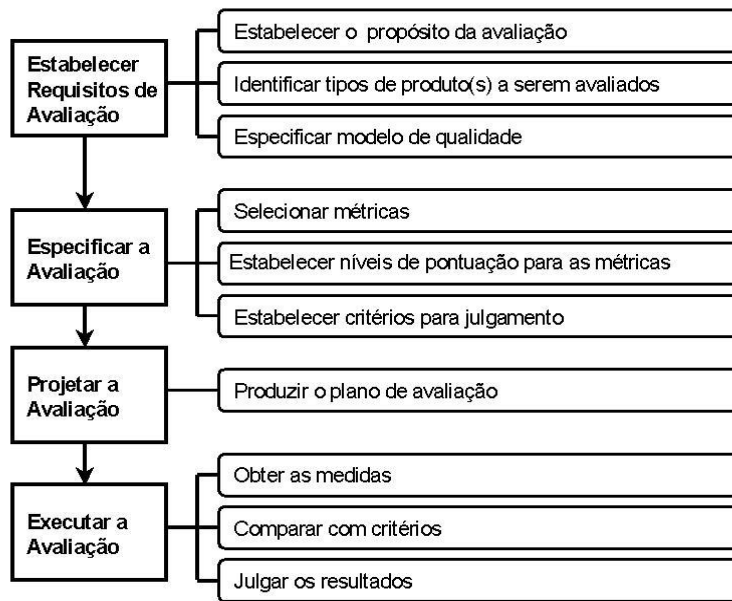
A norma NBR ISO/IEC 14598-1 fornece requisitos e recomendações para implementação prática da avaliação de produto de software. O processo de avaliação proposto pode ser usado para avaliar produtos já existentes ou produtos intermediários, isto é, em desenvolvimento. Pode ser utilizada por laboratórios de avaliação, fornecedores de software, compradores de software, usuários e entidades certificadoras, cada qual com seu objetivo.

Em termos de características de qualidade, pode ser usada a norma NBR ISO/IEC 9126-1, por ser uma norma reconhecida internacionalmente. Entretanto, o processo de mensurar diretamente essas características não é prático; é necessário o desdobramento dessas características em atributos que possam ser medidos e pontuados. Os atributos devem ser os mais objetivos possíveis, para que a avaliação não sofra interferência da opinião do avaliador.

Para que a subjetividade da avaliação seja mínima, é necessário que o processo de avaliação seja repetível, reproduzível, imparcial e objetivo. Essas características são apresentadas na norma NBR ISO/IEC 14598-5, como características fundamentais esperadas do processo de avaliação, e são detalhadas a seguir:

- Repetível – a avaliação repetida de um mesmo produto, com mesma especificação de avaliação, realizada pelo mesmo avaliador, deve produzir resultados que podem ser aceitos como idênticos;
- Reproduzível – a avaliação do mesmo produto, com mesma especificação de avaliação, realizada por um avaliador diferente, deve produzir resultados que podem ser aceitos como idênticos;
- Imparcial – a avaliação não deve ser influenciada frente a nenhum resultado particular;
- Objetiva – os resultados da avaliação devem ser factuais, ou seja, não influenciados pelos sentimentos ou opiniões do avaliador.

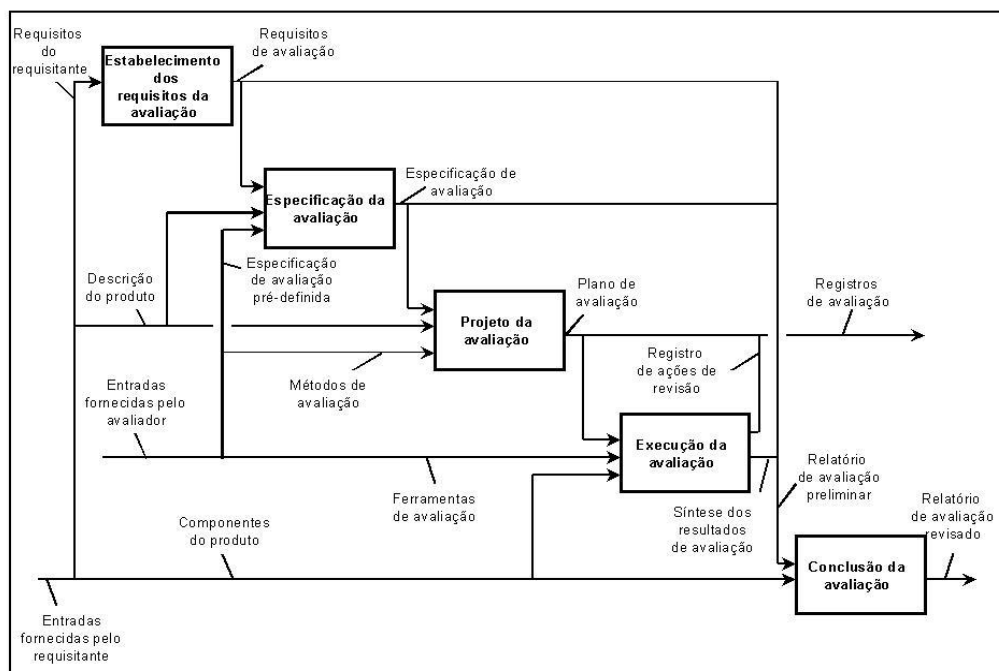
O processo de avaliação proposto pela norma NBR ISO/IEC 14598-1 inclui quatro etapas de avaliação, contendo dez atividades, como mostra a Figura 5.3.



**Figura 5.3 – Processo de avaliação segundo a norma NBR ISO/IEC 14598-1.**

O processo de avaliação proposto pela norma NBR ISO/IEC 14598-5 é semelhante ao da parte 1, incluindo uma etapa a mais para as etapas da avaliação, a saber: análise de requisitos da avaliação, especificação da avaliação, projeto da avaliação, execução da avaliação e conclusão da avaliação, como pode ser visto na Figura 5.4. Nessa figura, são mostradas, adicionalmente, quais as entradas e saídas de cada etapa da avaliação.





**Figura 5.4 – Processo de avaliação segundo a norma NBR ISO/IEC 14598-5.**

A seguir, é detalhada cada uma das atividades:

a) Estabelecimento dos requisitos da avaliação

Nesta etapa, deve-se descrever os objetivos da avaliação. Vários pontos de vista podem ser considerados, dependendo dos diferentes usuários do produto, tais como comprador, fornecedor, desenvolvedor e operador; devem ser descritos os objetivos e os requisitos da avaliação que dependem da necessidade do solicitante da avaliação, ou seja, do objetivo final da avaliação.

Os solicitantes de uma avaliação podem ser:

- 1) Equipes de desenvolvimento, que usam o resultado da avaliação para identificar ações corretivas e determinar estratégias de evolução;
- 2) Vendedores, que usam a qualidade como marketing;
- 3) Compradores, para avaliar os produtos que estão competindo no mercado;
- 4) Usuários, para obter confiança no produto;

- 5) Comunidade de software, para identificar e validar os métodos mais adequados de avaliação e, assim, aumentar a credibilidade das técnicas;
- 6) Laboratórios de avaliação, para desenvolver uma abordagem de avaliação mais consistente.

A análise de requisitos visa definir os requisitos de qualidade do produto. Deve partir dos objetivos da avaliação que traduzem diretamente o interesse do requisitante da avaliação. A análise define a profundidade, a abrangência, o relacionamento da presente avaliação com outras, e os objetos a serem avaliados.

b) Especificação da avaliação

Nesta etapa, deve-se definir o escopo da avaliação e as medidas a serem executadas no produto submetido à avaliação, nos seus vários componentes. O nível de detalhes na especificação da avaliação deverá ser tal que, na sua base, a avaliação seja repetível e reproduzível.

c) Projeto da avaliação

Nesta etapa, devem ser documentados os procedimentos a serem usados pelo avaliador, para executar as medidas especificadas na fase anterior. O avaliador deve produzir um Plano de Avaliação, que descreve os recursos necessários para executar a avaliação especificada, assim como a distribuição desses recursos nas várias ações a serem executadas.

d) Execução da avaliação

Nesta etapa, devem ser obtidos resultados da execução de ações para medir e verificar o produto de software, de acordo com os requisitos de avaliação, como descrito na especificação da avaliação e como planejado no plano de avaliação. Ao executar essas ações, tem-se o rascunho do relatório de avaliação e os registros da avaliação.

e) Conclusão da avaliação

Nesta etapa, deve-se revisar o relatório da avaliação e disponibilizar os dados resultantes da mesma para o requisitante da avaliação.

Para uma avaliação profissional, esses dados são sigilosos e exclusivos ao requisitante da avaliação, qualquer publicação deles é de sua inteira responsabilidade.

No processo de avaliação, a identificação do produto a ser avaliado é ainda preliminar. No decorrer das etapas do processo, mais informações são obtidas, o que contribui para uma melhor identificação do produto a ser avaliado. Dúvidas que poderão ocorrer, dependendo da fase em que se encontrar a avaliação:

- Quando se trata de produto final, de acordo com o escopo da avaliação, poderá ser selecionado todo o produto de software, ou eventualmente apenas alguns de seus componentes. A definição

ocorrerá quando, no mínimo, os requisitos básicos de qualidade estiverem definidos. Será então necessário voltar a esta fase de definição de produtos para a sua complementação.

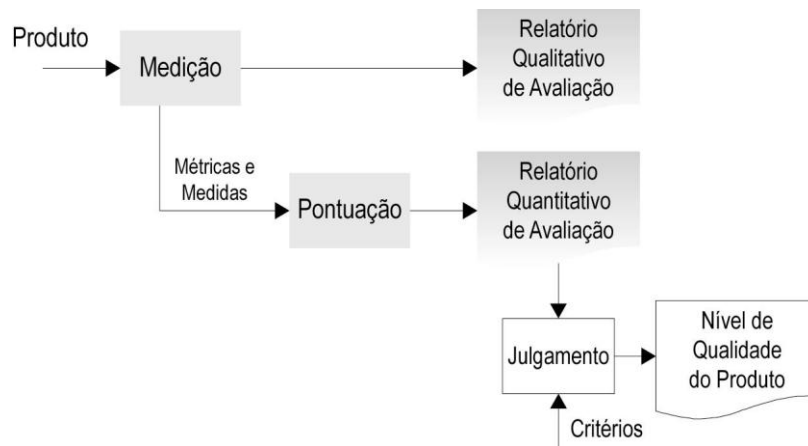
Um fator que pode ser determinante na seleção dos componentes a serem avaliados é a disponibilidade de métodos de avaliação na organização que irá realizar a avaliação. Por exemplo, suponha-se que um propósito de avaliação seja a escolha entre alguns produtos de mercado e que um dos requisitos de qualidade para essa escolha seja segurança de acesso a dados. Suponha-se, também, que a organização não disponha de métodos de avaliação desse requisito de qualidade. A não disponibilidade poderá determinar que os componentes do produto que tratam especificamente de segurança de acesso sejam desconsiderados para efeito de avaliação, ou que esse requisito, caso seja muito importante, tenha que ser avaliado por outra organização.

- Para o produto intermediário, a definição de qual produto intermediário será avaliado é mais complexa, pois depende, em primeiro, lugar do ciclo de vida de desenvolvimento adotado pela organização e do estágio em que se encontram seus respectivos produtos. Além disso, deve-se considerar que as medidas internas devem ser escolhidas de modo a refletir a futura qualidade externa do produto. Assim sendo, é necessário conhecer os requisitos externos, para então definir que medidas internas são aplicáveis aos produtos intermediários, de modo a obter uma avaliação efetiva.

As primeiras vezes que essas definições de produtos intermediários para avaliação são feitas, não se constituem em um trabalho simples. Porém, em termos práticos, a partir da existência de um histórico de medidas aplicadas na organização, é provável que exista uma referência empírica a ser considerada, tanto para a seleção de medidas como para identificação dos produtos a serem avaliados. E, conforme já mencionado, a escolha inicial tende a ser refinada nas demais fases de avaliação.

## **5.2 Especificação da avaliação**

A fase de especificação da avaliação de um produto de software, neste contexto, traz consigo três importantes conceitos: medição, pontuação e julgamento, como mostra a Figura 5.5.



**Figura 5.5 – Conceitos num processo de avaliação.**

- Medição é a ação de aplicar uma medida de qualidade de software a um produto de software específico.
- Pontuação é a ação de transformar a medida obtida em um sistema numérico preestabelecido.
- Julgamento é a emissão de um juízo sobre a qualidade do produto de software.

Por meio da medição de um produto de software, é possível conhecer qualitativamente o nível de qualidade; após a pontuação, é possível conhecer quantitativamente o nível de qualidade. Assim, com a aplicação dos critérios de julgamento, é possível fazer o julgamento da qualidade.

Para fazer medição em produto de software, é necessária a utilização de medida de qualidade de software. Esta, por sua vez, precisa de um “Método de avaliação” e de uma “Escala”, previamente escolhidos, que podem ser usados para determinar o valor que uma particularidade recebe em um produto de software específico.

O “Método de avaliação” é um procedimento com que se descrevem as ações a serem realizadas por um avaliador, para obter o resultado da medição ou verificação especificadas, quando aplicadas num produto ou em componentes especificados de um produto de software.

A “Escala” atribui rótulos numéricos aos atributos de uma entidade, resultando a representação da fidedignidade do atributo. Ela é um conjunto de valores com propriedades definidas.

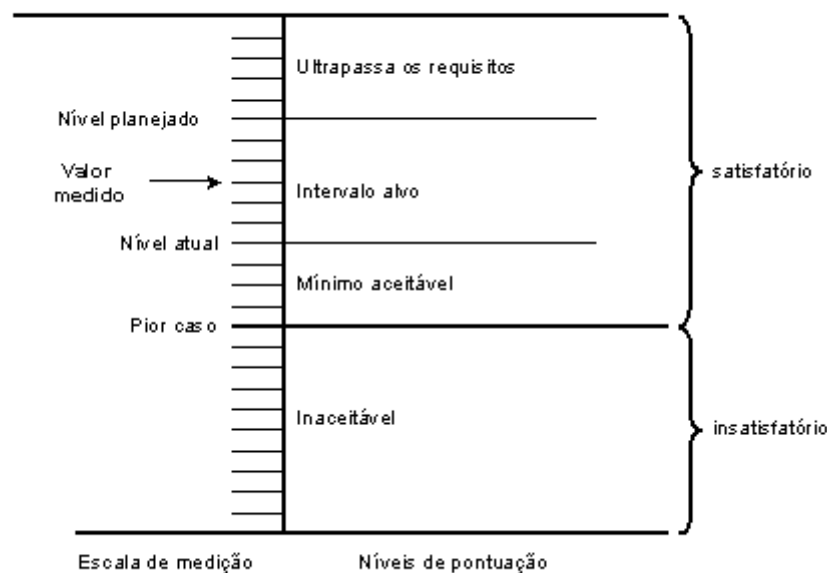
Exemplos de tipos de escalas são:

- Escala nominal – que corresponde a um conjunto de categorias;
- Escala ordinal – que corresponde a um conjunto ordenado de pontos de escala;

- Escala de intervalos – que corresponde a uma escala ordenada, com pontos de escala equidistantes;
- Escala de proporção – que tem pontos de escala equidistantes e também um zero absoluto.

Medidas usando escalas ordinais ou nominais produzem dados qualitativos, e medidas usando escalas de intervalos ou de proporção produzem dados quantitativos.

De acordo com a norma NBR ISO/IEC 14598-1, uma escala pode assumir diferentes níveis de pontuação, como mostra a Figura 5.6. O nível de pontuação deve ter um valor-limite entre o que seja satisfatório e insatisfatório.



**Figura 5.6 – Níveis de pontuação para as medidas** Fonte: NBR ISO/IEC 14598-1 (2002).

A qualidade de uma escala nominal, em sentido abrangente, descreve sua habilidade de refletir veridicamente uma medida obtida. Uma parte significativa desta construção envolve a semântica da escala – os adjetivos, palavras, frases e construção das sentenças, que irão trazer à tona a resposta correta sobre o assunto.

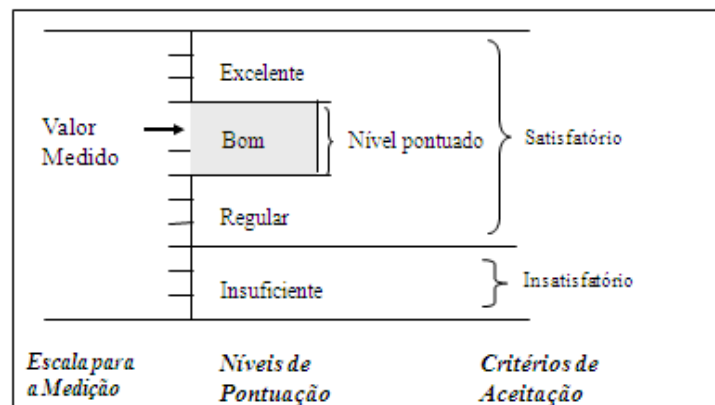
Ao avaliar um produto de software, o avaliador sempre sofre influências, sejam elas advindas de preferências pessoais, de experiências anteriores, do conhecimento da área de aplicação do software. Desse modo, é necessário que as medidas a serem feitas estejam claras e objetivas ao avaliador. A questão da subjetividade do avaliador deve ser eliminada na medida do possível, procurando executar

uma avaliação que seja imparcial e objetiva, com a utilização das diretrizes especificadas na norma NBR ISO/IEC 14598-5.

Para realizar uma avaliação, não basta ter o modelo de qualidade definido por meio das características até os atributos ainda é preciso estabelecer quais serão os critérios de aceitação, as medidas e os níveis de pontuação.

- Critério de Aceitação – predeterminação dos níveis de pontuações considerados satisfatórios, ou não satisfatórios, para um atributo de uma entidade, de acordo com a norma NBR ISO/IEC14598-1, Medida e Nível de Pontuação.
- Medida – Compreende um método e uma escala de medição. Por escala entende-se: um conjunto de valores com propriedades definidas; por medição, a determinação de um valor (que pode ser um número ou uma categoria) para um atributo de uma entidade.
- Nível de pontuação – pontuações de uma escala ordinal, que é utilizada para categorizar uma escala de medição.

A norma NBR ISO/IEC 9126-1 lembra a necessidade do uso de medidas e níveis de pontuação, além de mostrar a relação existente entre eles, conforme mostra a Figura 5.7. No entanto, a norma não os define, pois eles podem variar de acordo com a organização e/ou com a aplicação.



**Figura 5.7– Valor medido e nível de pontuação (NBR ISO/IEC 9126-1, 2001).**

Dois outros fatores devem ser considerados numa avaliação: o grau de importância de cada característica de qualidade e o ponto de vista sob o qual o produto de software está sendo analisado. Esses fatores podem ser apontados como bastante subjetivos, mas é possível, numa avaliação comparativa de diferentes aplicações de software, ponderar as características de qualidade utilizando um sistema de pesos proporcionais a sua importância na área de aplicação do software.

O grau de importância de cada característica de qualidade varia de software para software. Alguns exemplos atestam a variação: a confiabilidade é mais importante para software de um sistema de missão crítica; a eficiência é mais importante para software de sistema em tempo real; a usabilidade é mais importante para software interativo.

A norma aponta três pontos de vista a serem considerados em uma avaliação da qualidade de software: do usuário, da equipe de desenvolvimento e do gerente. Cada um deles tem foco de interesse próprio:

- O usuário está interessado no uso do software, e não nos seus aspectos internos. Ele quer saber se as funções requeridas estão disponíveis no software e quão eficiente, confiável e fácil de usar é o produto.
- A equipe de desenvolvimento está interessada tanto na qualidade dos produtos intermediários como na qualidade do produto final. O processo de desenvolvimento requer que o usuário e a equipe de desenvolvimento utilizem as mesmas características de qualidade de software, uma vez que elas se aplicam tanto para os requisitos como para a aceitação.
- A visão do gerente é comercial. Ele está interessado na qualidade de forma geral, e não em características específicas de qualidade. Sua responsabilidade é otimizar a qualidade, dentro das limitações de custo, recursos humanos e prazos.

Resumidamente, a norma ISO/IEC 9126-1 apresenta um modelo para avaliação da qualidade de produto de software, baseado no desmembramento das seis características de qualidade. Além disso, alerta para a necessidade de se definir, caso a caso, as medidas, níveis de pontuação, critérios de aceitação, grau de importância de cada característica e do ponto de vista a ser considerado na avaliação. Essa abertura permite a aplicação desse modelo nas diversas organizações de software, em diferentes necessidades.

As próximas seções trazem requisitos de qualidade que podem ser medidos e avaliados.

### **5.3 Requisitos para pacotes de software**

Entende-se por pacote de software o “conjunto completo e documentado de programas fornecidos a diversos usuários para uma aplicação ou função genérica”. São exemplos de Pacotes de Software: processadores de texto, planilhas eletrônicas, bancos de dados, software gráficos, programas para funções técnicas ou científicas e programas utilitários, entre outros.

É essa classe específica de produtos de software que é conhecida como pacotes de software, e internacionalmente estabeleceu-se a sigla *COTS – Commercial Off-The-Shelf*, ou seja, produto de software comercial de prateleira.

A norma NBR ISO/IEC 25051 é aplicável à avaliação de pacotes de software na forma em que são oferecidos e liberados para uso no mercado – COTS. Incluem-se como possíveis usuários dessa norma:

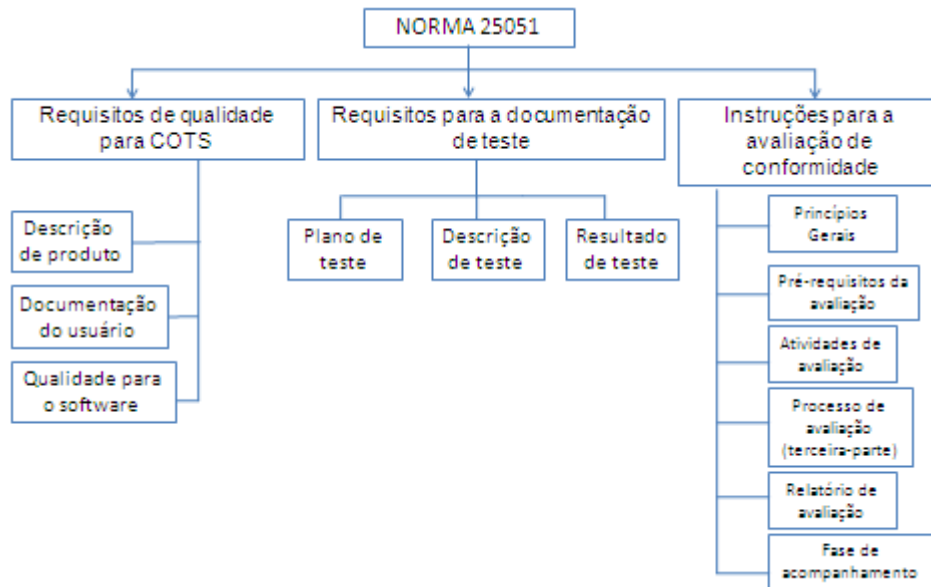
- fornecedores que estejam especificando os requisitos para um pacote de software; projetando um modelo para descrever produtos; julgando seus próprios produtos; emitindo declarações de conformidade; submetendo produtos à certificação ou à obtenção de marcas de conformidade;
- entidades de certificação que pretendam estabelecer um esquema de certificação por terceira parte;
- laboratórios de teste, que terão de seguir as instruções durante a execução de testes para certificação ou para emissão de marca de conformidade;
- laboratórios de avaliação, que executam medição e pontuação de requisitos de qualidade de software;
- entidades de credenciamento que credenciam entidades de certificação e laboratórios de testes e de avaliação;
- auditores, quando julgam a competência de laboratórios de teste e de avaliação;
- compradores que pretendam comparar seus próprios requisitos com os aqui descritos; comparar os requisitos necessários para executar uma determinada tarefa com a informação presente nas descrições de produtos existentes; procurar por produtos certificados; verificar se os requisitos foram atendidos;
- usuários que se pretendam beneficiar com produtos melhores.

Essa norma estabelece os requisitos para pacotes de software, requisitos de documentação de teste, instruções para avaliação de conformidade de um pacote de software com relação aos requisitos preestabelecidos. Alguns produtos de software necessitam de requisitos adicionais: por exemplo, software em que o requisito de segurança seja um fator crítico; por exemplo, software com diferentes níveis de acesso de usuários. Nesses casos, devem ser incluídos outros requisitos importantes para sua determinada classe de aplicação.

Os requisitos de qualidade estabelecidos nessa norma podem também ser utilizados para produtos de software que não sejam especificamente COTS. Isso pode ser feito verificando, entre os requisitos estabelecidos na norma, aqueles que podem ser aplicados para quaisquer produtos de software.

A Figura 5.8 mostra a estrutura básica do conteúdo da norma, relacionando os requisitos de qualidade para pacotes, para documentação de teste e avaliação de conformidade e também as atividades de avaliação de conformidade para um pacote de software.





**Figura 5.8 – Estrutura da norma NBR ISO/IEC 25051.**

A seguir, será apresentado cada um dos elementos dessa estrutura.

### 5.3.1 Requisitos de qualidade para pacotes

É importante ressaltar que os requisitos estabelecidos na norma “podem” ou “devem” estar presentes no pacote de software. As formas verbais “pode” e “deve” são explicitadas a cada requisito estabelecido, as quais podem ser Mandatórias (termo “deve”) ou Recomendáveis (termo “pode”). É importante observar que o uso de um requisito como recomendável está diretamente relacionado com o tipo do produto, ou seja, para alguns tipos de produtos esses requisitos podem ser mandatórios. Por exemplo, um sistema de reserva de passagens aéreas tem como um dos requisitos mandatórios a Confiabilidade; em um sistema de entretenimento, a Confiabilidade não é Mandatória, e sim Recomendável.

Um pacote de software deve possuir toda a documentação do pacote, que é composta pela Descrição de Produto, Documentação do Usuário e Software. A seguir são descritos os requisitos de qualidade de cada um desses componentes.

#### a) Descrição de produto

A descrição de produto é um documento que expõe as principais propriedades de um pacote de software, com os objetivos de:

- Auxiliar o usuário ou os potenciais compradores desse produto, na avaliação da adequação do produto às suas reais necessidades; e

- Servir como base para testes.

Este documento deve estar disponível ao usuário, independentemente da aquisição do produto, por meio de um catálogo, de uma mídia digital e de apresentação ou qualquer outro meio disponível que alcance esse objetivo. A descrição deve ser clara, compreensível e harmônica com os outros documentos associados. A norma propõe aspectos práticos e diretos, indicando “o que” deve conter a descrição. Na Tabela 5.2, estão os itens que compõem os requisitos do documento Descrição de produto; nos requisitos propriamente ditos, há uma descrição observando quais são os mandatórios e os recomendáveis, segundo a norma.

Tabela 5.2 – Requisitos de qualidade para a descrição do produto.

Item	Requisitos
Disponibilidade	Deve estar disponível para adquirentes.
Conteúdo	Deve conter as informações necessárias, não deve conter inconsistências internas, e as informações devem poder ser testadas ou verificadas.
Identificação e indicações	Deve apresentar uma identificação única; o COTS deve ser indicado pelo nome, versão e data; deve conter nome e endereço do fornecedor; deve identificar as tarefas e serviços que podem ser realizados; deve indicar o uso de vários usuários ou para um único usuário em um sistema; devem ser identificadas as interfaces, ou software, se houver referência; deve indicar se o produto COTS depende de um software e/ou hardware; deve indicar se é oferecido suporte; e deve indicar se é oferecida manutenção.
Declarações sobre funcionalidade	Deve conter informações sobre as subcaracterísticas; deve apresentar uma visão geral das funções disponíveis, os valores-limite, se existirem, e os dispositivos de segurança de acesso ao produto, quando necessários.
Declarações sobre confiabilidade	Deve conter informações sobre as subcaracterísticas ; deve indicar a capacidade do software de permanecer em operação e deve apresentar as informações sobre os procedimentos para salvar e recuperar dados.
Declarações sobre usabilidade	Deve conter informações sobre as subcaracterísticas; deve apresentar o tipo de interface com o usuário; se é necessário algum conhecimento técnico específico para o seu uso; se o produto pode ser adaptado às necessidades do usuário; se há proteção técnica contra infração e se há indicações de acessibilidade.
Declarações sobre eficiência	Deve conter informações sobre as subcaracterísticas; pode incluir informações sobre configurações do sistema.
Declarações sobre manutenibilidade	Deve conter informações sobre as subcaracterísticas; deve conter declarações sobre a manutenibilidade do produto.
Declarações sobre portabilidade	Deve conter informações sobre as subcaracterísticas; deve especificar as configurações de hardware e software e deve fornecer informações sobre instalação.
Declarações sobre qualidade em uso	Deve conter informações sobre as características e deve disponibilizar o relatório de testes.

## b) Documentação de usuário

A documentação de usuário é o conjunto completo de documentos, disponíveis na forma impressa ou não, fornecidos para utilização de um produto. Essa documentação deve incluir todos os dados para a instalação, para o uso da aplicação e para a manutenção do produto de software. A norma ISO 9127 pode ser utilizada para auxiliar nessa documentação. Os principais requisitos da documentação de usuário estão descritos na Tabela 5.3, com a mesma estrutura da tabela anterior. Os requisitos obrigatórios prevalecem nesse tipo de documentação.

Tabela 5.3 – Requisitos de qualidade para a documentação do usuário.

Item	Requisitos
Completeza	Deve conter todas as informações necessárias para o uso do produto, tais como estabelecer todas as funções do pacote, procedimentos de instalação, de backup e os valores-limite.
Correção	A informação apresentada deve estar correta e sem ambiguidade.
Consistência	Deve haver plena coerência entre a documentação e a descrição do produto. Cada termo deve ter um único significado.
Inteligibilidade	Deve ser compreensível pela classe de usuários que desenvolve atividades com o produto, utilizando termos apropriados, exibições gráficas e explicações detalhadas.
Apreensibilidade	Deve ser apresentada de forma que facilite uma visão geral, por meio de índices e tabelas de conteúdo.
Operacionalidade	Deve indicar como imprimir a documentação, se necessário; deve conter sumário e deve definir termos que não sejam comuns.

### c) Software

O último componente dos requisitos de qualidade é referente ao software; para isso utiliza as mesmas definições das características de qualidade da norma NBR ISO/IEC 9126-1. As características de Funcionalidade, Confiabilidade e Usabilidade são destacadas e devem ser verificadas com o uso do produto. Não há requisitos específicos para os aspectos de Eficiência, Manutenibilidade, Portabilidade e Qualidade em uso. Qualquer requisito declarado na documentação do pacote, referente às características citadas, deve estar em conformidade. Os principais requisitos para o software estão descritos na Tabela 5.4.

Tabela 5.4 – Requisitos de qualidade para software

Item	Requisitos
Funcionalidade	Devem ser verificados os procedimentos para instalação do produto; a presença de todas as funções mencionadas; a execução correta dessas funções; a ausência de contradições entre a descrição do produto e a documentação do usuário.
Confiabilidade	O usuário deve manter o controle do produto, sem corromper ou perder dados, mesmo que a capacidade declarada seja explorada até os limites ou fora deles, se uma entrada incorreta é efetuada, ou ainda se instruções explícitas na documentação são violadas.
Usabilidade	A comunicação entre o programa e o usuário deve ser de fácil entendimento, por meio das entradas de dados, mensagens e apresentação dos resultados, utilizando um vocabulário apropriado, representações gráficas e funções de auxílio (help), entre outras; o programa também deve proporcionar uma apresentação

	e organização que facilitem uma visão geral das informações, além de procedimentos operacionais que o auxiliem: por exemplo, a reversão de uma função executada e o uso de recursos de hipertexto em funções de auxílio, entre outros.
Eficiência	Deve estar em conformidade com as declarações na descrição do produto.
Manutenibilidade	Deve estar em conformidade com as declarações na descrição do produto.
Portabilidade	O software deve poder ser instalado; a instalação e a operação devem ser verificadas em todas as plataformas e deve fornecer meios de desinstalação.
Qualidade em uso	Deve estar em conformidade com as declarações na descrição do produto.

### 5.3.2 Requisitos para documentação de teste

O objetivo da documentação de teste é demonstrar a conformidade do software em relação aos requisitos de qualidade para o software. A documentação de teste deve incluir: plano de teste, descrição do teste e resultados dos testes, conforme descritos na Tabela 5.5.

Tabela 5.5 – Requisitos da documentação de teste

Item	Requisitos
Requisitos gerais	As informações contidas na documentação de teste devem ser verificáveis e corretas e não devem apresentar contradições. A documentação de teste deve conter o plano, a descrição e os resultados de testes; deve conter a lista de todos os documentos, e cada documento deve estar completo.
Plano de teste	Todas as características de qualidade devem estar sujeitas a casos de teste e devem ser objetivo de pelo menos um caso de teste. Todas as funções descritas devem estar sujeitas a casos de teste e devem ser objetivo de pelo menos um caso de teste. Os casos de teste devem demonstrar a conformidade do software. Caso os documentos de requisitos sejam mencionados na descrição de produto, devem estar sujeitos a pelo menos um caso de teste. O nível de decomposição funcional, o método do projeto do caso de teste, os procedimentos de instalação, os limites operacionais, possíveis violações, exemplos indicados na documentação de usuário devem ser casos de teste.
Descrição do teste	O objetivo do teste, identificador único, dados de entrada e limites, detalhamento dos passos, saída esperada para cada caso de teste e critérios para interpretação do resultado.
Resultados dos testes	Relatório da execução dos casos de teste, relatório de não conformidade e julgamento dos resultados do teste.

### 5.3.3 Instruções para avaliação de conformidade

A avaliação da conformidade de pacotes de software pode ser vista como uma avaliação de terceira-parte. Sendo a terceira-parte um laboratório com estruturas de certificação ou um laboratório de avaliação interno, que seja independente do desenvolvedor.

Este item especifica como um produto deve ser avaliado em relação aos requisitos de qualidade. Na Tabela 5.6, são mostradas essas especificações, dividindo as instruções em fases e recomendações.

Tabela 5.6 – Instruções para avaliação de conformidade

Fases	Recomendações
Pré-requisitos	Todos os itens descritos devem estar presentes e conformes
Atividades	Requisitos especificados na descrição, requisitos especificados na documentação e requisitos especificados no software devem ser avaliados.
Processo de avaliação	O fornecedor deve prover subsídios para avaliação de terceira-parte, obedecendo às atividades do item anterior.
Relatório	O relatório deve conter: identificação do produto, data de encerramento da avaliação, descrição dos sistemas computacionais utilizados, documentos utilizados, resumo das atividades de avaliação, resumo dos resultados de avaliação de conformidade, resultado detalhado da avaliação de conformidade e relação de não conformidades. Esse documento deve ter suas páginas numeradas e o número total de páginas
Avaliação na fase de acompanhamento	As partes alteradas nos documentos e software, as partes inalteradas, as que devem ser influenciadas e as que podem influenciar pelas partes alteradas ou pelas alterações em um determinado sistema devem ser avaliadas como se fosse um novo software.

## 5.4 Outros requisitos de qualidade

Produtos de software geralmente são destinados para um grande número de usuários e um diversificado tipo de usuários. Vê-se como é importante que o produto forneça uma interface, isto é, uma interação homem-máquina, adequada ao uso e uma documentação completa do usuário, para ser consultada.

Assim, apresenta-se a seguir um conjunto de requisitos para a usabilidade na interface e um conjunto de requisitos para a documentação de usuário.

### 5.4.1 Requisitos de usabilidade na interface

Projetos de interfaces com usuários e serviço ao cliente gerarão provavelmente mais valor agregado às empresas de computadores que a manufatura de hardware; e interfaces com usuários refletem um caminho para diferenciação de produtos, num mercado dominado pela tendência dos sistemas abertos.

Há um tempo a interação entre o homem e o computador recebeu a designação de *user friendly*, porém este termo não é o mais adequado, uma vez que:

- Usuários não precisam que as máquinas sejam amigáveis, precisam de máquinas que não interfiram nos seus métodos, quando eles tentam realizar suas tarefas.
- As necessidades do usuário podem ser descritas ao longo de uma única dimensão, por sistemas que são mais ou menos amigáveis. Na realidade, diferentes usuários têm diferentes necessidades, e um sistema que é amigável para um pode parecer muito tedioso para outro.

Segundo Nielsen (1993), em certo grau, usabilidade é um conceito restrito, comparado com a ampla questão de aceitabilidade de sistema. Tal conceito é basicamente, a constatação de que o sistema é

suficiente para satisfazer todas as necessidades e requisitos dos usuários e outros potenciais *stakeholders*, tais como clientes e gerentes. Grande parte da aceitabilidade de um sistema de computador é uma combinação da aceitabilidade social e da aceitabilidade prática.

Existe, atualmente, uma procura por regulamentação com relação à usabilidade. Algumas voltadas para ergonomia em hardware, que também inclui usabilidade de software. Existem normas que estão sendo criadas, serão oficializadas em alguns países e, certamente, terão grande impacto em outros. Pode-se utilizar usabilidade com a seguinte definição:

- Usabilidade – medida na qual um produto pode ser usado por usuários específicos a fim de atingir objetivos específicos, como eficácia, eficiência e satisfação, em um determinado contexto de uso.

Usabilidade deve ser considerada como fator importante no projeto de produtos de software porque diz respeito a quanto os usuários dos produtos são capazes de trabalhar eficazmente, eficientemente e satisfeitos.

A seguir serão apresentados alguns requisitos que auxiliam na especificação de uma interface, definidos no conjunto de normas da série ISO 9241 (1996 e 1997). Assim, serão explicitadas três orientações, abordando: Princípios de diálogo, Orientações sobre a usabilidade e Organização da informação numa interface. Esses assuntos são tratados nas partes 10, 11, 12, respectivamente.

#### a) Princípios de diálogo

A norma ISO 9241-10, que define princípios de diálogo, estabelece sete princípios importantes para o projeto e avaliação de um diálogo com computador, ou seja, interface. São eles:

- Adequação à tarefa – um diálogo é adequado para uma tarefa quando ele apoia o usuário na conclusão efetiva e eficiente da tarefa;
- Autodescrição – um diálogo é autodescrito quando cada passo é imediatamente compreensível por meio da resposta do sistema ou é explicado ao usuário, quando solicitado;
- Controlabilidade – um diálogo é controlável quando o usuário pode iniciar e controlar a direção e ritmo da interação, até que seu objetivo tenha sido atingido;
- Conformidade com expectativas do usuário – um diálogo está em conformidade com as expectativas do usuário, quando é consistente e corresponde às características do usuário, tais como: conhecimento da tarefa, educação e experiência, e às convenções comumente aceitas;
- Tolerância a erro – um diálogo é tolerante a erros se, apesar de erros de entrada evidentes, o resultado esperado pode ser obtido com pouca ou nenhuma ação corretiva do usuário;

- Adequação à individualização – um diálogo é capaz de individualização quando a interface pode ser modificada, para se adequar às necessidades da tarefa, preferências individuais e habilidades do usuário;
- Adequação ao aprendizado – um diálogo é adequado para aprendizado quando auxilia e orienta o usuário a usar o sistema.

Com esses sete princípios, a interface de um produto de software ficará mais adequada ao usuário, permitindo melhor utilização do produto.

#### b) Orientações sobre a usabilidade

A norma ISO 9241-11 estabelece orientações sobre a usabilidade, atributos de usabilidade e objetivos dos testes de usabilidade.

Esta norma trata dos benefícios de medir a usabilidade em termos de performance e satisfação do usuário, que são medidas pela extensão com que as metas pretendidas são alcançadas, pelos recursos a serem gastos para alcançar as metas, e pela extensão com que os usuários avaliam o produto como aceitável.

Esta norma enfatiza que a usabilidade dos computadores depende do contexto de uso, e o nível de Usabilidade alcançada dependerá de circunstâncias específicas em que o produto é usado.

O contexto de uso consiste em usuários, tarefas, equipamentos, tais como hardware, software e materiais; ambiente físico e social, que podem influenciar a Usabilidade de um produto num sistema de trabalho. Medidas de performance e satisfação do usuário avaliam o sistema de trabalho como um todo. Quando um produto é o foco de interesse, essas medidas fornecem informações sobre a usabilidade daquele produto num contexto de uso particular, fornecido pelo resto do sistema de trabalho. Esses efeitos de mudanças em outros componentes do sistema de trabalho, tais como a quantidade de treinamento ao usuário, ou melhoria da iluminação, podem ser medidos pela satisfação e performance do usuário.

A usabilidade dos produtos pode ser melhorada com a incorporação de características e atributos conhecidos, para beneficiar os usuários em um contexto de uso particular. Para determinar o nível de usabilidade obtida, é necessário medir a performance e a satisfação dos usuários que trabalham com o produto. Medida de usabilidade é particularmente importante do ponto de vista da complexidade das interações entre o usuário, os objetivos, as características das tarefas e os outros elementos do contexto de uso. Um produto pode ter significativamente diferentes níveis de usabilidade, quando usado em diferentes contextos.

#### c) Organização da informação

A norma ISO 9241-12 estabelece a organização da informação distribuída numa interface.

A apresentação da informação deve permitir ao usuário realizar tarefas de maneira eficiente, eficaz e com satisfação. Para atingir esse objetivo, os seguintes atributos devem ser considerados:

- Clareza – o conteúdo da informação é passado rápida e corretamente;
- Descrição – a informação apresentada pode ser corretamente distinguida;
- Concisão – os usuários não são sobrecarregados com informações fora do contexto;
- Consistência – *design* único, em conformidade com a expectativa do usuário;
- Detecção – a atenção do usuário é direcionada para a informação requerida;
- Legibilidade – a informação é fácil de ser lida;
- Compreensibilidade – o significado é claramente compreendido, sem ambiguidade, interpretável e reconhecido.

As orientações apresentadas nessas três normas podem ser consideradas na elaboração de medidas de usabilidade de interface, num produto de software; são orientações em que existe consenso na área.

#### **5.4.2 Requisitos para documentação de usuário**

A documentação de software é um produto fundamental, gerado no processo de desenvolvimento de software. É importante que o documento retrate fielmente o software, de modo que as atividades como avaliação e modificação do software possam ser realizadas sem maiores transtornos, possibilitando facilidade e maior eficiência no manuseio do software por parte dos usuários.

A seguir apresentam-se duas normas sobre documentos de usuário. Uma norma é a ANSI/IEEE 1063 (1987), que especifica os requisitos mínimos sobre a estrutura e o conteúdo da informação de um documento do usuário de software; a outra é a norma ISO 9127 (1988), sobre documentação de usuário e informação de capa para compradores de produtos de software.

##### **5.4.2.1 Documento do usuário de software**

Segundo a norma ANSI/IEEE 1063, documento do usuário de software é o corpo do material que fornece informações aos usuários; tipicamente na forma de material impresso ou armazenado em algum outro meio, no formato de um documento impresso. Os requisitos básicos de documentação de usuário estão apresentados na Tabela 5.7

Tabela 5.7 – Requisitos básicos de Inclusão de um Documento de Usuário do Software.



Componente	Volume único		Multivolumes	
	8 Páginas ou Menos	Mais que 8 Páginas	Primeiro Volume	Demais Volumes
Página de Título	M	M	M	M
Restrições	M	M	M	M
Garantias	R	R	R	R
Índice	O	M	M	M
Lista de ilustrações	O	O	O	O
Introdução				
Descrição do Público-alvo	R	M	M	R
Aplicabilidade	M	M	M	M
Objetivo	R	M	M	R
Uso do Documento	R	M	M	R
Documentos relacionados	R	R	R*	R
Convenções	M	M	M	R
Relato de problemas	R	M	M	R
Corpo do Documento				
Modo Instrutivo	1	1	1	1
Modo de Referência	1	1	1	1
Condições de erro	R	R	R	R
Anexos	O	O	O	O
Bibliografia	M	M	M**	M**
Glossário	M	M	M**	M**
Índice remissivo	2	2	M**	M**

Legenda da Tabela 5.7

- M – Obrigatório: deve ser incluído quando a informação existe.
- O – Opcional.
- R – Referência: incluído numa seção ou referenciado onde a informação pode ser encontrada dentro do conjunto de documentos.
- \* – deve mostrar o relacionamento com os outros volumes do conjunto de documentos.
- \*\* – Obrigatório em pelo menos um dos volumes do conjunto de documentos, e referência a informações, nos demais volumes.
- 1 – Todo documento tem um corpo; cada conjunto de documentos deverá apontar as necessidades de modo instrutivo e de referência do público-alvo.

- 2 – Para documentos com 40 ou mais páginas, é obrigatório um índice, sendo opcional para documentos com menos de 40 páginas.

Essa norma se aplica à documentação que direciona o usuário na instalação, operação e gerenciamento do software.

O documento do usuário de software deve identificar o produto de software, suas aplicações e o público-alvo que irá utilizar o produto; deve determinar o conjunto de documentos para cada público-alvo e o modo de uso de cada documento. O modo de uso pode ser para aprender sobre o software, ou seja, o seu uso é Modo Instrutivo; ou para ajudar em uma busca rápida sobre algum ponto específico do software: então, seu uso é Modo de Referência.

A norma fornece os requisitos para inclusão de informação; alguns requisitos são mandatórios, outros são opcionais. Esses requisitos podem variar, dependendo de que a documentação esteja incluída num único volume ou em mais volumes, e foram sumariamente apresentados na Tabela 5.7.

Além dos requisitos de conteúdo, a norma descreve como apresentar o material na documentação de modo que os documentos sejam fáceis de ler e compreender.

#### ***5.4.2.2 Documentação de usuário e Informação de capa para compradores de pacotes de software***

A norma ISO 9127 descreve a documentação do usuário e a informação de capa fornecida para compradores de pacotes de software.

Segundo a norma, a documentação do usuário fornece ao usuário final toda a informação necessária para instalar e executar o software. Tipicamente, essa documentação, em forma impressa ou eletrônica, vem incluída dentro do pacote de software; ela só estará disponível ao usuário após a compra do pacote de software. A informação de capa está disponível para o possível comprador, externamente ao produto, para que ele possa decidir sobre a aplicabilidade do software, mediante suas necessidades.

Uma lista de itens recomendados para o conteúdo de uma documentação de usuário é apresentada na Tabela 5.8.

Tabela 5.8 – Itens de informação recomendados para um Documento de Usuário

Item da informação	Categoria
Identificação do pacote	ESS
Conteúdo do pacote	ESS
Descrição funcional do software	ESS
Instalação do software	ESS

Utilização do software	ESS
Informação técnica do software	CON
Teste	OPT
Informação contratual	ESS
Glossário	CON
Índice	CON
Comentários de usuários	OPT

Legenda da Tabela 5.8:

- **ESS** – Informação essencial.
- **OPT** – Informação opcional.
- **CON** – Informação condicional, se necessária.

O objetivo da documentação do usuário é fornecer, ao usuário final, informação suficiente para um claro entendimento dos seguintes assuntos: objetivo, funções e características do software; como instalar usar o software; e direitos e responsabilidades contratuais.

Uma lista de itens recomendados para o conteúdo de uma informação de capa é apresentada na Tabela 5.9.

Além da documentação de usuário, a norma sugere outro documento, denominado informação de capa, com o objetivo de permitir que potenciais compradores possam avaliar a aplicabilidade do software quanto às suas necessidades.

Tabela 5.9 – Itens de informação recomendados para o conteúdo da Informação de capa.

Item da informação	Categoria
Identificação do pacote	ESS
Propósito e campo de aplicação	ESS
Ambiente	ESS
Entrada	OPT
Saída	OPT
Restrições	CON
Instruções para uso	OPT
Informação suplementar	OPT
Informação contratual	ESS
Contatos do fornecedor	OPT

Itens fornecidos	ESS
Padrões e leis	OPT
Certificação independente	OPT
Código do produto	ESS
Preço	OPT
Marca registrada	OPT

Legenda da Tabela 5.9:

- **ESS** – Informação essencial.
- **OPT** – Informação opcional.
- **CON** – Informação condicional, se necessária.

O objetivo deste capítulo foi relatar as normas existentes para avaliação e requisitos de qualidade e mostrar como um produto de software pode ser avaliado, de acordo com requisitos estabelecidos em normas internacionais de qualidade.

No Capítulo 6, será apresentada a metodologia de avaliação de produto de software, de acordo com os requisitos de qualidade e processo de avaliação definidos nas normas de qualidade apresentadas até aqui.

## CAPÍTULO 6

### **Metodologia do processo de avaliação**

Descritas as normas reconhecidas internacionalmente com modelos e processos da área de qualidade de produto de software, o próximo passo é apresentar um método sistemático para avaliação. Essa metodologia está de acordo com o processo de avaliação estabelecido pela norma NBR ISO/IEC 14598-1 e 14598-5, com o foco de visão do usuário; foi desenvolvida no âmbito do CTI e demonstrou ser uma metodologia útil para avaliar produtos de software.

No processo de avaliação, existem vários agentes atuantes, ou pessoas com diferentes funções a serem exercidas em determinadas atividades desse processo, tais como:

- Requisitante da avaliação – é a pessoa ou entidade que tem interesse na avaliação; ela especificará o objetivo da avaliação e, finalmente, receberá o resultado.
- Coordenador da avaliação – é a pessoa responsável por conduzir o processo de avaliação como um todo.
- Avaliador – é a pessoa responsável pela medição do processo.
- Profissional estatístico – é a pessoa de apoio estatístico, no momento de estabelecer a pontuação das medidas e a classificação dos dados resultantes da avaliação.
- Comitê de julgamento – em alguns casos de avaliação, pode ser necessário montar um comitê de profissionais (com conhecimento específico na área de aplicação do produto de software), para auxiliar na fase de pontuação e na fase de especificação dos critérios de julgamento, ajudando a estabelecer os juízos de valores das medidas.

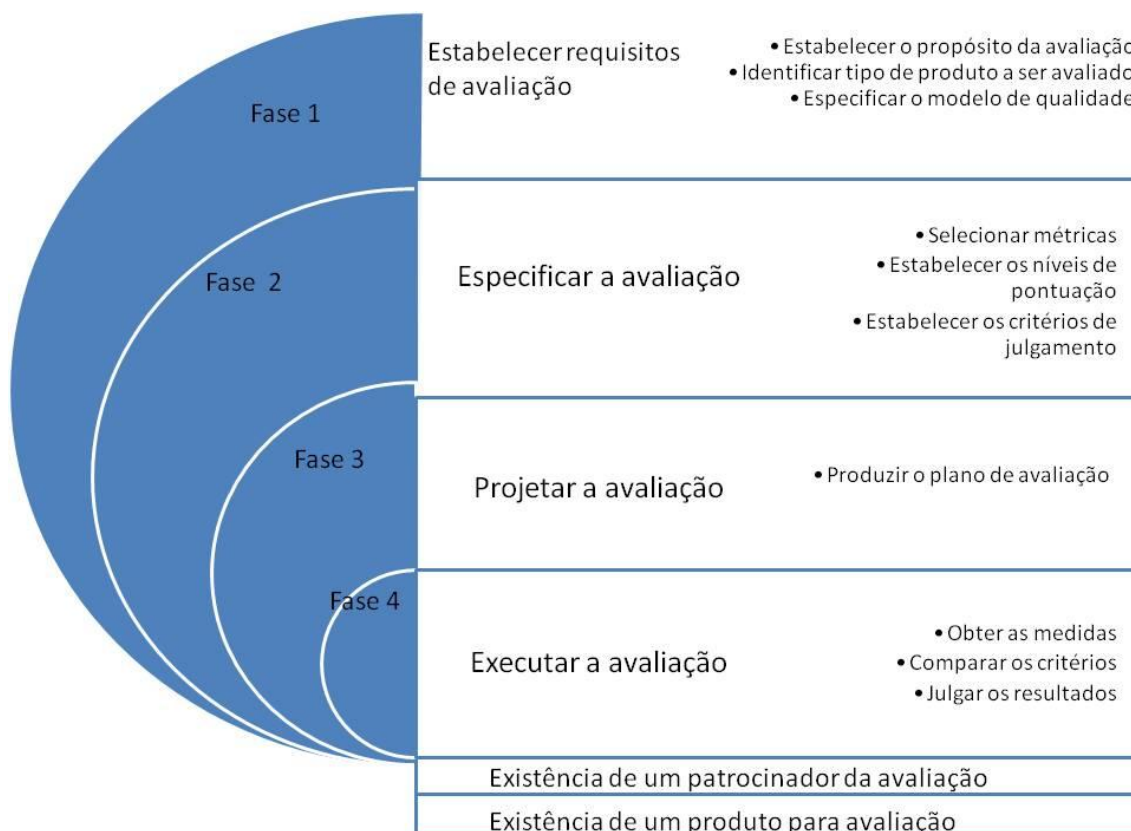
Essas funções podem ser executadas por diferentes pessoas ou, em alguns casos, o coordenador da avaliação e o avaliador podem ser uma só pessoa.

O processo de avaliação descrito neste capítulo é composto de cinco fases, que apresentam uma estrutura bem definida:

- Estabelecer requisitos de avaliação,
- Especificar a avaliação,
- Projetar a avaliação e
- Executar a avaliação.

- Conclusão da avaliação.

Cada uma delas apresenta passos bem determinados, que podem garantir o resultado esperado do processo. A Figura 6.1 apresenta, de forma esquemática, as atividades de um processo de avaliação genérico, que estão descritas na norma NBR ISO/IEC 14598-1, sendo que a conclusão da avaliação é uma sugestão de atividade proposta na parte 5. Esta figura mostra a importância da existência de um patrocinador e de um produto executável para avaliação.



**Figura 6.1 – Processo de Avaliação segundo a norma NBR ISO/IEC 14598-1.**

A seguir, será apresentada cada uma das fases do processo de avaliação, com sugestões para programar as dez atividades nele definidas.

### 6.1 Estabelecer requisitos de avaliação

O objetivo desta fase é estabelecer o que se quer avaliar e o que se quer como resultado da avaliação. O requisitante da avaliação e o coordenador da avaliação têm atuação essencial nesta fase, para que, juntos, possam consolidar os requisitos da avaliação e esclarecer o resultado da avaliação a ser obtido.

Nesta fase do processo de avaliação, estão previstas as seguintes atividades:

- Estabelecer o propósito da avaliação;
- Identificar o tipo de produto a ser avaliado e
- Especificar o modelo de qualidade.

### 6.1.1 Estabelecer o propósito da avaliação

Para que o objetivo da avaliação seja bem claro, é importante que o requisitante da avaliação identifique o software e a versão a ser avaliada. O seguinte questionário é sugerido:

Nome do produto:

Versão do produto:

1. Qual é o domínio da aplicação do produto?
2. Qual é seu objetivo em relação à avaliação?
  - ☐ Assegurar a qualidade do produto?
  - ☐ Indicar pontos para melhoria no produto?
  - ☐ Adequar o produto às normas de qualidade para produto de software?
  - ☐ Comparar com outros produtos similares?
  - ☐ Obter um laudo técnico da sua qualidade?
  - ☐ Classificar e premiar um conjunto de produtos de software?
  - ☐ Pré-qualificar produtos de software numa licitação?
  - ☐ Outros. Quais? Resp.:
3. Quais os aspectos de qualidade do produto que o requisitante da avaliação pretende que sejam avaliados e com que ênfase?
  - ☐ Funcionalidade. Ênfase (1 a 5): \_\_\_\_\_
  - ☐ Confiabilidade. Ênfase (1 a 5): \_\_\_\_\_
  - ☐ Usabilidade. Ênfase (1 a 5): \_\_\_\_\_

☐ Portabilidade. Ênfase (1 a 5): \_\_\_\_\_

☐ Eficiência. Ênfase (1 a 5): \_\_\_\_\_

☐ Completitude. Ênfase (1 a 5): \_\_\_\_\_

Ênfase varia de 1 a 5 da seguinte maneira:

1 – nenhum interesse

2 – baixo interesse

3 – médio interesse

4 – largo interesse

5 – grande interesse

### 6.1.2 Identificar o tipo de produto a ser avaliado

A norma NBR ISO/IEC 25051 estabelece que, para ser um pacote de software, este deve conter: descrição do produto, documentação de usuário e software. Esta afirmação serve também para avaliação de software de diferentes categorias, como as citadas no Capítulo 3. Em casos específicos, algumas adaptações no processo de avaliação são necessárias para adequação ao produto a ser avaliado. A seguir, apresentam-se algumas questões para identificar o produto a ser avaliado e identificar as características do produto a ser avaliado, com relação ao seu porte, documentos que o produto oferece, e requisitos de hardware e software necessários para executar o produto.

Para que seja possível identificar o produto a ser avaliado, é sugerido que o requisitante da avaliação identifique o software e a versão a ser avaliada. O seguinte questionário é sugerido:

Nome do produto:

Versão do produto:

1. Descrição geral do produto:

- De quantas funções o produto é composto? (Obs.: Funções é o número completo de funções da interface do usuário) Resp.:
- Quais são as principais tarefas do produto? Resp.:
- Quais funções merecem maior dedicação durante a avaliação? Resp.:
- Quantas janelas de interação de dados com o usuário o produto possui? Resp.:



- Quem são os principais usuários do produto? Resp.:
- 2. Como é o ambiente no qual o produto será inserido?
- Nível de conhecimento exigido dos usuários em relação à informática. Resp.:
- Nível de conhecimento exigido dos usuários em relação ao domínio da aplicação em si. Resp.:
- 3. Quais são os principais componentes do produto que serão submetidos à avaliação?

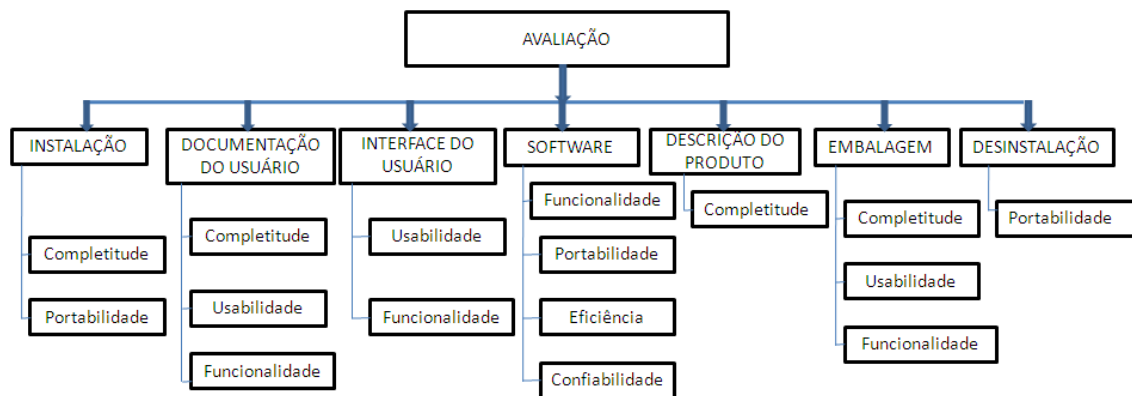
Documento	Impresso (nº de págs.)	On Line (nº de janelas)
Manual do Usuário		
Manual de Operação		
Procedimentos para Instalação		
Descrição do Produto (folder, catálogo, site, etc.)		
Outros. Quais?		

- 4. Existe massa de dados disponível para a avaliação, ou seja, dados-exemplos para agilizar a avaliação? Resp.:
- 5. Especificar os requisitos de hardware e software para executar o produto de software.

Rede (S/N):	Quais:
Processador:	
Clock MHz:	
Espaço necessário em Disco Mb:	
Memória Mb:	
Monitor:	
Unidade de Drive:	
Ambiente Operacional:	
Hardware Adicionais (impressora, scanner, hardlock..) :	
Software Adicionais:	
Multiplataforma (S/N):	Quais:

### 6.1.3 Especificar o modelo de qualidade

O modelo de qualidade adotado foi o definido pela norma NBR ISO/IEC 9126-1. A sugestão é a definição do modelo de qualidade para avaliação de software, apresentado na seção 4.4, o qual é estruturado por sete componentes, e cada um desses componentes com suas respectivas características de qualidade, como mostra a Figura 6.2:



**Figura 6.2 – Estrutura do modelo de qualidade.**

Este modelo de qualidade pode ser visto como um modelo genérico para avaliar software, independente da sua área de aplicação. Em determinados casos ele pode ser adaptado ao contexto do software a ser avaliado, dependendo do objetivo da avaliação explicitado pelo requisitante. Por exemplo, se o software não tiver embalagem ou descrição do usuário esses componentes podem ser desconsiderados.

## 6.2 Especificar a avaliação

O objetivo desta fase é definir as medidas a serem utilizadas na avaliação e estabelecer suas respectivas pontuações, para serem representadas como resultado da avaliação. O coordenador da avaliação e o profissional com conhecimentos estatísticos têm atuação essencial nesta fase, pois será necessário converter as medidas obtidas numa escala numérica normalizada.

Nesta fase do processo de avaliação, estão previstas as seguintes atividades:

- Selecionar medidas;
- Estabelecer níveis de pontuação para as medidas e
- Estabelecer critérios para julgamento.

### 6.2.1 Selecionar medidas

De acordo com o modelo de qualidade definido na Seção 6.1.3, o próximo passo é desdobrar as subcaracterísticas de qualidade em atributos que possam ser medidos e pontuados.

A metodologia adotada aqui foi a elaboração de uma lista de verificação, onde cada componente do software foi desdobrado em questões e itens que possam ser verificados e respondidos pelo avaliador. O desdobramento em itens tem o objetivo de criar uma escala adequada, facilitando o uso pelos

avaliadores, pois os itens são usados como guias. Devem ser, portanto, os mais completos e abrangentes possíveis, expressando corretamente o significado do atributo, do ponto de vista da avaliação. A soma desses itens fornecerá o escore do atributo; os valores possíveis dessa soma formarão a escala.

A proposta é ter um conjunto de itens que mais se relacione com a ideia expressa no atributo, visando obter um conjunto de questões, definindo uma escala diferente para cada assunto expresso no atributo. Assim, aqueles assuntos mais complexos terão escalas mais detalhadas, e aqueles mais simples terão escalas mais reduzidas, facilitando o uso pelos avaliadores e posicionando-os adequadamente frente a cada situação.

O número de itens do atributo depende da complexidade e do grau de subjetividade dele; assim, o número de itens varia de acordo com cada atributo.

Quando é solicitado ao avaliador que avalie um produto por meio de um conjunto de atributos, é dado a ele um conjunto de opções, ou conceitos, cuja situação relativamente ao produto o avaliador deve relatar. O conjunto de conceitos utilizados para um atributo reflete uma escala de respostas que serão usadas, numericamente, por meio de escores. O conjunto total de atributos válidos será, então, usado para avaliar o produto.

Usando somente os itens válidos, por meio de um algoritmo simples, pode-se determinar o escore final do atributo. Observa-se que não é o conceito dado a um simples item que determina o valor do escore, mas sim o conjunto de itens que o atributo possui.

Em relação às questões que fazem parte dessa lista de verificação, o avaliador deve considerar que as questões são proposições lógicas sobre um atributo a ser verificado em uma avaliação. Cada proposição deverá ser o mais objetiva possível, envolvendo apenas um aspecto a ser medido.

Algumas respostas possíveis para as questões são:

- “S” (Sim): para proposições verdadeiras;
- “N” (Não): para proposições falsas;
- “NA” (Não se Aplica): para proposições que fazem referência a um aspecto que não se enquadra no produto em avaliação.

O avaliador deve estar atento para o fato de que ausência daquilo que está sendo avaliado nem sempre significa que a atribuição será “N” (Não). O avaliador deve verificar se a proposição se aplica ao produto de software. Exemplo: se para colocar um produto de software em uso, não existe a necessidade de utilizar nenhum dispositivo de entrada de dados, então o avaliador deverá optar pela alternativa “NA” (Não se Aplica).

- “AP” (Avaliação Prejudicada): para proposições que o avaliador não está em condições de avaliar, ou por falta de recursos, por falta de informações, ou mesmo por falta de conhecimento específico sobre o assunto abordado.

A atribuição dessa alternativa é pouco desejável, pois significa que os recursos de que se dispõe são insuficientes. Exemplo: se a avaliação estiver sendo executada com um equipamento cuja resolução gráfica é diferente da exigida, então o avaliador deverá optar pela alternativa “AP” (Avaliação Prejudicada).

Podem existir outros tipos de respostas, dependendo da proposição lógica feita. Há situações em que a medida pode ser quantificada em “faixas de valores”, tais como:

- “A” (Algumas vezes), “N” (Nunca) ou
- “M” (Muitas), “P” (Poucas) ou
- “M” (Muitos), “N” (Nenhum), “P” (Poucos) ou
- “A” (Alguns), “N” (Nenhum), “T” (Todos).

A Tabela 6.1 mostra os sete componentes do modelo de qualidade e para cada componente, os atributos a serem avaliados.

Tabela 6.1 – Subconjunto do modelo de qualidade

Componente	Característica	Atributo
Instalação	Compleitude	Especificação dos requisitos de Hardware na Documentação do usuário
	Portabilidade	Instruções para Instalação
Documentação do usuário	Compleitude	Identificação do produto
	Usabilidade	Apresentação visual
	Funcionalidade	Acurácia
Interface	Usabilidade	Clareza
	Funcionalidade	Padrões Internos
Software	Funcionalidade	Acesso Seletivo
	Eficiência	Interação com periféricos
	Confiabilidade	Backup
	Portabilidade	Ambiente de Software
Descrição do Produto	Compleitude	Identificação do Produtor
Embalagem	Compleitude	Identificações
	Usabilidade	Aspectos práticos

	Funcionalidade	Aspectos de robustez
Desinstalação	Portabilidade	Instruções

Exemplos de medidas podem ser baseados nos relatórios técnicos ISO/IEC 9126-2, 9126-3 e 9126-4. Esta metodologia utilizou medidas com resultados qualitativos. A seguir são apresentados exemplos de medidas para cada atributo de qualidade da Tabela 6.1.

## Componente: Instalação

Primeiramente, na fase da **Instalação**, mostra-se o atributo “Especificação dos requisitos de hardware na Documentação do Usuário”, com as seguintes medidas, em forma de proposições:

Na documentação do usuário:

- (\_\_\_\_) .1. está especificado o tipo de processador necessário para colocar o produto em uso.

**S**=Sim; **N**=Não.

- (\_\_\_\_) .2. está especificada a *memória RAM* necessária para colocar o produto em uso;

**S**=Sim; **N**=Não.

- (\_\_\_\_) .3. estão especificados os dispositivos de entrada de dados necessários para colocar o produto em uso. Ex.: *CD-Rom*, microfone, placa de som, teclado, *scanner* etc.;

**T**=Todos; **A**=Alguns; **N**=Nenhum.

- (\_\_\_\_) .4. estão especificados os dispositivos de saída de dados necessários para colocar o produto em uso. Ex.: *CD e DVD*, alto-falante, fones de ouvido, placa de som, impressoras, *plotters* etc.;

**T**=Todos; **A**=Alguns; **N**=Nenhum.

- (\_\_\_\_) .5. está especificado o tamanho do *periférico* de armazenamento necessário para colocar o produto em uso.

**S**=Sim; **N**=Não.

- (\_\_\_\_) .6. estão especificadas as *placas de expansão* necessárias para colocar o produto em uso. Ex.: placa de aceleração gráfica;

**S**=Sim; **N**=Não.

Em outra medida, na fase da **Instalação**, mostra-se o atributo “Instruções para Instalação”, com as seguintes medidas em forma de proposições:

Se a instalação pode ser conduzida pelo usuário, a documentação do usuário:

(\_\_\_\_) .1. apresenta instruções para instalação, disponíveis antes de realizá-la.

**S**=Sim; **N**=Não.

Apresentando, as instruções de instalação:

(\_\_\_\_) .2. orientam passo a passo como executar as ações da instalação;

**S**=Sim; **N**=Não.

(\_\_\_\_) .3. estão claras quanto ao que deve ser realizado;

**T**=Todas; **A**=Algumas; **N**=Nenhuma.

(\_\_\_\_) .4. fornecem as informações necessárias para instalar o *software*.

**S**=Sim; **N**=Não.

### **Componente: Documentação do usuário**

No componente do produto de software **Documentação do Usuário Impressa e/ou On-line**, mostra-se o atributo “Identificação do Produto”, com as seguintes medidas, em forma de proposições:

Os *documentos do usuário* impressos estão identificados na capa com:

(\_\_\_\_) .1. o nome do produto;

**T**=Todos; **A**=Alguns; **N**=Nenhum.

(\_\_\_\_) .2. a versão ou data de criação do produto;

**T**=Todos; **A**=Alguns; **N**=Nenhum.

A *documentação do usuário* on-line está identificada com:

(\_\_\_\_) .3. o nome do produto;

**S**=Sim; **N**=Não.

(\_\_\_\_) .4. a versão ou data de criação do produto;

**S**=Sim; **N**=Não.

Em outro exemplo de medida, no componente do produto de software **Documentação do Usuário Impressa e/ou On-line**, mostra-se o atributo “Apresentação Visual”, com as seguintes medidas, em forma de proposições:

A documentação do usuário impressa:

(\_\_\_\_) .1. possui tamanhos de letra de fácil leitura;

**T**=Todos; **Q**=Quase todos; **A**=Alguns; **N**=Nenhum.

(\_\_\_\_) .2. possui tipos de letra de fácil leitura;

**T**=Todos; **Q**=Quase todos; **A**=Alguns; **N**=Nenhum.

(\_\_\_\_) .3. apresenta textos, tabelas e gráficos com uma diagramação bem distribuída;

**S**=Sempre; **A**=Algumas vezes; **N**=Nunca.

A documentação do usuário on-line:

(\_\_\_\_) .4. possui tamanhos de letra de fácil leitura;

**T**=Todos; **Q**=Quase todos; **A**=Alguns; **N**=Nenhum.

(\_\_\_\_) .5. possui tipos de letra de fácil leitura;

**T**=Todos; **Q**=Quase todos; **A**=Alguns; **N**=Nenhum.

(\_\_\_\_) .6. apresenta textos, tabelas e gráficos com uma diagramação bem distribuída.

**S**=Sempre; **A**=Algumas vezes; **N**=Nunca.

Em outro exemplo de medida, no componente do produto de software **Documentação do Usuário Impressa e/ou On-line**, mostra-se o atributo “Acurácia”, com as seguintes medidas, em forma de proposições:

Na documentação do usuário impressa:

(\_\_\_\_) .1. foram encontradas evidências de informações incorretas. Por exemplo: informações sobre convenções: unidades de medida, nome da moeda corrente, número de casas decimais, notação etc.

**M**=Muitas; **P**=Poucas; **N**=Nenhuma.

Na documentação do usuário on-line:

(\_\_\_\_) .2. foram encontradas evidências de informações incorretas. Ex.: informações sobre convenções: unidades de medida, nome da moeda corrente, número de casas decimais, notação etc.

**M**=Muitas; **P**=Poucas; **N**=Nenhuma.

## Componente: Interface

No componente do produto de software **Interface de Usuário**, mostra-se o atributo “Clareza”, com as seguintes medidas, em forma de proposições:

A interface:

(\_\_\_\_) .1. apresenta erros gramaticais;

**M**=Muitos; **P**=Poucos; **N**=Nenhum.

(\_\_\_\_) .2. apresenta erros ortográficos;

**M**=Muitos; **P**=Poucos; **N**=Nenhum.

A interface:

(\_\_\_\_) .3. destaca as palavras em outro idioma. Exemplos: entre aspas, itálico etc;

**T**=Todas; **A**=Algumas; **N**=Nenhuma.

(\_\_\_\_) .4. utiliza códigos e terminologias não significativas.

**S**=Sim; **N**=Não.

O texto apresentado na interface:

(\_\_\_\_) .5. dá margem a interpretações ambíguas.

**M**=Muitas; **P**=Poucas; **N**=Nenhuma.

Em outro exemplo de medida, no componente do produto de software **Interface de Usuário**, mostra-se o atributo “Padrões Internos”, com as seguintes medidas, em forma de proposições:

A interface mantém uma padronização própria em relação:

(\_\_\_\_) .1. ao idioma do mercado a que se destina o produto;

**S**=Sim; **N**=Não.

(\_\_\_\_) .2. à configuração de janelas;

**S**=Sim; **N**=Não.

(\_\_\_\_) .3. à formatação de ícones;

**S**=Sim; **N**=Não.

(\_\_\_\_) .4. às mensagens;



**S=Sim; N=Não.**

(\_\_\_\_) .5. à apresentação de resultados ao usuário;

**S=Sim; N=Não.**

(\_\_\_\_) .6. à posição de um determinado botão que executa uma mesma função em caixas de diálogo distintas. Ex.: botão 'Cancelar', 'Imprimir', 'OK' etc.;

**S=Sim; N=Não.**

(\_\_\_\_) .7. aos formatos dos campos de entrada de dados;

**S=Sim; N=Não.**

(\_\_\_\_) .8. ao formato do cursor para uma mesma situação;

**S=Sim; N=Não.**

### **Componente: Software**

No componente **Software**, mostra-se o atributo "Acesso Seletivo", com as seguintes medidas, em forma de proposições:

O software:

(\_\_\_\_) .1. tem implementado o recurso para acesso seletivo. Ex.: permite acesso de usuários a determinadas tarefas por meio de senhas.

**S=Sim; N=Não.**

Possuindo o recurso para acesso seletivo:

(\_\_\_\_) .2. é compatível com o tipo de informação que manipula;

**S=Sim; N=Não.**

(\_\_\_\_) .3. possui o recurso de controle de número de tentativas para a digitação de senhas;

**S=Sim; N=Não.**

(\_\_\_\_) .4. permite que cada usuário altere sua própria senha;

**S=Sim; N=Não.**

(\_\_\_\_) .5. impede a entrada de usuários não autorizados.

**S=Sim; N=Não.**

Em outro exemplo de medida, no componente **Software**, mostra-se o atributo “Interação com periféricos”, com as seguintes medidas, em forma de proposições:

Quando o software está interagindo com os periféricos (modem, impressora, *scanner* etc.):

(\_\_\_\_) .1. o tempo de resposta é compatível com o periférico;

**T**=Todos; **A**=Alguns; **N**=Nenhum.

(\_\_\_\_) .2. isso impede o uso de outras funções.

**T**=Todos; **A**=Alguns; **N**=Nenhum.

Em outro exemplo de medida, no componente **Software**, mostra-se o atributo “*Backup*”, com as seguintes medidas, em forma de proposições:

O software:

(\_\_\_\_) .1. possui recursos para *backup* dos dados.

**S**=Sim; **N**=Não.

Possuindo recurso de backup, ele oferece flexibilidade para determinar:

(\_\_\_\_) .2. o intervalo de tempo entre backups;

**S**=Sim; **N**=Não.

(\_\_\_\_) .3. após qual operação deve ser feito o backup dos dados.

**S**=Sim; **N**=Não.

Finalmente, em outro exemplo de medida, no componente **Software**, mostra-se o atributo “Ambiente de Software”, com as seguintes medidas, em forma de proposições:

O software:

(\_\_\_\_) .1. pode ser instalado em outro ambiente operacional de software especificado na documentação do usuário (Ex.: *Windows NT*, *MAC OS*, *Unix*, etc.). Qual (is):

**S**=Sim; **N**=Não.

(\_\_\_\_) .2. pode ser instalado em ambiente de rede.

**S**=Sim; **N**=Não.

Quando instalado em outro ambiente operacional de software especificado, o software:

- (\_\_\_\_) .3. executa do mesmo modo, tanto num ambiente quanto no outro, sem que o usuário possa perceber diferenças.

**S**=Sempre; **A**=Algumas vezes; **N**=Nunca.

### Componente: Descrição do Produto

No componente **Descrição do Produto**, mostra-se o atributo “Identificação do Produtor”, com as seguintes medidas em forma de proposições:

No documento de descrição do produto está indicado:

- (\_\_\_\_) .1. o nome do produtor. Pode ser carimbo ou etiqueta impressa;

**S**=Sim; **N**=Não.

- (\_\_\_\_) .2. o endereço do produtor. Pode ser carimbo ou etiqueta impressa;

**S**=Sim; **N**=Não.

- (\_\_\_\_) .3. o telefone, *fax*, *e-mail*, *site* ou outra forma de contato com o produtor.

**S**=Sim; **N**=Não.

### Componente: Embalagem

No componente **Embalagem**, mostra-se o atributo “Identificações”, com as seguintes medidas, em forma de proposições:

A embalagem possui corretamente:

- (\_\_\_\_) .1. o nome do produto;

**S**=Sim; **N**=Não.

- (\_\_\_\_) .2. a versão ou data de criação do produto;

**S**=Sim; **N**=Não.

- (\_\_\_\_) .3. o nome do produtor;

**S**=Sim; **N**=Não.

- (\_\_\_\_) .4. os requisitos de hardware necessários para colocar o produto em uso;

**S**=Sim; **N**=Não.

- (\_\_\_\_) .5. os requisitos de software necessários para colocar o produto em uso;

**S=Sim; N=Não.**

- (\_\_\_\_) .6. identificação das tarefas ou funções que podem ser executadas, utilizando-se o software;

**S=Sim; N=Não.**

Em outro exemplo de medida, no componente **Embalagem**, mostra-se o atributo “Aspectos Práticos”, com as seguintes medidas, em forma de proposições:

A embalagem:

- (\_\_\_\_) .1. possui um encaixe prático e simples para abrir e fechar;

**S=Sim; N=Não.**

- (\_\_\_\_) .2. dificulta o manuseio da mídia;

**S=Sim; N=Não.**

- (\_\_\_\_) .3. dificulta o manuseio da documentação do usuário impressa;

**S=Sim; N=Não.**

- (\_\_\_\_) .4. aciona os outros componentes, de forma a dificultar seu manuseio. Ex.: *hardlock*, cabos etc.

**S=Sim; N=Não.**

Em outro exemplo de medida, no componente **Embalagem**, mostra-se o atributo “Aspectos de Robustez”, com as seguintes medidas, em forma de proposições:

A embalagem:

- (\_\_\_\_) .1. quebra ou rasga com facilidade;

**S=Sim; N=Não.**

- (\_\_\_\_) .2. aciona os componentes, de forma a protegê-los contra choques, umidade. Ex.: apresenta, no seu interior, materiais, tais como: plástico de bolhas, espuma, isopor, papelão etc.;

**S=Sim; N=Não.**

- (\_\_\_\_) .3. contém a mídia embalada;

**S=Sim; N=Não.**

- (\_\_\_\_) .4. contém os manuais embalados separadamente dos outros componentes. Ex.: envelope, invólucro plástico etc.;

**S=Sim; N=Não.**

### **Componente: Desinstalação**

Finalmente, na fase de **Desinstalação**, mostra-se o atributo “Instruções para Desinstalação na Documentação do Usuário”, com as seguintes medidas em forma de proposições:

A documentação do usuário:

- (\_\_\_\_) .1. apresenta instruções para realizar sua desinstalação.

**S=Sim; N=Não.**

Apresentando, as instruções de desinstalação:

- (\_\_\_\_) .2. orientam, passo a passo, como executar as ações da desinstalação;

**S=Sim; N=Não.**

- (\_\_\_\_) .3. fornecem as informações necessárias para desinstalar o software com sucesso.

**S=Sim; N=Não.**

Dessa maneira, outras medidas podem ser definidas, desdobrando os atributos do modelo de qualidade.

O Apêndice A apresenta outros exemplos de medidas que estão agrupadas em forma de lista de verificação, seguindo este modelo de qualidade.

#### **6.2.2 Estabelecer níveis de pontuação para as medidas**

Depois da seleção de medidas, o próximo passo é estabelecer níveis de pontuação para os resultados obtidos pelas medidas. Tais resultados são do tipo nominal e, para que seja possível sua pontuação, será necessário fazer um mapeamento das respostas tipo “Nominal” para valores numéricos.

Nesta metodologia, as medidas são questões que têm vários tipos de resposta; para poder pontuar essas respostas, é feito um mapeamento de cada tipo de resposta em um valor numérico.

Cada atributo deve ter como resultado um valor numérico; esse valor deve ser normalizado, ou seja, a média obtida entre os resultados deve estar entre zero (0) e um (1). Isso é importante, pois todos os atributos devem ter pesos iguais, mesmo que eles tenham um número de itens variáveis. É normal isso acontecer, pois cada atributo é desdobrado em itens, para que se possa verificá-lo de forma mais objetiva

e clara. Assim, num atributo que tem dois, três, ou quatro itens, os valores numéricos desses itens vão assumir valores proporcionais e estarão entre zero (0) e um (1), dependendo de quantos itens o atributo for composto. É necessário que a média desses itens, isto é, que os valores numéricos estejam entre zero (0) e um (1).

Os itens que receberem como tipo de resposta “**NA**” ou “**AP**” devem ter um tratamento diferente, pois esse tipo de resposta não pode prejudicar a avaliação do produto. Nesse caso, é atribuído um valor chamado *missing*, e não se atribui um valor numérico a tais itens.

A seguir, relacionam-se alguns exemplos de tipos de respostas para cada tipo de questão, com seus valores numéricos associados.

Nas medidas em que as opções de respostas sejam “S” e “N”, e “S” seja uma proposição VERDADEIRA, e “N” seja uma proposição FALSA, os valores numéricos são estabelecidos como na Tabela 6.2:

Tabela 6.2 – Tipo de questão 01

Tipo de resposta	Valores numéricos	Significado do tipo de resposta
AP	*	Avaliação prejudicada
N	00,00	Não
NA	*	Não se aplica
S	01,00	Sim

Outro exemplo: nas medidas em que as opções de respostas sejam “S” e “N”, e “S” seja uma proposição FALSA, e “N” seja uma proposição VERDADEIRA, os valores numéricos são estabelecidos como na Tabela 6.3:

Tabela 6.3 – Tipo de questão 02

Tipo de resposta	Valores numéricos	Significado do tipo de resposta
AP	*	Avaliação prejudicada
N	01,00	Não
NA	*	Não se aplica
S	00,00	Sim

Nas medidas em que as opções de respostas sejam “S”, de SEMPRE; “N”, de NUNCA; e “A”, de ALGUMAS VEZES, os valores numéricos são estabelecidos como na Tabela 6.4:

Tabela 6.4 – Tipo de questão 03

Tipo de resposta	Valores numéricos	Significado do tipo de resposta
A	00,50	Algumas vezes
AP	*	Avaliação prejudicada
N	00,00	Nunca
NA	*	Não se aplica
S	01,00	Sempre

Em outro exemplo de medida, em que as opções de respostas sejam “M”, de MUITAS; “N”, de NENHUMA; e “P”, de POUCA, os valores numéricos são estabelecidos como na Tabela 6.5:

Tabela 6.5 – Tipo de questão 04

Tipo de resposta	Valores numéricos	Significado do tipo de resposta
AP	*	Avaliação prejudicada
M	01,00	Muitas
N	00,00	Nenhuma
NA	*	Não se aplica
P	00,50	Poucas

Em outro exemplo em que as opções de respostas sejam “A”, de ALGUNS; “N”, de NENHUM; “Q” de QUASE TODOS; e “T”, de TODOS, os valores numéricos são estabelecidos como na Tabela 6.6:

Tabela 6.6 – Tipo de questão 05

Tipo de resposta	Valores numéricos	Significado do tipo de resposta
A	00,33	Alguns
AP	*	Avaliação prejudicada
N	00,00	Nenhum
NA	*	Não se aplica
Q	00,66	Quase todos
T	01,00	Todos

Esses exemplos mostram como transformar a resposta de cada questão tipo “Nominal” para um valor numérico. Na Seção 6.2.3, é explicado como obter a nota do “Atributo”; em seguida, a nota da “Característica” e, finalmente, a nota do “Componente”, possibilitando quantificar a qualidade do software.

### 6.2.3 Estabelecer critérios para julgamento

“Julgar a qualidade” significa, em essência, interpretar os resultados das medições. O primeiro passo nesse sentido já foi realizado na Seção 6.2.2, quando se estabeleceram níveis de pontuação para as medidas. Em prosseguimento, seria desejável obter conclusões sobre qualidade, a partir do conjunto de valores obtidos da aplicação de medidas.

A seguir, apresenta-se uma sugestão de procedimento para obter resultados sintéticos de avaliação, descrito em algumas etapas:

- Etapa 1 – mapear todos os resultados de medidas para uma escala [0, 1], em que “0” significa o pior resultado possível, enquanto “1” representa o melhor. As medidas sugeridas nesta metodologia seguiram esse padrão.
- Etapa 2 – estabelecer pesos para as características de qualidade de software. Os pesos devem representar a importância relativa de cada item, no julgamento global de qualidade do produto. Os valores devem ser obtidos, principalmente, a partir do requisitante da avaliação – é quem, em última instância, deve dizer o que é importante e o que não é importante, na sua visão, sobre a qualidade do produto; e também a partir de profissionais especialistas na área de aplicação do software.
- Etapa 3 – calcular médias ponderadas, usando os valores das medidas e os pesos das respectivas características. É sugerido, aqui, usar os valores (de 1 a 5) atribuídos para cada característica, na questão 3 da seção 6.1.1. Não se recomenda atribuir pesos aos atributos, uma vez que:
  - i) os atributos não estão rigorosamente definidos na norma NBR ISO/IEC 9126-1;
  - ii) um mesmo atributo pode ser utilizado para avaliar várias subcaracterísticas.

É possível, assim, calcular médias agrupando as medidas de cada característica de qualidade. Por exemplo, calcular a média para usabilidade, dadas as medidas relacionadas com os atributos de usabilidade. O mesmo pode ser feito com cada característica, separadamente.

- Etapa 4 – finalmente, o processo pode ser repetido no nível de “Componente” do software, permitindo que se calcule um índice ou nota, representando a qualidade do software.

O cálculo de notas deve ser feito seguindo abordagem *bottom-up* do modelo de qualidade adotado na Figura 6.2, isto é, primeiro obtendo o valor da nota para cada “Item” (que depende da resposta atribuída



ao produto, dada por cada avaliador); em seguida, a nota do “Atributo”; depois a nota da “Subcaracterística”, da “Característica”, do “Componente” e, finalmente, a nota do “Produto” de software.

A seguir, apresenta-se uma descrição do algoritmo de cálculo das notas, com suas respectivas fórmulas matemáticas, as definições e os cálculos das notas, nos diferentes níveis do modelo de qualidade, conforme a Figura 6.2:

a) Nota do Item

Para  $i = 1 \dots I_j^{klm}$ , seja  $A_i^{jklm}$  a nota do item  $i$ , que pertence ao atributo  $j$ , da subcaracterística  $k$ , da característica  $l$ , do componente  $m$ .

b) Nota do Atributo

Para  $j = 1 \dots J_k^{lm}$ , definimos a nota do atributo  $j$ , da subcaracterística  $k$ , da característica  $l$ , do componente  $m$ :

$$B_j^{klm} = \frac{\sum_{i=1}^{I_j^{klm}} A_i^{jklm}}{I_j^{klm}}$$

c) Nota da Subcaracterística

Para  $k = 1 \dots K_l^m$ , definimos a nota da subcaracterística  $k$ , da característica  $l$ , do componente  $m$ :

$$C_k^{lm} = \frac{\sum_{j=1}^{J_k^{lm}} B_j^{klm}}{J_k^{lm}}$$

d) Nota da Característica

Para  $l = 1 \dots L_m$ , definimos a nota da característica  $l$ , do componente  $m$ :

$$D_l^m = \frac{\sum_{k=1}^{K_l^m} C_k^{lm}}{K_l^m}$$

e) Nota do Componente

Para  $l = 1 \dots L_m$ , seja  $p_l^m$  o peso da característica  $l$ , do componente  $m$ .

Para  $m = 1 \dots M$ , definimos a nota do componente  $m$ :

$$E_m = \frac{\sum_{l=1}^{L_m} p_l^m D_l^m}{\sum_{l=1}^{L_m} p_l^m}$$

f) Nota do Produto

Para  $m = 1 \dots M$ , seja  $q_m$  o peso do componente  $m$ .

Definimos a nota do produto:

$$P = \frac{\sum_{m=1}^M q_m E_m}{\sum_{m=1}^M q_m}$$

Com a utilização desse procedimento, consegue-se obter um valor quantitativo da qualidade, isto é, uma nota para a qualidade do software. Dessa forma, o procedimento permite também:

- 1) confrontar diretamente produtos, no caso de avaliações comparativas, visando seleção para posterior aquisição;
- 2) examinar os resultados da avaliação em diferentes níveis de detalhe: desde o resultado único e sintético, até os resultados colhidos individualmente pelas medidas, passando por médias ponderadas para cada característica.

A comparação só é possível quando a avaliação utilizada sobre os produtos for idêntica, ou seja: utilizar as mesmas medidas e mesmos pesos, além de mesmo ambiente de avaliação, hardware, usuários e outros parâmetros que possam influenciar na avaliação.

### 6.3 Projetar a avaliação

O objetivo nesta fase é produzir o plano de avaliação. Esse plano informa qual método de avaliação será utilizado, com instruções de como utilizá-lo, e especifica os recursos necessários, juntamente com o cronograma das ações, para o avaliador. O plano de avaliação deve conter, explicar e definir o método de avaliação, para que o avaliador possa executar a avaliação completamente.

#### 6.3.1 Método de avaliação

O método de avaliação, aqui apresentado, é composto por uma lista de verificação, como do Apêndice A, que contém, também, um conjunto de instruções a serem dadas ao avaliador, e por um modelo de

relatório que o avaliador pode redigir como o exemplo do Apêndice C, relatando os pontos a serem melhorados e os pontos que estão atendendo os requisitos de qualidade, de acordo com as respostas às questões da lista de verificação.

### 6.3.2 Instruções ao avaliador

O avaliador deve ser uma pessoa treinada no método, conhecer todos os seus atributos. É interessante, também, que o avaliador seja uma pessoa que tenha o seguinte perfil: conhecedor de informática, seja uma pessoa detalhista, disciplinada e organizada. É desejável que conheça o domínio de aplicação do software, porém, caso isso não seja possível, deve haver um treinamento específico com especialistas da área de aplicação.

A seguir, são sugeridas algumas instruções ao avaliador:

- Cada atributo é composto por itens de atributo, que variam em termos de quantidade.
- Todos os itens de todos os atributos devem ser verificados e respondidos.
- As respostas serão atribuídas aos itens.
- Todos os itens com respostas desfavoráveis (i.e. aspectos a serem melhorados) para o produto, ou com respostas “**NA**” ou “**AP**”, devem ser justificados por escrito. A justificativa deverá estar identificada com o número do atributo e do item, para ajudar na elaboração do relatório de avaliação.
- Algumas questões necessitam de anotações do avaliador como complemento da resposta dada; essas anotações deverão ser documentadas, logo em seguida da questão.
- Caso o produto de software apresente erros e/ou falhas durante sua avaliação, o avaliador deverá anotar os passos realizados que tiveram como consequência o erro ou a falha ocorrida, além de anotar o procedimento adotado para solucioná-los, caso o tenha identificado. Isso possibilitará a reprodução posterior do erro ou da falha. É conveniente imprimir as telas com mensagens de erro.
- Caso o avaliador não se sinta seguro para avaliar uma questão, anotar e prosseguir a avaliação, avaliando somente quando sentir segurança da resposta a ser atribuída na questão.
- Algumas questões, para serem avaliadas, requerem um conhecimento maior do produto. Para isso, é importante o avaliador manipular o produto, ler a documentação, executar as funções que o produto possui, navegar pelas janelas, enfim, criar familiaridade com o produto.

### 6.3.3 Recursos e cronograma

O avaliador deverá ter um computador exclusivo para executar a avaliação do produto, com os requisitos de hardware e software necessários para tal execução. Além desse recurso, ele deverá ter um cronograma para fazer a avaliação; de preferência, de maneira contínua e conclusiva, do início ao fim, para não perder o foco da avaliação.

## 6.4 Executar a avaliação

Nesta fase do processo de avaliação, o avaliador utilizará tudo o que foi preparado nas fases anteriores, e estão previstas as seguintes atividades:

- Obter as medidas;
- Comparar com critérios e
- Julgar os resultados.

### 6.4.1 Obter as medidas

Para medição, as medidas selecionadas são aplicadas ao produto de software pelo avaliador. Como resultado, obtém-se a lista de verificação, preenchida com as notas de cada questão; posteriormente será calculada a pontuação das medidas, como sugerido na seção 6.2.2.

Nesta fase é importante que o avaliador siga um procedimento de avaliação estabelecido, para não se perder, e que consiga obter as medidas de forma correta. Assim, são sugeridas as seguintes ações:

#### A) Ações iniciais da avaliação:

- Com o produto em mãos, identificar os folhetos e documentos que vieram dentro e fora da embalagem, ou que foram obtidos no *site*.
- Ler atentamente as informações referentes aos requisitos mínimos de hardware e software, constantes em folhetos, embalagem, documentos e *site*, caso existam.
- Com os requisitos mínimos de hardware e software claros, verificar se o equipamento a ser utilizado para a avaliação atende a essas exigências.
- Finalizar as atividades, preenchendo o formulário de “Identificação da Avaliação” com as seguintes informações: “Data inicial da avaliação”, “Avaliador”, “Produtor”, “Produto”, “Versão” do produto, “Descrição resumida” do produto, “Material apresentado”, “Condições de Instalação e Operação”, conforme documentação e dados disponíveis do equipamento utilizado na avaliação. Esse formulário está apresentado no Apêndice B.

#### B) Elaboração do relatório da avaliação

- Elaborar o relatório de avaliação como sugerido no Apêndice C.
- As observações e justificativas das atribuições que foram anotadas na lista de verificação são muito importantes para a elaboração do relatório de avaliação, pois nela deverão ser registrados:
  - aspectos considerados positivos.
  - aspectos a serem revistos, com recomendações.

#### C) Ações finais da avaliação

- O Avaliador deverá checar se os campos do Apêndice B “Identificação da Avaliação”: nome do “Produto”, “Versão” e nome do “Produtor”, que foram previamente preenchidos, correspondem à realidade apresentada pelo produto; caso tenha havido alguma inconsistência, atualizar esses campos.
- Preencher, no formulário “Identificação da Avaliação”, a data final da avaliação.

### 6.4.2 Comparar com critérios

Na etapa de pontuação, os valores medidos são comparados com critérios pré-determinados. Para isso, é útil utilizar um programa como o “Excel” da Microsoft, ou algum outro programa de software estatístico, como, por exemplo, o “SAS” – *Statistical Analysis System* ou o “SPSS” – *Statistical Package for the Social Sciences*, para que se possam transformar em gráficos os valores medidos, para melhor visualização e interpretação dos resultados obtidos, fazendo uma análise estatística descritiva.

### 6.4.3 Julgar os resultados

O julgamento é a etapa final da fase de execução do processo de avaliação. O resultado é o relatório da avaliação, como apresentado no Apêndice C, e uma declaração de quanto o produto atendeu aos requisitos de qualidade especificados no modelo de qualidade. No julgamento, outros fatores como, por exemplo, tempo, esforço e custo da avaliação, podem ser considerados, caso sejam importantes para o requisitante da avaliação.

### 6.5 Conclusão da avaliação

Esta fase foi incluída aqui, seguindo a definição do processo de avaliação da norma NBR ISO/IEC 14598-5. O objetivo nesta fase é disponibilizar o resultado da avaliação para o requisitante da avaliação. O coordenador da avaliação deverá finalizar a avaliação com as seguintes atividades:

Arquivar os itens utilizados durante a avaliação:

- Relatório de avaliação do produto de software;
- Lista de verificação completamente preenchida;

Finalizar essa tarefa, entregando ao requisitante da avaliação:

- Produto de software nas mesmas condições recebidas, incluindo folhetos e anexos impressos;
- Cópia do relatório de avaliação do produto de software;
- Declaração do resultado da avaliação.

Esta conclusão finaliza as cinco fases do processo de avaliação de produto de software, sendo que esta metodologia é um exemplo de como avaliar sistematicamente e não se limita a criatividade ou necessidade de pessoas envolvidas em avaliações. Cada demanda pode desenvolver seu próprio método de avaliação, o que não pode acontecer é a avaliação ser conduzida sem uma sistemática documentada.

Essa metodologia é composta de uma lista de verificação, um manual para o avaliador e um modelo de relatório de avaliação. A lista de verificação tem medidas que são obtidas por meio de um conjunto de alternativas, isto é, ela utiliza escalas nominais como explicado no item 5.2 deste livro. Assim o resultado de uma avaliação é um relatório com resultados qualitativos sobre o nível de qualidade do produto. Adicionalmente uma análise estatística pode ser efetuada atribuindo notas geradas pela transformação da escala nominal em valores numéricos normalizados, podendo ser obtida uma nota de acordo com o item 6.2.3. A questão de associar escala nominal a valores numéricos para que se executem operações matemáticas tem sido polêmica na literatura, assim como o modelo de qualidade e as medidas da norma ISO/IEC 9126 (AL-KILIDAR, 2005). Exemplos de medidas para avaliar a qualidade de produto de software são encontrados nos relatórios técnicos da ISO/IEC 9126 nas partes 2, 3 e 4. A parte 2 contém exemplos de medidas externas do software, a parte 3 contém exemplos de medidas internas e a parte 4 exemplos de medidas de qualidade em uso.

A metodologia aqui exposta não pretende dar ênfase apenas a medidas para produtos de software e sim apresentar uma visão geral do estágio em que se encontra o produto. A partir dessa avaliação, melhorar o produto para satisfação e conforto do usuário. Isso pode ser obtido no relatório de avaliação quando se considera os pontos fortes e os pontos a serem melhorados em cada uma das características associadas aos componentes do software. Isso pode ser visto no apêndice E – Relatório de avaliação do software – Easy Learn.

Em casos onde existe apenas um produto a ser avaliado, não há comparação a ser feita, portanto não há necessidade de notas, pois o objetivo é conhecer qualidade do produto de forma global. Em casos onde existe a necessidade de uma comparação entre produtos é conveniente fazer uma pontuação das

notas dos produtos para facilitar a classificação dos produtos, um exemplo disso foi efetuado no prêmio Assespro (ROCHA, 2001).

## **6.6 Processo de avaliação para componentes de software**

Este item apresenta uma metodologia de processo de avaliação para componentes de software seguindo os passos do processo de avaliação da norma NBR ISO/IEC 14598-1 e 14598-5. O modelo de qualidade utilizado neste processo de avaliação foi definido no item 4.6.

### **6.6.1 Estabelecer os requisitos da avaliação**

#### **A. Propósito da avaliação**

O propósito da avaliação é avaliar quanto o componente de software está em conformidade com os requisitos especificados, permitindo sua admissão em um possível repositório.

Vários pontos de vista podem ser considerados, dependendo dos diferentes usuários do produto, tais como: comprador, integrador, arquiteto de software, fornecedor, desenvolvedor e operador. Devem ser descritos os objetivos e os requisitos da avaliação, que dependem da necessidade do solicitante, ou seja, do objetivo final da avaliação.

Os solicitantes de uma avaliação podem ser: 1) Equipes de desenvolvimento, que usam o resultado da avaliação para identificar ações corretivas e determinar estratégias de evolução; 2) Vendedores, que usam a qualidade como marketing; 3) Compradores, para avaliar os produtos que estão competindo no mercado; 4) Usuários, para obter confiança no produto; 5) Comunidade de software, para identificar e validar os métodos mais adequados de avaliação e, assim, aumentar a credibilidade das técnicas; 6) Laboratórios de avaliação, para desenvolver uma abordagem de avaliação mais consistente.

A análise de requisitos da avaliação visa definir os requisitos de qualidade do produto. Deve partir dos objetivos da avaliação que traduzem diretamente o interesse do requisitante da avaliação. A análise define profundidade, abrangência, relacionamento da presente avaliação com outras e objetos a serem avaliados.

#### **B. Tipos de produtos a serem avaliados:**

Foram identificados três tipos de produtos a serem avaliados:

Componentes de software – artefatos autocontidos, claramente identificáveis, que descrevem ou realizam uma função específica e têm interfaces claras, em conformidade com um dado modelo de arquitetura de software, documentação apropriada e um grau de reutilização definido. Por se tratar de componentes de software a serem disponibilizados em um repositório, terão as características comerciais

dos produtos COTS; portanto, devem ser acompanhados pelos seguintes artefatos: documento de descrição do componente e documentação para sua utilização.

**Descrição de Componente** – A descrição de componente que define o produto faz parte do conjunto de documentação do componente. Ela fornece informações sobre a documentação para sua utilização e sobre o componente propriamente dito. Os principais objetivos da descrição de componente são: Ajudar o integrador em potencial, na avaliação da adequação do componente às suas necessidades de utilização. Por extensão, ela também fornece informações para venda; Servir como base para testes. A descrição de componente deve estar disponível para pessoas interessadas no componente, antes da aquisição do mesmo, e deve seguir todos os requisitos aplicáveis para descrição de produto da norma NBR ISO/IEC 25051.

**Documentação para uso** – A documentação para uso deve conter as informações necessárias e subsídios para a construção da documentação do sistema a ser construído. Devem ser completamente descritos nessa documentação os seguintes itens:

- todos os serviços e as funções de negócio estabelecidos na descrição de componente;
- todas as características e elementos do modelo de componentes presentes no componente, a saber: interfaces providas, interfaces requeridas, interfaces de gerência, eventos recebidos, eventos enviados, propriedades internas ou externas, e interface de acesso;
- todas as facilidades oferecidas pelo componente, as variáveis de configuração ou customização e a manipulação das propriedades que permitem a configuração do estado dos componentes;
- todas as funções do componente a que os usuários tiverem acesso devem ser completamente descritas na documentação.

### **C. Definição do modelo de qualidade**

O modelo de qualidade a ser adotado pode ser o apresentado na Figura 4.15, no Capítulo 4.

#### **6.6.2 Especificação da avaliação**

Nesta etapa, deve-se definir o escopo da avaliação e as medidas a serem executadas no produto submetido à avaliação. O nível de detalhes na especificação da avaliação deve ser tal que a avaliação seja repetível e reprodutível.

##### **A. Medidas:**

As medidas devem ser desdobradas a partir dos atributos da qualidade externa de componentes de software; devem considerar o contexto de uso do sistema, lembrando que a avaliação deve ser em laboratório com ambiente adequado de execução.



## **B. Estabelecer níveis de pontuação**

Os níveis de pontuação dependem dos tipos de medidas a serem adotadas. Exemplo disso já foi abordado na seção 6.2.2.

## **C. Critérios de julgamento**

O critério de julgamento é a admissão ou não do componente no repositório. Por exemplo, pode-se definir que o componente avaliado atenda, a 70% dos requisitos de qualidade especificados. Este critério pode ser reavaliado nas avaliações-piloto do projeto.

### **6.6.3 Projeto da Avaliação**

Deverá ser produzido um plano de avaliação, que descreva o método de avaliação da qualidade de componentes de software. O Plano poderá ser composto de:

Lista de verificação, que é um conjunto de medidas baseadas no modelo de qualidade da Figura 4.15;

Modelo de relatório, que deve ser gerado para cada avaliação executada, relatando os pontos fortes e pontos a serem melhorados do produto avaliado.

As instruções para o avaliador devem estar, por exemplo, em um documento do tipo Manual do avaliador. Os recursos necessários para executar a avaliação especificada devem ser alocados e disponibilizados de forma a atender o cronograma da avaliação.

### **6.6.4 Execução da Avaliação**

Para obter as medidas da execução de atividades de avaliação, verificar o componente de software de acordo com os requisitos de avaliação, como descrito na especificação da avaliação e como planejado no plano de avaliação. De posse dessas medidas, elas devem ser analisadas estatisticamente com os critérios de julgamento, e assim obter o resultado da avaliação. Nesta fase tem início a primeira versão do relatório de avaliação e os registros da avaliação.

### **6.6.5 Conclusão da Avaliação**

Nesta etapa, deve-se revisar o relatório da avaliação e disponibilizar os dados resultantes da mesma para o requisitante da avaliação. Esta etapa tem como objetivos:

Conclusão do relatório de avaliação e arquivamento dos dados da avaliação.

Revisão conjunta pelo desenvolvedor e avaliador.

Destino dos dados e documentos de avaliação: Para uma avaliação profissional, esses dados são sigilosos e exclusivos ao requisitante da avaliação, sendo qualquer publicação deles de sua inteira responsabilidade.

Relatório de avaliação entregue formalmente:

- os documentos submetidos à avaliação devem retornar ao requisitante, ou ser arquivados por um tempo determinado, ou devem ser destruídos de maneira segura;
- o relatório de avaliação e os registros de avaliação devem ser arquivados por um tempo determinado;
- todos os outros dados devem ser arquivados por um tempo determinado ou destruídos de maneira segura.

O objetivo foi mostrar um processo de avaliação específico para componentes de software. Dessa forma, outros exemplos de avaliação poderiam ser apresentados: um deles é a avaliação específica da característica de qualidade – usabilidade; outro exemplo seria a avaliação focada por categorias de produtos ou por domínios de aplicação.

Este capítulo procurou ressaltar a importância de executar avaliação e produto de software seguindo um processo definido e de forma sistemática, para garantir a repetibilidade, a reprodutibilidade, a imparcialidade e a objetividade, características estas fundamentais para esse processo.

## CAPÍTULO 7

### **Avaliação de um software**

Este capítulo apresenta a avaliação de um software do tipo pacote (COTS) – o “*Easy Learn*”<sup>11</sup> - e seguiu como metodologia o processo de avaliação proposto no Capítulo 6 deste livro. Inicialmente é apresentada uma descrição desse software; e em seguida, são descritas todas as atividades realizadas na aplicação da metodologia.

#### **7.1 Apresentação do software**

O software “*Easy Learn*” foi comprado no mercado nacional sem nenhum objetivo específico. É um dicionário Inglês-Português, Português-Inglês e dicionário poliglota, em seis idiomas: alemão, espanhol, inglês, francês, italiano e português. Possui, também, várias funcionalidades, como um laboratório de pronúncia; apresentam-se situações de viagem, jogo da forca, associação de figuras e de palavras, ditado, tradução de frases, tabelas de gramática, filtro de lista - onde se possibilita filtrar um grupo de palavras de uma determinada categoria. Podem ser feitas, também, traduções de palavras em praticamente todos os programas compatíveis com Windows, desde textos até planilhas eletrônicas, banco de dados ou *browser* da internet. O usuário poderá, também, incluir palavras e até mesmo utilizar dicionários extras.

#### **7.2 Processo de avaliação**

O processo de avaliação aplicado no “*Easy Learn*” foi realizado seguindo-se as mesmas etapas recomendadas pela norma NBR ISO/IEC 14598-1 e 14598-5 (NBR ISO/IEC 14598-1 e 14598-5, 2002), como detalhadas a seguir:

##### **7.2.1 Estabelecer requisitos de avaliação**

Foi levantado o objetivo e a expectativa com relação à avaliação. O objetivo foi conhecer um software adquirido comercialmente; a expectativa foi avaliar o quanto esse software está em conformidade com as normas de qualidade. O requisitante foi o gerente do projeto de avaliação, que gostaria de ter um exemplo disponibilizado publicamente.

##### **A) Estabelecer o propósito da avaliação**

---

<sup>11</sup> O nome fictício “*Easy Learn*” foi usado para preservar a integridade do software.

Apresenta-se, a seguir, o formulário preenchido pelo requisitante da avaliação, que tem como objetivo esclarecer o objetivo da avaliação para ambas as partes: o requisitante da avaliação e o avaliador.

Nome do Produto: *"Easy Learn"*

Versão: 2.0

1. Qual é domínio da aplicação do produto?

Software educacional para ensino e aperfeiçoamento do idioma inglês. O Software é composto dos seguintes módulos: *"Easy Learn"*, Poliglota e o Tradutor.

2. Qual é o seu objetivo em relação à avaliação?

- ☐ Assegurar a qualidade do produto?
- ☐ Indicar pontos para melhoria no produto?
- ☒ **Adequar o produto às Normas de Qualidade para produto de software?**
- ☐ Comparar com outros produtos similares?
- ☐ Obter um Laudo técnico da sua qualidade?
- ☐ Classificar e premiar um conjunto de Produtos de Software?
- ☐ Pré-qualificar Produtos de software numa licitação?
- ☐ Outros. Quais?

3. Quais os aspectos do produto que o requisitante pretende que sejam avaliados e com que ênfase?

☒ **Funcionalidade. Ênfase (1 a 5): 4**

☐ Confiabilidade. Ênfase (1 a 5): 3

☒ **Usabilidade. Ênfase (1 a 5): 5**

☐ Portabilidade. Ênfase (1 a 5): 1

☐ Eficiência. Ênfase (1 a 5): 3

☒ **Compleitude. Ênfase (1 a 5): 3**

## **B) Identificar tipos de produtos a serem avaliados**

A seguir apresenta-se o formulário preenchido pelo requisitante da avaliação, para poder especificar o tamanho do software a ser avaliado.

Nome do Produto: *"Easy Learn"*

Versão: 2.0

1. Descrição geral do produto:

- De quantas funções o produto é composto?

Resp.: Aproximadamente 70 funções, que são acessadas pelo usuário por meio da interface.

- Quais são as principais tarefas do produto?

Resp.: Realiza traduções; treina a pronúncia das palavras; realiza exercícios; inclui palavras no dicionário e/ou utiliza dicionários extras; fornece dicas de utilização de gramática, com tabelas de gramáticas; fornece sinônimo das palavras selecionadas; procura palavras, mesmo sem saber como escrevê-las, por meio de rimas; ditado; anagramas.

- Quais funções merecem maior dedicação durante a avaliação?

Resp. As funções do módulo *“Easy Learn”*.

- Quantas janelas de interação com o usuário o produto possui?

Resp.: Aproximadamente 80 janelas.

- Quem são os principais usuários do produto?

Resp.: Indivíduos ou grupos de pessoas que queiram aprender o idioma inglês.

## 2. Como é o ambiente no qual o produto será inserido

- Nível de conhecimento exigido dos usuários em relação à informática.

Resp.: Conhecimento básico de operação do sistema operacional Microsoft Windows.

- Nível de conhecimento exigido dos usuários em relação ao domínio da aplicação em si.

Resp.: Não é exigido nenhum requisito do domínio de aplicação.

## 3. Quais serão os principais componentes do produto que serão submetidos à avaliação? Marque-os.

(Informar o tamanho - número de páginas ou de janelas, caso seja a documentação on-line)

<b>Documento</b>	<b>Impresso</b> (nº de págs.)	<b>On Line</b> (nº de janelas)
<input type="checkbox"/> Manual do Usuário	50	80
<input type="checkbox"/> Manual de Operação	X	X
<input type="checkbox"/> Procedimentos para Instalação	1	1
<input type="checkbox"/> Descrição do Produto (folder, catálogo, site,...)	Os dados estão na própria embalagem	X
<input type="checkbox"/> Outros. Quais? Guia de Referência Rápida	04	X

## 4. Existe massa de teste para a avaliação? (são dados exemplos para agilizar a avaliação?)

Resp.: Não é necessário, o próprio usuário insere dados durante a utilização e avaliação do produto.

## 5. Especificar os requisitos de hardware e software para executar o produto?

Rede (S/N): N	Quais:
Processador: disponível no mercado	
Ambiente Operacional: <b>Windows NT ou superior</b>	
<i>Hardware</i> Adicionais (impressora, scanner, hardlock..) : <b>Microfone e placa de som (opcional)</b>	

<i>Software</i> Adicionais: <b>Browser para internet</b>	
Multiplataforma (S/N): <b>N</b>	Quais:

### C) Especificar modelo de qualidade

O modelo de qualidade utilizado é o estabelecido na Figura 4.12 do Capítulo 4, e o produto contém a mídia com o software, documentação de usuário, descrição do produto e a embalagem.

Este software possui todos os componentes previstos no modelo de qualidade. Caso não apresentasse algum componente, este poderia ser excluído do modelo ou poderia ser mantido e, na fase de executar a avaliação, na atividade de obter as medidas, seria registrada a ausência daquele componente.

#### 7.2.2 Especificar a avaliação

O objeto avaliado foi o software escolhido e adquirido comercialmente, e o método utilizado foi o apresentado no Apêndice A.

##### A) Selecionar medidas

As medidas selecionadas para esta avaliação são as contidas na lista de verificação do Apêndice A.

##### B) Estabelecer níveis de pontuação para as medidas

A lista de verificação é composta pelos componentes do software, e nela cada componente com seus respectivos atributos foram desdobrados em questões e itens que possam ser verificados e respondidos pelo avaliador, obtendo-se as medidas a serem contabilizadas.

Em relação às medidas que fazem parte dessa lista de verificação, o avaliador deve considerar que são questões do tipo “proposições lógicas” sobre um atributo a ser verificado em uma avaliação.

As respostas possíveis para as questões são:

"S" (Sim) para proposições verdadeiras;

"N" (Não) para proposições falsas;

"NA" (Não se Aplica) para sentenças que fazem referência a um aspecto que não se enquadra no produto em avaliação.

"AP" (Avaliação Prejudicada) para sentenças que o avaliador não está em condições de avaliar, ou por falta de recursos, por falta de informações, ou mesmo por falta de conhecimento específico no assunto abordado.

Assim, para obter os níveis de pontuação das medidas, foram utilizados os valores numéricos estabelecidos nas Tabelas 6.2 e 6.3 do Capítulo 6.

##### C) Estabelecer critérios para julgamento

Como critério de julgamento para este software foi considerado um levantamento feito pelo CTI, que possui um banco de dados com avaliações de aproximadamente 300 produtos de software (MARTINEZ, 1999), e desse levantamento pode-se considerar como critério de julgamento o seguinte:

Para um software obter o **nível Alto**, ele deve estar de acordo com pelo menos 70% dos requisitos das normas internacionais. Isso significa que a nota pode variar de 10,0 a 7,0, numa escala de 0 a 10;

para um software obter o **nível Médio**, ele deve estar de acordo com pelo menos 50% dos requisitos das normas internacionais. Isso significa que a nota pode variar de 7,0 a 5,0;

para um software obter o **nível Baixo**, a nota pode variar de 5,0 a 0,0.

Se a nota do software estiver no intervalo do nível alto, isso significa um nível bom de qualidade; se a nota estiver no nível médio, significa que ele ainda precisa de alguns ajustes; finalmente, se a nota está no nível baixo, significa que terão que ser implementadas correções e o software deve voltar a ser avaliado. Em todos os casos, é gerado um relatório qualitativo do resultado da avaliação, no qual constarão os pontos a serem melhorados.

A seguir é mostrado, na Figura 7.1, o gráfico com os critérios de aceitação.

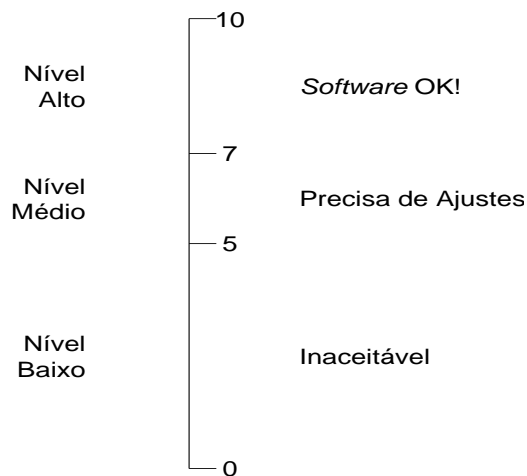


Figura 7.1 – Critérios de aceitação

### 7.2.3 Projetar a avaliação

Foi definido o plano de avaliação:

- a) o método utilizado: lista de verificação do Apêndice A; b) a equipe de avaliação foi composta por uma pessoa conhecedora das normas e do método. O tempo de avaliação foi estimado em 20 horas, durante um período de cinco dias.

O resultado da avaliação foi uma lista de observações referentes aos atributos avaliados. Esse resultado foi apresentado no relatório da avaliação do “*Easy Learn*”, como no Apêndice E

#### **7.2.4 Executar a avaliação**

O avaliador instalou o produto e, seguindo rigorosamente as instruções constantes na documentação impressa ou “*on line*” do produto, exercitou todas as funções disponíveis na versão e finalmente desinstalou.

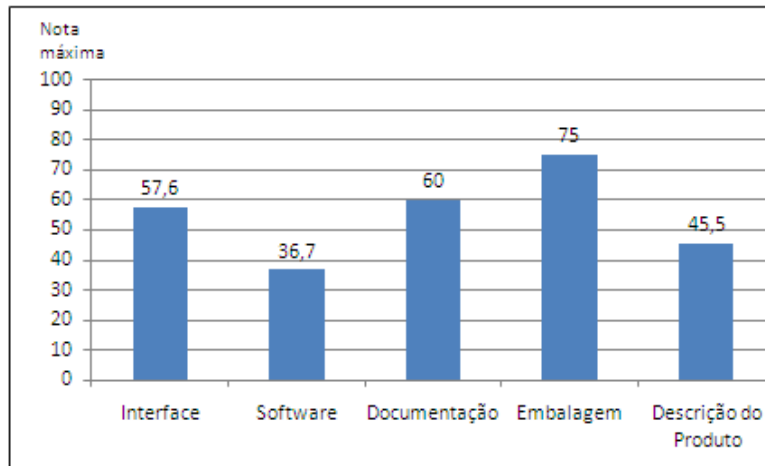
##### **A) Obter as medidas**

Para esta atividade foi seguido o procedimento de avaliação sugerido na metodologia. Como resultado, tem-se a “Identificação da Avaliação”, de acordo com o modelo apresentado no Apêndice B. Concluída a obtenção das medidas, o avaliador elaborou o relatório da avaliação do produto, como sugerido no modelo de relatório de avaliação apresentado no Apêndice E.

##### **B) Comparar com critérios**

Após a avaliação, é interessante que se faça a classificação gráfica dos resultados da avaliação, para ter uma visão geral do comportamento do produto. Isso foi feito por meio da metodologia estatística Análise Descritiva, que verifica e mostra a proporção dos itens atendidos em relação ao conjunto total dos itens das medidas estabelecidas na lista de verificação, em forma de nota entre 0,0 e 100,0, de cada componente do software. O gráfico foi gerado pelo software “Excel”, do Windows 98 da Microsoft. O gráfico é mostrado na Figura 7.2.





**Figura 7.2 – Notas obtidas do “Easy Learn”**

### **C) Julgar os resultados**

Como pôde ser observado, o desempenho do software teve a seguinte pontuação: Embalagem atingiu o nível Alto, com nota 75. Os componentes “documentação” e “interface” obtiveram as notas 60 e 57.6, respectivamente, atingindo o nível médio da escala da Figura 7.1. Os componentes “descrição do produto” e “software” obtiveram as notas 45.5 e 36.7, respectivamente, atingindo o nível baixo. A nota final, como um todo, foi de 55; essa nota, de acordo com o critério de aceitação, atingiu o nível médio. Isso significa que o software precisa de ajustes, ainda não pode considerar-se que esteja de acordo com os requisitos mínimos exigidos pelas normas de qualidade.

#### **7.2.5 Conclusão da avaliação**

Nesta fase são arquivados a lista de verificação, preenchida pelo avaliador, e o relatório da avaliação.

Com relação ao resultado da avaliação de qualidade do “Easy Learn”, é fortemente sugerido que sejam implementados os aspectos a serem revistos, documentados no relatório de avaliação apresentado a seguir.

Comparando a nota dos componentes do “Easy Learn” da Figura 7.2 com os aspectos de destaque positivo e aspectos a serem revistos do relatório de avaliação, pode-se constatar:

O componente Embalagem, que alcançou a nota 75, obteve vários destaques positivos e apresentou poucos aspectos a serem revistos;

Os componentes Documentação do usuário e Interface obtiveram as notas 60 e 57.6, respectivamente, e no Relatório de avaliação podem ser observados vários aspectos a serem melhorados;

O componente Descrição do Produto obteve a nota 45.5; apesar de o produto apresentar a maioria das identificações e declarações exigidas pela norma, ainda faltaram algumas, consideradas importantes;

O componente Software obteve a nota 36.7; à primeira vista, parece ser um valor baixo principalmente por este software já ser comercializado. Pelo Relatório de avaliação, pode-se constatar que ele possui todas as funções declaradas, porém ainda apresenta falhas durante a execução das funções.

Infelizmente, ainda não se tem uma referência pública na qual seria possível fazer uma comparação entre produtos de software. Poucas são as empresas desenvolvedoras de software que conhecem e aplicam os requisitos de qualidade definidos nas normas. Entretanto, os esforços governamentais e de associações têm incentivado e patrocinado as empresas nacionais a melhorarem seus produtos para atender a competitividade dos mercados nacional e internacional.

## Considerações finais

O objetivo deste livro foi apresentar, principalmente, as normas de qualidade de produto de software e as experiências de aplicações dessas normas. Adicionalmente, o livro tem a sutil pretensão de tratar dos seguintes assuntos: i) ressaltar a importância da qualidade de produto de software mediante o atual movimento para a melhoria da qualidade de processos; ii) divulgar os modelos para qualidade de produto de software; iii) conscientizar as pessoas envolvidas, como fornecedores e desenvolvedores, para melhorar o software brasileiro e para compradores e usuários de software saberem exigir seus produtos com o mínimo de qualidade em relação à padrões reconhecidos internacionalmente.

A qualidade de produto de software é um assunto trabalhoso, fica difícil explicitar por meio de requisitos de qualidade as necessidades implícitas e explícitas do usuário. Nesse sentido, o grande benefício da norma NBR ISO/IEC 9126-1 é apresentar um modelo de qualidade básico e genérico para a qualidade de produtos de software. O modelo pode ser desdobrado em atributos de qualidade para as características específicas correspondentes a cada produto, caso necessário. Esse assunto é de difícil compreensão e os modelos práticos aqui apresentados mostram exemplos de como tratar dessa questão.

O livro apresenta a ideia de qualidade genericamente, qualidade de produto de software, produto esse com características tão especiais que se tenta classificá-los para melhor estudá-los. Essa prática facilita a construção de modelos de qualidade com características específicas para cada categoria. Modelos de qualidade são ferramentas úteis que surgiram para ajudar e entender um produto com características tão peculiares, num mercado onde a demanda é grande e os envolvidos muitas vezes não sabem como descrever o que necessitam e desejam. Algumas normas até sugerem modelos de qualidade, outros modelos são criados especificamente para alguns tipos de produtos e é necessário e útil que se pratiquem a construção de novos modelos de qualidade para produtos de software.

O modelo de qualidade ajuda a especificar requisitos de qualidade que serão implementados e supervisionados durante o ciclo de vida de desenvolvimento do produto. O modelo ajuda também nas atividades de avaliação dos produtos intermediários e finais.

Aqui foram ressaltadas as atividades de avaliação da qualidade de produtos de software, entretanto as atividades de implementação de requisitos de qualidade, durante o desenvolvimento do produto ainda é uma área carente de experiências práticas.

As comunidades que pesquisam e praticam qualidade de processo e produto de software se mantêm separadamente nas suas atividades, sendo uma das intenções aqui colocadas é que essas comunidades olhem juntas para um só objetivo que é a qualidade de software como um todo, tanto do ponto de vista da gestão do desenvolvimento quanto do ponto de vista do usuário.

As fábricas de software, com tanta demanda, trabalham numa produção não sistemática, e dependente de pessoas. Isso chamou a atenção de pesquisadores para que houvesse uma

sistematização (organização) nos processos de produção de software. O produto gerado acabou ficando sem a devida atenção no que se refere à qualidade e padronização. As normas de qualidade de produto tratam desses aspectos, no entanto não são ainda devidamente conhecidas pelos produtores, nem pelo mercado.

As avaliações da qualidade de produto de software tratadas no livro são do ponto de vista do usuário final, o que necessita de um produto já em fase final de produção ou num processo diferenciado de produção que contemple a qualidade de produto. Avaliação pode ser realizada do ponto de vista do usuário final, como é o assunto em questão, mas também pode ser do ponto de vista do desenvolvedor, do fabricante, do comprador, entre outros.

A metodologia MEDE-PROS® que é composta de um modelo de qualidade e de um método de avaliação, faz avaliações da qualidade de produtos de software e serve de base para avaliações de outros tipos de software. É evidente que adaptações são necessárias para cada caso em particular, onde, por exemplo, documentação do usuário e descrição do produto são partes importantes de qualquer tipo de software e podem ser consideradas ou não. No caso de software do tipo pacote (COTS) esses componentes citados são obrigatórios de acordo com a norma NBR ISO/IEC 25051.

Um processo sistemático de avaliação da qualidade de produto de software é mostrado no capítulo 6 com detalhes, que tem como base a norma NBR ISO/IEC 14598, partes 1 e 5. Para sua execução é necessário especificar os requisitos de avaliação de acordo com um modelo de qualidade e a utilização de medidas que devem ser oriundas do mesmo modelo de qualidade. Daí a importância do modelo de qualidade que está sendo revisto na nova série ISO/IEC 25000 ser um dos motivos dessa revisão. Esta série enfatiza a utilização do modelo de qualidade para a especificação de requisitos e avaliação.

O MEDE-PROS® utiliza as normas de qualidade para especificar os requisitos e avaliar produtos de software, sendo uma metodologia genérica e que serve para avaliar qualquer produto de software de domínios diversos. A experiência prática apresentada mostrou-se útil e proveitosa para todos os envolvidos e demonstrou grandes benefícios aos produtos avaliados.

Com o auxílio da norma NBR ISO/IEC 9126-1, pode-se especificar e avaliar requisitos de qualidade, de forma objetiva, tanto a “qualidade do produto de software” quanto a “qualidade do processo de seu desenvolvimento e sua manutenção”.

No modelo CMMI e também no MPS, a organização deve trabalhar em parceria com o cliente, entendendo suas necessidades e promovendo um desenvolvimento ou manutenção de produtos de software, que estabelece os requisitos do cliente de forma documentada e garante a alocação desses requisitos aos componentes do software.

Portanto, o CMMI, o MPS e a norma NBR ISO/IEC 9126-1 reconhecem o fato de que a qualidade de um software é altamente influenciada pela qualidade do seu processo. Assim, qualquer iniciativa no sentido de melhorar a qualidade de produto de software deve estar engajada na melhoria da qualidade

de processo de software da organização. Nesse contexto, é possível dizer que qualidade de software é a integração entre qualidade de processo e qualidade de produto, isto é: uma das modalidades isoladamente não será privilegiada.

Isso vem ao encontro da proposta de integração das abordagens de qualidade de software: processo e produto, em benefício do próprio software e da sua utilização. Com essa iniciativa, o foco estará na “qualidade do software”, e não só no processo, ou só no produto final. Ou seja, devem ser executadas as melhores práticas no processo para desenvolver e manter um produto que possua as características de qualidade reconhecidas internacionalmente.

Todas as áreas de processo distribuídas nos níveis de maturidade do modelo de processo CMMI ou MPS possuem instâncias apontadas aqui, com potencial oportunidade de envolver e utilizar os conceitos da norma NBR ISO/IEC 9126-1. Isso reflete que independentemente do nível de maturidade da organização, a oportunidade de utilizar os conceitos, os modelos e o processo de avaliação da qualidade de produto é sempre possível, aplicável e recomendável. Esse assunto também deve ser pesquisado, debatido e exercitado na prática, para que se obtenha resultados concretos.

Na tomada de decisão de avaliar um produto de software ou não, deve ser analisada a relação custo-benefício. Ou seja, alguns aspectos devem ser observados: a execução de um processo de avaliação para produtos de software demanda tempo e recursos financeiros tanto para a preparação dos recursos materiais e humanos necessários ao processo, quanto dos artefatos a serem gerados (atas de reuniões e relatório de avaliação). Portanto, a experiência mostra que é necessário analisar a viabilidade desse tipo de processo ser executado na sua totalidade. Em muitos casos é suficiente aplicar parte desse processo de avaliação para se garantir a qualidade esperada a custo reduzido. Dividir a metodologia em fases é uma boa prática. A existência do atributo ou não, pode ser observada já numa fase inicial, sendo que a sua correta execução pode ser analisada em uma segunda fase.