

UNIVERSIDADE DE TAUBATÉ
Fernando Rodrigues Castanho

**MÉTODO DE AFERIÇÃO DA QUALIDADE DE PRODUTOS DE
SOFTWARE UTILIZANDO QFD**

Taubaté – SP
2009

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

UNIVERSIDADE DE TAUBATÉ
Fernando Rodrigues Castanho

**MÉTODO DE AFERIÇÃO DA QUALIDADE DE PRODUTOS DE
SOFTWARE UTILIZANDO QFD**

Dissertação apresentada para obtenção do Título de Mestre pelo Curso de Mestrado Profissional em Engenharia Mecânica do Departamento de Engenharia Mecânica da Universidade de Taubaté.

Área de Concentração: Produção

Orientador: Dr. Carlos Alberto Chaves

Taubaté – SP
2009

C346m Castanho, Fernando Rodrigues.
Método de aferição da qualidade de produtos de software utilizando QFD./ Fernando Rodrigues Castanho. – Taubaté: Unitau, 2009.

134 f:il;30 cm.

Dissertação (Mestrado) – Universidade de Taubaté. Faculdade de Engenharia Mecânica. Curso de Mestrado em Engenharia de Produção Mecânica.

Orientador: Carlos Alberto Chaves

1. Engenharia de Software. 2. Norma NBR ISO/IEC 9126:2003. 3. QFD - Desdobramento da Função Qualidade. I. Universidade de Taubaté. Departamento de Engenharia Mecânica. II. Título.

CDD(21) 658

FERNANDO RODRIGUES CASTANHO

**MÉTODO DE AFERIÇÃO DA QUALIDADE DE PRODUTOS DE
SOFTWARE UTILIZANDO QFD**

Dissertação apresentada para obtenção do Título de Mestre
pelo Curso de Mestrado Profissional em Engenharia
Mecânica do Departamento de Engenharia Mecânica da
Universidade de Taubaté.
Área de Concentração: Produção
Orientador: Dr. Carlos Alberto Chaves

Data: 30 de Janeiro de 2009

Resultado: Aprovado

BANCA EXAMINADORA

Prof. Dr. Carlos Alberto Chaves – Universidade de Taubaté

Assinatura  _____

Prof. Álvaro Azevedo Cardoso, PhD – Universidade de Taubaté

Assinatura  _____

Prof. Dr. Pedro Luiz de Oliveira Costa Neto – Universidade Paulista

Assinatura  _____

AGRADECIMENTOS

À DEUS, pelo dom da vida.

Ao Prof. Dr. Carlos Alberto Chaves pela incansável orientação, confiança e pelo incentivo atribuído ao desenvolvimento deste trabalho.

Ao Prof. Álvaro Azevedo, PhD por suas importantes opiniões e contribuições no aprimoramento deste trabalho.

Ao meu pai Manuel com muitas saudades.

A minha mãe Izabel pela constante motivação e apoio.

Aos meus médicos Dr. José Pinotti, Dra. Thelma Sanches e Dra. Joanna de Ângelis que me mantiveram em pé na luta pela vida e que tanto me ajudaram e apoiaram nos momentos de extrema dificuldade.

Aos amigos Caio Rodrigues e Edson Nadai pelo apoio e amizade verdadeira.

Agradeço finalmente, a todos que de alguma forma contribuíram para que fosse possível a conclusão deste trabalho.

RESUMO

O presente trabalho de dissertação tem por objetivo o desenvolvimento de um método de aferição da qualidade de produtos de *software*, por meio da aplicação da metodologia QFD. Para atingir o objetivo desta pesquisa, foi conduzido um estudo da literatura especializada, buscando caracterizar a necessidade de se identificar e padronizar parâmetros de qualidade de software, os fatores que a afetam e as normas e padrões conhecidos para sua gestão eficaz. Em seguida, foi desenvolvido o método que se utilizou da metodologia QFD para o desdobramento da qualidade exigida pelos clientes em atributos de software. A padronização destes atributos foi realizada por meio de sua correlação com as características de qualidade encontradas na norma ABNT ISO/IEC 9126:2003. Com essa padronização da qualidade demonstrou ser possível coletar medidas quantitativas e criar uma base de dados históricos, podendo ser utilizada para melhoria contínua da qualidade dos produtos desenvolvidos. Conclui-se que a aplicação da metodologia QFD possibilitou a criação de um método de aferição de qualidade de produtos de software, que desdobra a qualidade esperada pelo cliente em medidas utilizadas para a padronização da qualidade destes produtos.

Palavras-chave:. *Engenharia de Software, Norma NBR ISO/IEC 9126:2003, Qualidade de Software, QFD - Desdobramento da Função Qualidade, Mensuração em Software.*

ABSTRACT

The present dissertation aims at the development of a method of quality golge of software products by means of the application of the QFD methodology. To reach the objective of this research, a study of specialized literature was performed in order to characterize the necessity of identification and standardization parameters of quality of software, the factors that affect it and the known norms and standards for its efficient management. Following, was developed the method that used the QFD methodology for the unfolding of the quality demanded by the customers for software attributes. The standardization of these attributes was carried out supported by its correlation with the quality characteristics found in norm ABNT ISO/IEC 9126:2003. With this quality standardization was demonstrated to be possible to collect quantitative measures and to create a historical database which can be used for the continuous improvement of the quality of the developed products. Concluding, the application of the QFD methodology made it possible the creation of a method of quality golge of software products that unfolds the client quality expectations into measures used for the quality standardization of these products.

Key Words: *Software Engineering, Norm NBR ISO/IEC 9126:2003, Quality Software, QFD - Quality Function Deployment, Software Measurement.*

SUMÁRIO

LISTA DE ILUSTRAÇÕES.....	9
LISTA DE TABELAS.....	10
1 INTRODUÇÃO	11
1.1 Tema.....	14
1.1.1 Justificativa	14
1.2 Engenharia de Software	16
1.2.1 Processo de Software.....	19
1.2.2 Melhoria de Processo de Software	20
1.2.3 Qualidade de Software	26
1.3 Desdobramento da Função Qualidade (QFD)	30
1.3.1 Histórico.....	30
1.3.2 Origem do QFD	31
1.3.3 Definições do QFD	32
1.3.4 Abordagens do QFD.....	39
1.3.5 Abordagem de Akao	40
1.3.6 Abordagem de Bob King.....	41
1.3.7 Abordagem de Makabe	43
1.3.8 A Casa da Qualidade.....	45
1.3.9 Elementos da Casa da Qualidade	47
1.3.10 Tabela das Características de Qualidade	51
1.3.11 Modelo conceitual do QFD	55
1.4 Mensuração em Software.....	57
1.5 Norma ABNT (2003) - Engenharia de Software Qualidade de Produto.....	60
1.5.1 Modelo de qualidade para qualidade externa e interna.....	62
1.5.2 Modelo de qualidade para qualidade em uso	65
1.6 Metodologia de desenvolvimento de produtos de software	66
1.7 Medidas de qualidade para produtos de software	70
1.7.1 Norma ABNT (2003) – Medidas de qualidade de software (I)	71
1.7.2 Padrão IEEE-982 (1988) – Medidas de qualidade de software (II).....	72
1.7.3 Metodologia Praxis – Medidas de qualidade de software (III).....	73
2 OBJETIVO DO TRABALHO	75
3 REFERENCIAL METODOLÓGICO	76
3.1 Tipo de pesquisa	76
3.2 Limites do trabalho	76
3.3 Modelo conceitual do método de aferição da qualidade de produtos de software utilizando a metodologia QFD.....	77
3.3.1 Extração, conversão e correlação das matrizes QFD.....	78
3.3.2 Nível de Avaliação	78
3.3.3 Medidas de qualidade.....	79
3.3.4 Instanciação ¹ do método para metodologia Praxis	80
3.4 Aplicação do método	80

4	PROPOSTA DO MÉTODO DE AFERIÇÃO DA QUALIDADE DE PRODUTOS DE SOFTWARE (MODELO CONCEITUAL)	83
4.1	Importância da metodologia QFD e da norma ABNT (2003) no desenvolvimento do método.....	84
4.1.1	Desenvolvimento do método para aferição da qualidade em produtos de software.....	85
4.1.2	Nível de Avaliação.....	89
4.1.3	Matriz da Qualidade do Usuário.....	91
4.1.4	Matriz da Qualidade em Uso.....	93
4.1.5	Matriz da Qualidade Externa.....	97
4.1.6	Matriz da Qualidade Interna.....	99
4.2	Seleção e classificação das medidas de qualidade do Modelo conceitual.....	103
4.3	Instanciação do método de aferição de qualidade para metodologia Praxis.....	105
4.3.1	Instanciação da Matriz de Qualidade do Usuário ao Praxis.....	106
4.3.2	Instanciação da Matriz de Qualidade em Uso ao Praxis.....	107
4.3.3	Instanciação da Matriz de Qualidade Externa ao Praxis.....	109
4.3.4	Instanciação da Matriz de Qualidade Interna ao Praxis.....	110
5	ESTUDO DE CASO E DISCUSSÃO: Aplicação do método de aferição da qualidade proposto a um produto de software	113
5.1	Método: Aplicação da Matriz de Qualidade do Usuário ao MotorcyclesControl.....	113
5.2	Método: Aplicação da Matriz de Qualidade em Uso ao MotorcyclesControl.....	120
5.3	Método: Aplicação da Matriz de Qualidade Externa ao MotorcyclesControl.....	123
5.4	Método: Aplicação da Matriz de Qualidade Interna ao MotorcyclesControl.....	127
6	CONCLUSÕES	132
6.1	Sugestões para trabalhos futuros.....	133
7	REFERÊNCIAS BIBLIOGRÁFICAS	134
8	ANEXOS	139
8.1	Anexo 1 - ABNT (2003) - Medidas internas da norma.....	139
8.1.1	Response Time.....	139
8.1.2	Functional Adequacy.....	140
8.1.3	Fault Removal.....	141
8.1.4	Input Validity Checking.....	142
8.1.5	User Operation Cancellability.....	143
8.2	Anexo 2 - Padrão IEEE-982 (1988) - Medidas internas do padrão.....	144
8.2.1	Cumulative Failure Profile.....	144
8.2.2	Requirements Tracebility.....	145
8.3	Anexo 3 – Metodologia Praxis - Medidas internas do padrão.....	146
8.3.1	Defeitos de Testes de Aceitação.....	146
8.3.2	Defeitos de Avaliação pelo Usuário.....	146

LISTA DE ILUSTRAÇÕES

Figura 1.	Modelo de Kano	28
Figura 2.	Relação entre QFD, QD e QFDr	33
Figura 3.	Matriz da Qualidade	37
Figura 4.	Associação das Matrizes QFD	39
Figura 5.	Modelo de QFD de Akao	41
Figura 6.	Modelo conceitual de Bob King	42
Figura 7.	Representação da abordagem de Makabe	44
Figura 8.	Matriz da Qualidade Teórica	46
Figura 9.	Características da Qualidade	51
Figura 10.	Unidades Básicas do QFD	55
Figura 11.	Elementos de uma Matriz QFD	56
Figura 12.	<i>Framework</i> de atributos de Fenton	60
Figura 13.	Qualidade no ciclo de vida do <i>software</i>	62
Figura 14.	Modelo de qualidade para qualidade externa e interna	63
Figura 15.	Modelo de qualidade para qualidade em uso	65
Figura 16.	Método proposto – (<i>Modelo conceitual</i>)	84
Figura 17.	<i>Modelo conceitual</i> para aferição de qualidade para produtos de <i>software</i>	87
Figura 18.	Matriz da Qualidade do Usuário	91
Figura 19.	Matriz da Qualidade em Uso	94
Figura 20.	Matriz da Qualidade Externa	97
Figura 21.	Matriz da Qualidade Interna	101
Figura 22.	Classificação de atributos de qualidade adaptada (<i>Modelo conceitual</i>) ...	104
Figura 23.	Planilha de priorização das funções do MotorcyclesControl	116
Figura 24.	Matriz de Qualidade do Usuário para o MotorcyclesControl	117
Figura 25.	Gráfico de importância dos atributos de qualidade	118
Figura 26.	Matriz de Qualidade em Uso para o MotorcyclesControl	120
Figura 27.	Gráfico de importância das características de qualidade em uso	121
Figura 28.	Níveis de avaliação para características de qualidade em uso	122
Figura 29.	Matriz de Qualidade Externa para o MotorcyclesControl	123
Figura 30.	Gráfico de importância das subcaracterísticas de qualidade externa	125
Figura 31.	Níveis de avaliação para características de qualidade externa	126
Figura 32.	Matriz de Qualidade Interna para o MotorcyclesControl	129
Figura 33.	Priorização das medidas internas de qualidade	130

LISTA DE TABELAS

Tabela 1.	Seqüência de matrizes	43
Tabela 2.	Requisitos dos clientes	50
Tabela 3.	Características e subcaracterísticas de qualidade interna e externa	64
Tabela 4.	Características de qualidade em uso	66
Tabela 5.	Fase e interações do Praxis	67
Tabela 6.	Fluxos técnicos e gerenciais do Praxis	68
Tabela 7.	Artefatos do Praxis	69

1 INTRODUÇÃO

Com o acirramento da concorrência na área de desenvolvimento de *software*, a preocupação com qualidade é fator preponderante, uma vez que pode ser um diferencial para as empresas. A garantia da qualidade, esperada pelo usuário, é uma atividade dispendiosa e, muitas vezes, dependente da experiência da organização para definir as medidas para realizá-la. Os projetos de *software*, devido a sua natureza, estão sujeitos a uma série de incertezas e riscos relacionados com custos, escopo/requisitos, recursos humanos e materiais, prazos, expectativas do cliente, da equipe, do investidor, dentre outros (HERBSLEB, 2007).

A definição de medidas de qualidade adequadas para as características da organização, por meio da utilização de um processo organizado, pode resultar em uma melhor adequação da qualidade de seus produtos. Os produtos de *software* são cada vez mais utilizados como ferramenta crítica em várias empresas. Para as quais, o sucesso do negócio pode depender da qualidade do *software* que utilizam. Dessa forma, a qualidade se torna um critério chave na aquisição de seus produtos, porém, na maioria das vezes, a qualidade não é a principal preocupação no desenvolvimento de *software*. Uma empresa fornecedora costuma priorizar o cumprimento do prazo de entrega e a produtividade do projeto, em detrimento da qualidade esperada pelo usuário. Ao invés de acompanhá-la durante a execução do projeto, é, muitas vezes, somente no final que se verifica se a qualidade do produto está adequada. A correção das falhas para adequar o produto causa atrasos no projeto, e conseqüentemente, a insatisfação do cliente. Portanto, é importante acompanhar a qualidade do *software* desde o início do seu desenvolvimento (HERBSLEB, 2007).

O acompanhamento da qualidade de produto pode ser realizado com o auxílio de um processo de definição de medidas, como, *Goal-Question-Metric (GQM)*, *Goal-Driven*

Software Measurement (GDSM) e *Practical Software Measurement (PSM)*. Em linhas gerais, esses processos auxiliam na definição de medidas usadas no desenvolvimento de *software* a partir das metas da organização. Uma organização pode ter meta de desenvolver produtos de alta qualidade. Para definir medidas para essa meta, os processos acima citados desdobram a meta em perguntas, relacionadas com a qualidade do produto, e, para responder a essas perguntas, são atribuídas medidas, que serão coletadas durante o desenvolvimento do *software*. Contudo, o correto desdobramento dessa meta em perguntas e, em seguida, em medidas, depende do conhecimento prévio no acompanhamento da qualidade de produtos de *software* (BASILI, 2000).

Segundo Falbo (2002), o conhecimento adquirido no desenvolvimento de projetos de *software* pode ser uniformizado por meio da padronização da experiência adquirida de acordo com modelos e procedimentos. Na área de engenharia de *software*, alguns modelos e procedimentos são divulgados, por organizações como a *International Standards Organization (ISO)* e o *Institute of Electrical and Electronic Engineers (IEEE)*, no formato de normas e padrões. Entre as normas estabelecidas, é possível citar a norma ABNT (2003). Sendo que, dentre os padrões relacionados com qualidade de *software* cita-se o padrão IEEE-982 (1988).

Em especial, a norma ABNT (2003) categoriza a qualidade de produtos de *software* em características de qualidade, sendo elas divididas em dois modelos: qualidade interna e externa e qualidade em uso. O modelo de qualidade em uso, de acordo com a norma, categoriza a qualidade em quatro características que definem a visão de qualidade do produto *software* em uso pelo usuário: efetividade, produtividade, segurança e satisfação. Já o modelo de qualidade interna e externa, de acordo com a norma, categoriza a qualidade do produto em seis características: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Essas

características são divididas em subcaracterísticas que podem ser mensuradas por medidas internas e externas (FALBO, 2002).

Falbo (2002) acrescenta que a mensuração das subcaracterísticas do modelo de qualidade, da norma ABNT (2003), auxilia no controle da qualidade do produto durante a sua construção. O controle da qualidade possibilita ao gerente tomar ações corretivas no desenvolvimento do produto e, assim, alcançar a qualidade esperada pelo usuário. O controle da qualidade, na área de engenharia de *software*, é assunto discutido há pouco tempo assim como a norma ABNT (2003).

De acordo com Juran (1998), em outras engenharias, o controle da qualidade no desenvolvimento de produtos é assunto discutido há mais tempo. O controle de qualidade na indústria foi abordado pela primeira vez em publicação de um livro em 1951. Nesse livro, o autor sugere a utilização de alguns métodos e ferramentas para auxiliar o planejamento, o controle e a melhoria da qualidade.

Entre esses métodos, a metodologia QFD – Desdobramento da Função Qualidade em inglês “*Quality Function Deployment*”, auxilia o processo de desenvolvimento do produto, buscando, traduzindo e transmitindo as necessidades e os desejos do cliente. Para organizar as necessidades do cliente, a metodologia QFD possui um processo com passos bem definidos que podem ser repetidos e assim, possibilitam que essas necessidades sejam mais bem geridas. Como elas variam para cada produto seria interessante que fossem organizadas de acordo com uma norma (CHENG et al., 2007).

1.1 Tema

Este trabalho baseia-se na aplicação da metodologia QFD (Desdobramento da Função Qualidade) no desenvolvimento de um método de aferição da qualidade de produtos de *software* com objetivo de padronizar a qualidade de produtos de *software* esperada pelos clientes e usuários diretos.

1.1.1 Justificativa

Este tema foi escolhido devido à falta de formalização das necessidades de qualidade esperada de produtos de *software* que são levantadas junto aos clientes e usuários diretos das organizações desenvolvedoras de produtos de *software*. Uma necessidade normalmente esperada pelo cliente é que o *software* seja rápido e fácil de usar. Essa necessidade pode significar diferentes características de qualidade para produtos de *software* como: eficiência e usabilidade. Para o engenheiro de *software*, é necessário relacionar as necessidades de qualidade esperadas pelo usuário com as características de qualidade padronizadas. Com o relacionamento das necessidades de qualidade com as características de qualidade padronizadas por uma norma, como a ABNT (2003), espera-se levantar e documentar melhor a qualidade esperada pelo usuário. Além de levantar corretamente a qualidade esperada pelo usuário, existe uma dificuldade do engenheiro de produtos de *software* em definir as medidas internas usadas para acompanhar a qualidade durante a construção do produto. A qualidade esperada pelo cliente precisa ser mensurada no processo de desenvolvimento do produto para que ela seja devidamente contemplada. Porém, normalmente, é difícil criar um modelo único que relacione a qualidade final do produto e as medidas de qualidade dos atributos intermediários do desenvolvimento de produtos de *software*.

A adoção da metodologia QFD se justifica por permitir desdobrar a qualidade esperada pelo usuário, com auxílio da equipe de desenvolvimento, para o processo de produção da empresa. Com esse método são percebidos os pontos do processo de desenvolvimento que impactam a qualidade final do produto.

A proposta apresentada por esta pesquisa poderá subsidiar a implementação do método em áreas e empresas do segmento de desenvolvimento de *software*, com o objetivo de validar as práticas propostas identificando possibilidades de melhoria, bem como a elevação dos índices de qualidade.

A importância de se divulgar a metodologia QFD para o segmento de desenvolvimento de *software* deve-se ao fato que a percepção de qualidade esperada pelo usuário é um conceito dinâmico, que sofre uma evolução ao longo do tempo, motivada por fatores como a evolução da tecnologia e aspectos sociais. A metodologia QFD já fora utilizado para o levantamento de requisitos de produtos de *software*, porém pouco utilizado na definição de medidas para acompanhar a qualidade durante o desenvolvimento do *software*.

Esta dissertação de mestrado é de caráter explicativo, e visa ampliar a utilização da metodologia QFD na área de desenvolvimento de *software*, bem como contribuir para a melhoria e otimização do processo de desenvolvimento de produtos de *software*.

1.2 Engenharia de *Software*

O termo Engenharia de *Software* – (*Software Engineering*) foi definido pela primeira vez na conferência NATO (*North Atlantic Treat Organization*) em 1968. Segundo Naur (2006), em 1967 um grupo de estudos em ciência da computação foi estabelecido pelo comitê científico da NATO para discutir os problemas do *software*. Naquela reunião foi então proposta a realização de uma conferência de trabalho em engenharia de *software*. O termo engenharia de *software* foi deliberadamente escolhido como uma provocação, para implicar na necessidade da construção de *software* ser baseada nos moldes dos fundamentos teóricos e disciplinas práticas, que são tradicionais nos tipos de engenharia já estabelecidos. Desde então a comunidade tem procurado entender a natureza das atividades de *software* e sua relação com a engenharia. Várias técnicas, modelos e conceitos têm sido desenvolvidos, mas um problema ainda permanece: a distância entre teoria e prática. A criação do SEI (*Software Engineering Institute*) em 1980 com a meta de prover liderança no avanço do estado da prática de engenharia de *software* para melhorar a qualidade de sistemas que dependem de *software* e um balanço mais recente realizado na edição de novembro de 2007 da revista IEEE *Software* (*Institute of Electrical and Electronics Engineers*), sobre o estado da prática, evidenciam essa distância entre teoria e prática (NAUR, 2006).

Naur (2006) critica aqueles que vêem o estado da prática como largamente caótico e sem sucesso e reconhece que muitos pesquisadores acadêmicos não sabem o que ocorre na prática da engenharia de *software*; e que o que ele considera como pior ainda: acham que sabem.

Pollice (2005) comenta que a disciplina engenharia de *software* ou desenvolvimento de *software* amadureceu e avançou na última metade de século, mas alguma coisa ainda falta. Muito, senão a maioria, dos *softwares* que criamos é ainda

produzido, não exatamente engenhariado. Ainda segundo Pollice (2005), não existe consenso se os alunos de engenharia de *software* estão aprendendo engenharia de *software*. Os alunos estão aprendendo a serem membros produtivos de uma equipe de desenvolvimento de *software*, com algum conhecimento de processo de como fazer o processo funcionar para a equipe e assim escrever *software* que satisfaça os requisitos e de lidar com as mudanças constantes que são partes do mundo real. Por tudo isto, o que temos hoje é desenvolvimento de *software* e não, pelo menos ainda, engenharia de *software*.

Mesmo a utilização de engenharia para *software* tem sido contestada. April (2005), propõe o fim da engenharia de *software* e o início de jogos de cooperação econômica como um modelo mais apropriado onde defende que a visão de *software* como um produto tem sido provavelmente um erro e que *software* deve ser considerado como uma mídia na qual armazenamos conhecimento.

Brooks (2006) foi um dos primeiros a reconhecer a importância da gerência de projetos na engenharia de *software* e também identificou a inerente complexidade das atividades de *software* e conseqüentemente, da impossibilidade de soluções mágicas para a engenharia de *software*. Estudou diversas definições de engenharia, suas relações com a engenharia de *software* e conclui que essas definições compartilham um conjunto de cláusulas.

Segundo Brooks (2006), essas cláusulas foram sintetizadas e comentadas em relação à engenharia de *software* como:

- A engenharia envolve a criação de soluções eficientes e não apenas solução de problemas, mas sim, sobre solução de problemas com um uso econômico de todos os recursos, inclusive recursos monetários, financeiros e temporais;
- Para problemas práticos a engenharia trata de problemas práticos, cujas soluções interessam a pessoas fora do domínio da engenharia, essas pessoas

são os clientes da engenharia por meio da aplicação de conhecimento científico;

- Engenharia resolve problemas de uma maneira particular: pela aplicação de ciência, matemática, análise e projeto;
- Construindo coisas enfatizando soluções que são usualmente objetos concretos. Essa característica tem se alterado recentemente na engenharia de *software*, por exemplo, enfatiza soluções abstratas, que são os sistemas de *software* a serviço da condição humana;
- Engenharia não serve apenas para o cliente imediato, mas também desenvolve tecnologia e conhecimento para a sociedade de modo geral. Para atingir esse objetivo, uma engenharia utiliza conhecimentos científicos sobre domínios tecnológicos que estão codificados de uma forma que seja diretamente útil para um engenheiro. Desse modo, esse conhecimento codificado provê respostas para questões que ocorrem com frequência na prática. Ou seja, esse conhecimento deve ser reutilizado para a geração de soluções.

O guia para o corpo de conhecimento da engenharia de *software* SWEBOK-*Guide to the Software Engineering Body of Knowledge* (IEEE, 2004), é uma caracterização, validada por consenso, da disciplina de engenharia de *software* e um guia para o corpo do conhecimento que apóia essa disciplina (ABRAN, 2008).

Segundo Abran (2008), o SWEBOK define engenharia de *software* como:

- A aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de *software*; ou seja, a aplicação de engenharia a *software*;
- O constante re-estudo destas abordagens.

1.2.1 Processo de *Software*

Abran (2008) cita que a utilização mais sistemática de processos foi iniciada nos anos de 1930. Nessa época, Walter Shewhart iniciou um trabalho em melhoria de processo com os princípios do controle estatístico. Esses princípios foram refinados por W. Edwards Deming e Joseph Juran, sempre focados na indústria de manufatura. Nos anos 1970, Philip Crosby notou que as organizações de manufatura com as quais ele trabalhou poderiam ser estudadas de acordo com a qualidade de sua produção e definiu cinco estágios seqüenciais e cumulativos do processo de produção, baseados principalmente nas atitudes gerenciais encontradas em cada estágio. Esses estágios são a Incerteza, o Despertar, o Esclarecimento, a Sabedoria e a Certeza que indicam a qualidade do processo de produção. Nos anos 1980, Ron Radice adaptou esses princípios de controle estatísticos e os estágios de qualidade ao desenvolvimento de *software*. Em seguida, Watts Humphrey estendeu esse trabalho e definiu cinco níveis de maturidade.

Fuggetta (2000) ao concluir uma breve visão geral da história e resultados da pesquisa em processo de *software* reconheceu que a utilização de processos para tratar a inerente complexidade de *software* ganhou força a partir dos anos 1980 e enfatizou que a visão do desenvolvimento de *software* como um processo tem ajudado significativamente a identificação das diferentes dimensões do desenvolvimento de *software* e os problemas que devem ser tratados para estabelecer práticas efetivas.

Ainda segundo Fuggetta (2000), deve-se prestar atenção para a complexa interpelação de numerosos fatores organizacionais, culturais, tecnológicos e econômicos do desenvolvimento de *software*. Ao concluir uma breve visão geral da história e resultados da pesquisa em processo de *software* reconheceu que a utilização de processos para tratar a inerente complexidade de *software* ganhou força a partir dos anos 1980 e enfatizou que a visão do desenvolvimento de *software* como um processo tem

ajudado significativamente a identificação das diferentes dimensões do desenvolvimento de *software* e os problemas que devem ser tratados para estabelecer práticas efetivas.

Na área de pesquisa em processos de *software*, um marco importante é o reconhecimento que processos de *software* são *software* também (EMAM, 2008).

Segundo Emam (2008), a área de processo de engenharia de *software* pode ser examinada em dois níveis. O primeiro nível compreende as atividades técnicas e gerenciais nos processos de ciclo de vida de *software* que são executados durante a aquisição, desenvolvimento, manutenção e aposentadoria de *software*. O segundo é uma meta nível, que compreende a definição, implementação, medição, gerenciamento, mudança e melhoria dos processos do ciclo de vida de *software* propriamente dito. O primeiro nível se confunde com a própria engenharia de *software*, enquanto que o segundo nível pode ser considerado como uma disciplina autônoma, algo como a engenharia da engenharia de *software*. Assim pode-se denominar o primeiro nível de Processos da Engenharia de *Software* e o segundo de Engenharia de Processos de *Software*.

1.2.2 Melhoria de Processo de *Software*

O estabelecimento do que tem sido chamado de Engenharia de Processos de *Software* e Sistemas como uma disciplina autônoma tem-se fortalecido nos últimos anos. A melhoria de processo de *software* tem mostrado na prática ser uma abordagem viável, eficaz e eficiente para a necessária melhoria das organizações intensivas em *software*. Melhoria de processo é na verdade uma abordagem para a melhoria das organizações, alinhada ao contexto e objetivos estratégicos, por meio da melhoria da capacidade de seus processos mais importantes. Normalmente os objetivos estratégicos são uma seleção e composição de fatores como controle e redução dos prazos, custos, e outros recursos para o desenvolvimento, manutenção e operação de sistemas intensivos em

software, que satisfaçam os usuários e a comunidade envolvida, com um número mínimo de erros na execução desses sistemas. Esses objetivos são essenciais com o aumento da utilização de *software* em praticamente todos os sistemas tecnológicos, utilizados direta ou indiretamente pelas pessoas. Do ponto de vista das organizações, esses objetivos também são fundamentais devido ao aumento da competição para o mercado (FARIA, 2005).

As abordagens para melhoria de processo em problemas e metas e as orientações para a melhoria utilizam como referência um modelo de processo que sistematiza e representa as melhores práticas, definem uma medição para avaliação da capacidade dos processos e provêm um roteiro racional para a melhoria dos processos. Os modelos mais utilizados são o SW-CMM versão 1.1 (*Capability Maturity Model for Software*), a Norma ISO/IEC 12207, a Norma ISO/IEC 15504 (*também conhecida como SPICE: Software Process Improvement and Capability Determination*) a evolução do SW-CMM lançada em 2000 o CMMI (*Capability Maturity Model Integration*) e a aplicação para *software* da ISO 9000, principalmente a ABNT (2003) (ROCHA, 2001).

Segundo Rocha (2001), várias abordagens para melhoria de processo de *software* têm como referência o ciclo PDCA (*Plan-Do-Check-Act, Planeja-Executa-Verifica-Atua*), desenvolvido por Shewhart na década de 1930 e popularizado por Deming. O ciclo PDCA é iniciado com o planejamento do que será feito, estabelecimento de metas e definição dos métodos para atingir as metas. A segunda fase envolve a realização das atividades com os métodos definidos e conforme o planejado para atendimento das metas. A terceira envolve a verificação dos resultados em relação às metas definidas. Finalmente, a quarta fase envolve correções de rotas, se for necessárias, e tomadas de ações corretivas ou de melhoria. Variações do ciclo PDCA incluem o ciclo SDCA e o ciclo DMAIC (*Define, Measure, Analyse, Improve, Control*). O ciclo SDCA (*Standard-Do-Check-Act - Padroniza-Executa-Verifica-Atua*) é composto pelas fases de padronização

(*standard*), desenvolvimento, verificação e ação. O ciclo DMAIC é definido como parte da metodologia *Six Sigma* desenvolvida por Bill Smith para a Motorola em 1986. As fases do ciclo DMAIC são: definir oportunidades, medir desempenho, analisar oportunidades, melhorar o desempenho e controlar o desempenho.

Segundo Mcfeeley (2006), uma abordagem para melhoria de processo de *software*, baseada no ciclo PDCA, é a abordagem IDEAL (*Initiating, Diagnosing, Establishing, Acting, Learning*) composta por cinco fases. A fase 1 denominada de “*Início*”, é realizada com estímulo para melhoria, com uma justificativa de negócio para a melhoria, define contexto e estabelece patrocínio, e estabelece infra-estrutura para melhoria. A fase 2, denominada de “*Diagnóstico*”, é realizada com avalia e caracteriza práticas correntes, desenvolve recomendações e documenta resultados. A fase 3, denominada de “*Estabelecimento*”, é realizada com define estratégia e prioridades, planeja ações, e estabelece equipes de ação. A fase 4, denominada de “*Ação*”, é realizada com define processo e medições, planeja e realiza pilotos, e planeja, executa e acompanha instalações. A fase 5, denominada de “*Aprendizado*”, é realizada com documenta e analisa ações e revisa abordagem organizacional.

Kasse (2004) define a Abordagem Baseada em Problemas e Metas (*Goal-Problem Approach*). Nesta abordagem as ações de melhoria são orientadas aos problemas que impedem o atendimento das metas de negócio da organização. Para tanto, as práticas dos modelos de processo podem ser utilizadas como referência, esta é uma abordagem alternativa às duas abordagens mais dominantes, uma baseada na documentação das práticas correntes e outra baseada no atendimento de modelos de processo, que são ambas as abordagens deficientes. Para esta abordagem são definidas três fases: desenvolver plano, implementar plano e verificar andamento. Uma forte tendência atual é a ampliação da abrangência da melhoria de processo de *software* para a melhoria de processo de sistemas.

Melhoria de processo de *software* e sistema é parte da disciplina de engenharia de processo de sistema e *software*, que, por sua vez, é ao mesmo tempo uma disciplina autônoma e parte da disciplina de engenharia de *software* e sistemas. Existe uma tendência atual de generalizar as atividades de *software* para sistemas, cujos componentes podem ser implementados por diferentes engenharias como, por exemplo, elétrica, eletrônica e *software*. Com isto podemos estender processos da engenharia de *software* para processos da engenharia de sistemas; e engenharia de processos de *software* para engenharia de processos de sistemas (FERREIRA, 2005).

Segundo O'Tolle (2005), a utilização de modelos para a melhoria não é suficiente para garantir o sucesso. Para a obtenção da melhoria, alguns fatores têm sido reconhecidos como fundamentais.

Ainda segundo O'Tolle (2005), existe uma estatística informal, que circula pela comunidade, que de cada três iniciativas de melhoria de processo apenas uma tem sucesso. Portanto, é fundamental reconhecer as razões desses resultados. Segundo sua opinião e observações próprias, os pontos relacionados a seguir, sem ordem de importância, são os principais fatores para o sucesso dos esforços para a melhoria:

- Entendimento das características, limitações e implicações da utilização da abordagem baseada em melhoria de processo;
- Escolha, entendimento, utilização e interpretação de um bom modelo de processo ou de um conjunto de modelos;
- Alinhamento dos esforços de melhoria com o contexto e objetivos estratégicos da organização;
- Estabelecimento de metas relevantes, viáveis e mensuráveis;
- Considerações aos aspectos gerenciais, técnicos e humanos da melhoria;
- Comprometimento da alta gerência, condução por empreendedores e participação de todos;

- Condução dos esforços de melhoria como um programa baseado em uma abordagem;
- Conhecimento do processo atual.

Salviano (2001) adverte que a qualidade de um *software* é altamente influenciada pela qualidade do processo utilizado no desenvolvimento e manutenção. Estas premissas apontam para soluções que em um primeiro momento priorize a utilização de princípios básicos de gerência de projeto para “*arrumar a casa*”, gerar resultados imediatos e preparar a organização para as próximas etapas da melhoria. Sem uma gerência de projetos bem estabelecida, o risco de qualquer outra iniciativa não produzir os resultados esperados é muito grande. Melhoria de processo de *software* baseada em modelos (*Model Based Software Process Improvement*) pode ser definida como:

“Uma abordagem para melhoria das organizações intensivas em software, baseada em modelos de capacidade de processo, que orienta ações para alteração dos processos utilizados para aquisição, fornecimento, desenvolvimento, manutenção e/ou suporte de sistemas de software, com o objetivo de estabelecer processos que satisfaçam de forma mais eficiente e eficaz os objetivos e necessidades de negócio da organização.” (SALVIANO 2001).

Ainda segundo Salviano (2001), melhoria de processo também pode ser definida como uma abordagem para *“aprender a trabalhar de forma inteligente para desenvolver e manter melhores sistemas de software, mais barato e em menos tempo”*. Capacidade de processo está relacionada com a habilidade de o processo ser executado de forma eficiente e eficaz. O termo capacidade (*capability*) objetiva ter um processo com as seguintes características:

- Praticado: ser executado de forma consistente sempre que necessário;
- Documentado: estar descrito em alguma notação de representação de processo, utilizando texto, figuras etc., correspondendo a como o processo é executado;
- Treinado: as pessoas que realizam atividades do processo devem ter o conhecimento, habilidade e experiência, necessários para o trabalho;
- Controlado: os artefatos do processo devem ser controlados para garantir sua integridade e disponibilidade, e propostas de mudanças devem ser analisada, aprovadas, planejadas e realizadas antes que sejam institucionalizadas;
- Apoiado: equipe de apoio, ferramentas e outros recursos apropriados devem ser disponibilizados para apoiar a realização do processo;
- Mantido: deve ser alterado para suportar uma evolução contínua;
- Apropriado: deve ajudar as pessoas a realizar o trabalho, e não ser mais um problema a atrapalhar;
- Flexível: deve permitir certa flexibilidade em como o processo é executado, para melhor adaptação baseado em necessidades específicas;
- Medido: medidas de produto e processo devem ser realizadas, consolidadas e analisadas para um melhor entendimento da capacidade do processo;
- Melhorado: deve ser melhorado constantemente para reduzir variações;
- Eliminar desperdícios.

Capps (2008) ressalta que um aspecto importante da melhoria de processo é o equilíbrio do grau de formalização do processo com a complexidade da comunicação em um projeto. Um excesso de formalização do processo pode causar “*burocracia*” enquanto que uma insuficiência de formalização pode causar “*anarquia*”.

1.2.3 Qualidade de *Software*

O conceito de qualidade, muitas vezes tratado subjetivamente, necessita ser definido de forma mais objetiva quanto ao seu significado. Entre as várias definições de qualidade que podem ser encontradas na literatura. Segundo Deming (1982), qualidade é o aperfeiçoamento contínuo e a firmeza de propósitos. Compreender o que acontece, construir e interpretar estatísticas e agir aperfeiçoando. Não há respostas corretas, apenas respostas geradas pelos métodos usados para gerá-las. O objetivo devem ser as necessidades do usuário, presentes e futuras.

O termo qualidade, de acordo com Juran (1998), é adequação ao uso.

A norma ISO/IEC 8402 (1995) define qualidade de produto como sendo a totalidade das características de uma entidade que lhe confere a capacidade de satisfazer às necessidades implícitas e explícitas.

No contexto de engenharia de *software*, Basili (2006) ressalta que qualidade é um conceito multidimensional. Entre suas dimensões podem ser incluídas: a entidade de interesse, o ponto de vista sobre essa entidade de interesse e os atributos de qualidade dessa entidade.

Ainda para a área de *software*, a norma IEEE-610 (1990), define qualidade de *software* como sendo um conjunto de características ou fatores de *software*, que determinam o nível de eficiência do *software* em uso, em relação ao atendimento das expectativas dos clientes.

Dentre essas definições, existe um ponto em comum que é o conceito de que a qualidade do produto deve atender às expectativas dos clientes. Essas expectativas variam da completeza dos requisitos funcionais do produto até o seu desempenho no ambiente do usuário. Dessa forma, o método de aferição da qualidade tem como base principal a qualidade no ponto de vista do usuário e começa com o levantamento da qualidade esperada do produto para o seu desdobramento em medidas de qualidade.

Para levantar a qualidade de forma precisa e padronizada, evitando divagações e direcionando para os atributos de qualidade pode-se, utilizar a técnica 5W1H. A técnica define um conjunto de perguntas, *Why* (Por quê?), *What* (O que?), *Who* (Quem?), *When* (Quando?), *Where* (Onde?) e *How* (Como?), que tem por objetivo auxiliar o usuário a identificar os atributos de qualidade, e assim, maximizar sua satisfação (VIVEIROS, 2006).

Segundo Kano et al. (1996), o aumento da satisfação do usuário depende de quais das suas expectativas foram atendidas pelo produto. O modelo de Kano descreve que todos os produtos e serviços possuem três curvas de expectativas do usuário relacionadas com sua satisfação: atrativa, linear ou básica. Nesse modelo, mostrada na Figura 1, o eixo X representa a presença das expectativas no produto e o eixo Y representa o grau de satisfação do usuário. Para identificar em qual curva as expectativas se encaixam no modelo de Kano, é preciso fazer duas perguntas para o usuário:

1. Como você se sente se a expectativa está ausente?
2. Como você se sente se a expectativa está presente?

A expectativa está presente na curva básica se a resposta para 1 tende a insatisfeito e para 2 tende a neutro. Por sua vez, a expectativa se enquadra na curva atrativa se a resposta para 1 tende a neutro e para 2 tende a satisfeito. Se a resposta do usuário for “*depende*”, a expectativa se enquadra na curva linear.

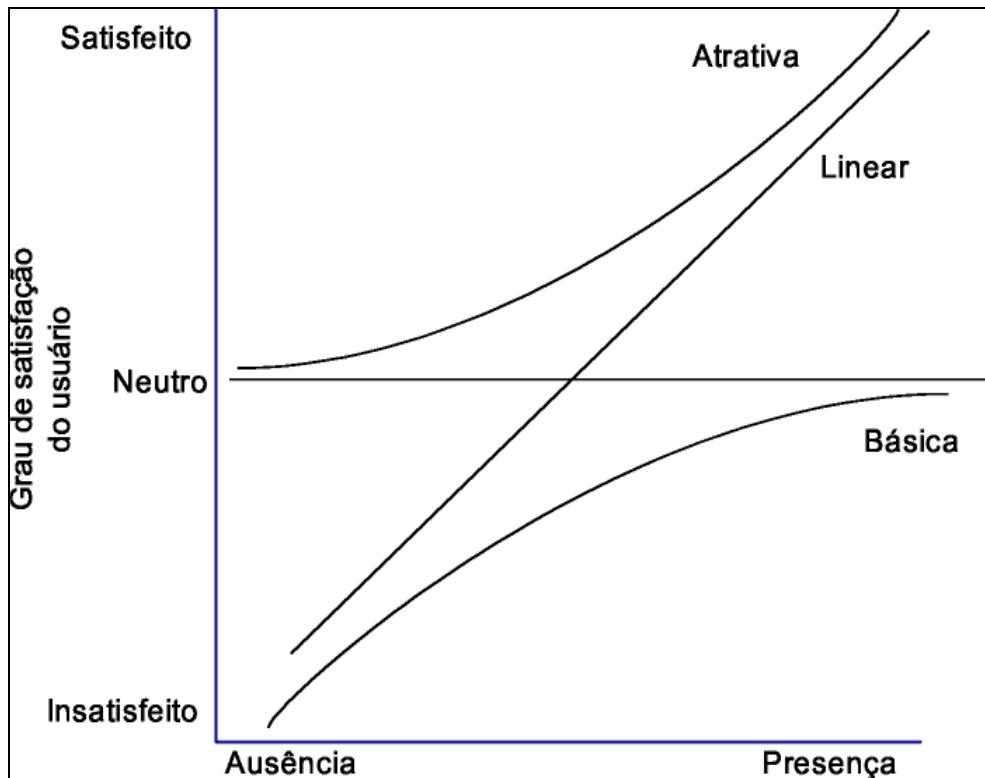


Figura 1– Modelo de Kano

Fonte: Kano (1996)

O modelo de Kano é usado no levantamento da qualidade esperada pelo usuário, da metodologia QFD como guia na identificação dos atributos de qualidade da Matriz de Qualidade. De acordo Kano et al. (1996), o usuário explicita somente as expectativas que estão na curva linear. Para identificar as expectativas que estariam na curva básica ou atrativa, o fabricante poderia usar o histórico da qualidade de outros produtos, se disponível. Para exemplificar o modelo de Kano, pode-se utilizar como produto o automóvel e as expectativas de seus usuários. Em um automóvel, é esperado que ele seja confortável e fácil de dirigir. Esses dois atributos de qualidade não são expressos pelo usuário, mas a ausência deles causa muita insatisfação do usuário em relação ao carro. Sendo assim, esses atributos de qualidade se encontram na curva básica, do modelo de Kano.

De acordo com Cheng et al. (1995), a implantação do QFD objetiva duas finalidades específicas:

1. Auxiliar o processo de desenvolvimento do produto, buscando, traduzindo e transmitindo as necessidades e desejos do cliente;
2. Garantir qualidade durante o processo de desenvolvimento do produto.

Ainda segundo Cheng et al. (1995), outro ponto extremamente atrativo do QFD é que o processo é bem definido, padronizado e permite o rastreamento entre entradas e resultados. Isto facilita sua implantação e também ajuda na criação e manutenção de bases históricas de conhecimento das empresas.

Zultner (2002) diz que a metodologia QFD já foi utilizada como referência para outros trabalhos na área de engenharia de *software*, como por exemplo, o SQFD (*Software Quality Function Deployment*). Nesses trabalhos, a metodologia QFD é utilizada para auxiliar o levantamento de requisitos de qualidade esperados pelo usuário, contudo existe pouca preocupação na padronização dessa qualidade. Outro ponto que pode ser observado nesses trabalhos é que não objetivam a definição de medidas para a qualidade levantada com o usuário.

Já o trabalho de Fehlmann (2002) mostra uma utilização da metodologia QFD para definir medidas, chamadas de “métricas combinatórias”, para o desenvolvimento de *software*. Nesse trabalho, Fehlmann (2002) desdobra as necessidades definidas pelo usuário no processo de desenvolvimento, verificando seus relacionamentos e tentando definir suas medidas. Porém, não ficam claro como são selecionadas as medidas para acompanhar a qualidade durante o desenvolvimento. Outro ponto a observar no trabalho é que a qualidade levantada com o usuário não é padronizada, dificultando o reuso do conhecimento adquirido nos projetos.

1.3 Desdobramento da Função Qualidade (QFD)

A base conceitual da metodologia considerada (QFD) está em TRANSFORMAR a voz dos clientes (suas necessidades) em especificações de produtos ou serviços, de forma que estas necessidades (requerimentos) estejam contidas aos produtos ou serviços finais. Em suma, o QFD é uma metodologia de planejamento estratégico, que traz como característica importante uma melhoria da Qualidade, no seu sentido mais amplo, satisfazendo os desejos e necessidades dos clientes (GUIMARÃES, 1996).

1.3.1 Histórico

Diferente de tudo o que aconteceu no passado, hoje a conjuntura mundial vive constantes e rápidas mudanças, caminhando para uma nova era. O ambiente mundial está sendo causa dos efeitos da revolução tecnológica, da internacionalização e da setorização da indústria, que estão causando mudanças que estão transformando a estrutura da própria indústria e do consumo (AKAO, 1996).

Considerando a diversificação do mercado, as alterações dos valores e das preferências vêm causando significativas transformações no comportamento dos consumidores e, mais, as empresas estão inseridas dentro de um ambiente econômico que sofre bruscas mudanças, implicando na sua própria sobrevivência, onde se pode citar: conflito comercial com outros países, valorização da moeda seguida de sua desvalorização, crise no Oriente Médio, etc. (AKAO, 1996).

Na década de 1960, período de grande desenvolvimento econômico, as indústrias japonesas, principalmente a indústria automobilística, cresceram muito rápido. Devido às constantes mudanças realizadas nos modelos de automóveis, implementou-se a Garantia da Qualidade (GQ) desde o estágio de projeto até o estágio de fabricação. Desta forma o

Controle da Qualidade, que consistia no Controle Estatístico do Processo (CEP), passou a ser Gestão pela Qualidade Total (GQT) (AKAO, 1996).

Após isto, foram introduzidos os conceitos da confiabilidade e, em seguida, as primeiras tentativas de Desdobramento da Qualidade, devido aos seguintes motivos: falta de clareza na determinação da qualidade do projeto, dúvidas relativas ao padrão técnico de processo. (AKAO, 1996).

A partir de 1966, foram feitas pesquisas em conjunto com as empresas e com a divulgação da Matriz da Qualidade, elaborada pelo estaleiro KOBE da *Mitsubishi Heavy Industries*. Foi consolidada a idéia sobre o Desdobramento da Função Qualidade através da junção do Desdobramento da Qualidade (AKAO, 1996).

Com a apresentação desta matéria numa revista da Associação Americana de Controle da Qualidade, em 1993, o Desdobramento da Função Qualidade passou a ser divulgado nos Estados Unidos como QFD (*Quality Function Deployment*) (AKAO, 1996).

A metodologia QFD assumiu um papel estratégico, sendo considerada uma arma competitiva no auxílio das empresas em identificar as necessidades dos clientes e os pontos fortes e fracos dos concorrentes (GARVIN, 1992).

1.3.2 Origem do QFD

O QFD foi criado no Japão, principalmente pelos professores Mizuno e Akao. Desde então, tem sido continuamente aperfeiçoado pelo grupo do professor Akao, hoje com base na Universidade de Tamagawa, em cooperação com empresas japonesas (CHENG, 1995).

O QFD foi originalmente sugerido por Yoji Akao em 1966 e praticado nos estaleiros KOBE da *Mitsubishi Heavy Industries* em 1972 (AKAO, 1966).

No período de grande desenvolvimento econômico, na década de 1960, as indústrias japonesas cresceram rapidamente, principalmente a indústria automobilística. As constantes mudanças nos modelos de automóveis deram origem a necessidade de implementar as atividades da Garantia da Qualidade (GQ), desde o estágio de estabelecimento da qualidade do projeto, pois bastava mais garantir a qualidade da fabricação. Desse modo, o Controle da Qualidade, consistia no Controle Estatístico do Processo (CEP), passou necessariamente para a Gestão pela Qualidade Total (GQT), vivendo uma fase de transição, entre 1960 e 1965, e a campanha “CQ praticado por todos” foi adquirindo força (AKAO, 1996).

No decorrer desta situação, foi introduzido, em primeiro lugar, o conceito da confiabilidade e, a seguir, foram iniciadas, a partir de 1966, as tentativas de Desdobramento da Qualidade. O desdobramento da Função Qualidade se consolidou com a junção do Desdobramento da Função Qualidade no sentido restrito com o Desdobramento da Qualidade o qual passou a ser amplamente divulgado e aplicado nos Estados Unidos como QFD (*Quality Function Deployment*) (AKAO, 1996).

1.3.3 Definições do QFD

O QFD é uma metodologia comprovadamente eficaz para se obter respostas rápidas do que os clientes necessitam. Utilizando equipes multifuncionais, com conhecimento sobre o desempenho dos produtos junto aos clientes, tem-se um eficaz meio de comunicação para “ouvir o cliente”, questão chave para transformar necessidades dos clientes em projetos de sucesso (AKAO, 1990).

Segundo Pande et al. (2001), o QFD é uma metodologia robusta, podendo ser empregada na priorização e seleção de projetos, comparação com os processos e produto da concorrência e *conversão* das necessidades dos clientes em desempenho.

Cheng et al. (1995) define o QFD como “uma forma de comunicar sistematicamente informação relacionada com qualidade e de explicitar ordenadamente trabalho relacionado com a obtenção da qualidade”.

Cheng et al. (2007) diz que o QFD é um método eficiente de desenvolvimento de novos produtos e mantendo sobre todo o desenvolvimento relações sólidas com os clientes.

O enfoque do QFD é alcançar a garantia da qualidade durante o desenvolvimento do produto, sendo sub-dividido em Desdobramento da Qualidade (QD) e Desdobramento da Função Qualidade (QFDr). A Figura 2 mostra a relação entre QFD, QD e QFDr (CHENG et al., 1995).

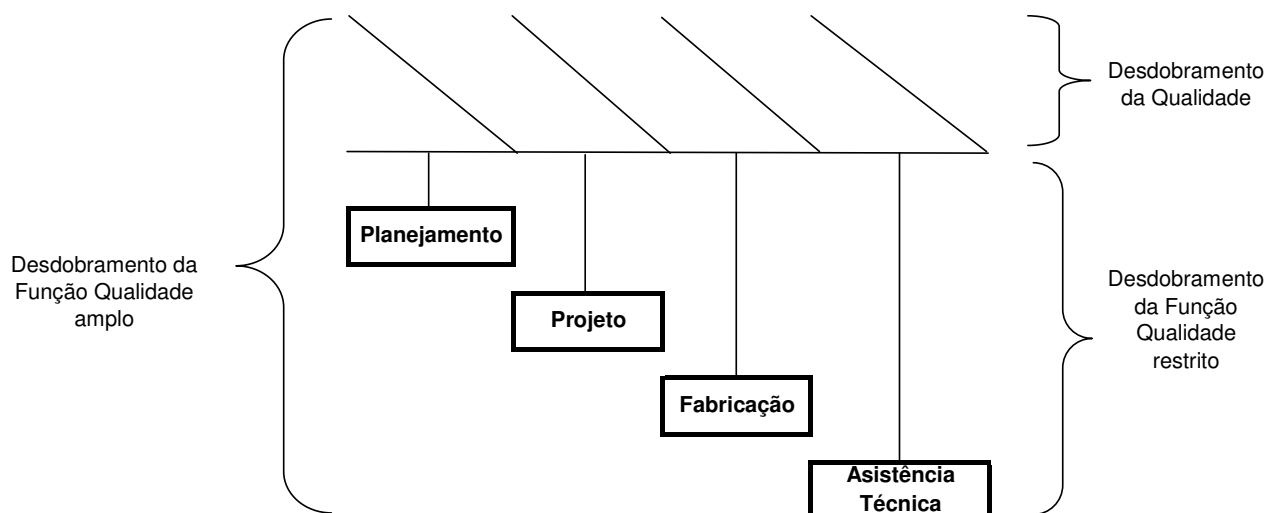


Figura 2 – Relação entre QFD, QD e QFDr

Fonte: Akao (1990)

Cheng et al. (1995) diz que o QFD possui duas partes constituintes, ou seja: QFD amplo é igual ao QD somado ao QFD restrito, onde:

- O QD visa o desdobramento da qualidade através da lógica da causa e efeito, partindo da voz do cliente e passando pelas características da qualidade do produto e obtendo um parâmetro de controle do Padrão Técnico de Processo;

- O QFD restrito visa o desdobramento da função trabalho, ou seja, um conjunto de procedimentos gerenciais e técnicos, formando o Padrão Gerencial do Desenvolvimento do Produto.

A metodologia QFD facilita e agiliza a troca de conhecimento entre as empresas e os clientes, sendo um instrumento para motivação e orientação no esforço de aprimoramento da gestão do desenvolvimento do produto, na qual a necessidade de se trabalhar em equipe é fundamental (AKAO, 1990).

O Sistema de Qualidade é o controle e a rede de procedimentos necessários para produzir e entregar os produtos dotados de padrão de qualidade especificada (AKAO, 1996).

O Desdobramento da Qualidade (DQ) pode ser definido como sendo a *conversão* das exigências dos usuários em características substitutivas (características de qualidade), definir a qualidade do projeto do produto, desdobrar esta qualidade em qualidades de outros itens tais como: qualidade de cada uma das peças funcionais, qualidades de cada parte e até os elementos do processo, apresentando sistematicamente a relação entre os mesmos (AKAO, 1996).

Segundo o Doutor Shigeru Mizuno, o Desdobramento da Função Qualidade no sentido restrito pode ser definido como sendo o desdobramento em detalhes das funções profissionais ou dos trabalhos que formam a qualidade, seguindo a lógica de objetivos e meios (AKAO, 1996).

Juran (1997) define o Desdobramento da Qualidade como sendo funções que formam a qualidade.

Miguel (2006) pondera que QFD tem sido amplamente utilizado para desenvolver novos produtos, porém nota-se que a utilização do QFD tem evoluído de forma eficaz para outras aplicações como: desenvolvimento de serviços, processos e outras aplicações não convencionais.

As organizações definem suas estratégias corporativas e de negócios para serem competitivas em um ambiente cada vez mais concorrido e o QFD é um importante método na elaboração e desdobramento das estratégias. Uma das aplicações não convencionais da utilização da metodologia QFD no planejamento estratégico é a formulação de estratégias de produção (MIGUEL, 2006).

O QFD contribui com a tomada de decisões complexas que envolvem o planejamento estratégico nas empresas; além disso, o método é flexível, permitindo ser customizado para diversas situações, podendo ser aplicado para as prioridades de produção e planos de ação (MIGUEL, 2006).

A aplicação da metodologia QFD na criação de um método de aferição de qualidade de produtos de *software* mostrou-se capaz em desdobrar a qualidade esperada pelo usuário em medidas. Com essas medidas é possível acompanhar a qualidade do produto desde a sua especificação, pela fase de sua construção até a entrega ao cliente. O acompanhamento, através das medidas, auxilia na identificação de falhas no produto e por consequência aumenta a satisfação do seu usuário. Além disso, a mensuração da qualidade desde o início do desenvolvimento antecipa a descoberta de erros, diminuindo o custo de suas correções (CASTANHO(1) et al., 2007).

O desdobramento da qualidade esperada pelo usuário através da aplicação da metodologia QFD possibilita a priorização das atividades de garantia da qualidade. A importância desta priorização deve ser ressaltada, pois no desenvolvimento de *software* o

investimento em qualidade precisa ser planejado para não extrapolar custos e alcançar seus objetivos de satisfazer as expectativas do usuário. A metodologia auxilia na definição das medidas adequadas a qualidade esperada pelo usuário possibilitando assim uma melhor priorização das atividades de garantia da qualidade (CASTANHO(2) et al., 2007).

Através da técnica QFD é possível identificar as qualidades demandadas por cada cliente e avaliar se estes possuem expectativas em comum (OLIVEIRA, 2006).

Segundo Castanho(3) (2007) a metodologia QFD é uma ferramenta do *Lean Six Sigma* que quando aplicada em uma abordagem de gestão e aferição da qualidade de produtos de *software* permite minimizar as incertezas e riscos de um projeto, podendo ser utilizada nas fases iniciais do projeto, durante o seu desenvolvimento até a entrega dos produtos de *software*.

A Matriz da Qualidade (Figura 3), que exerce um papel importante no desdobramento da qualidade, é definida como sendo a “sistematização das qualidades verdadeiras (exigidas pelos clientes), considerando principalmente as funções, e expressa a relação existente entre essas funções e as características da qualidade, que são características substitutivas” (AKAO, 1996).

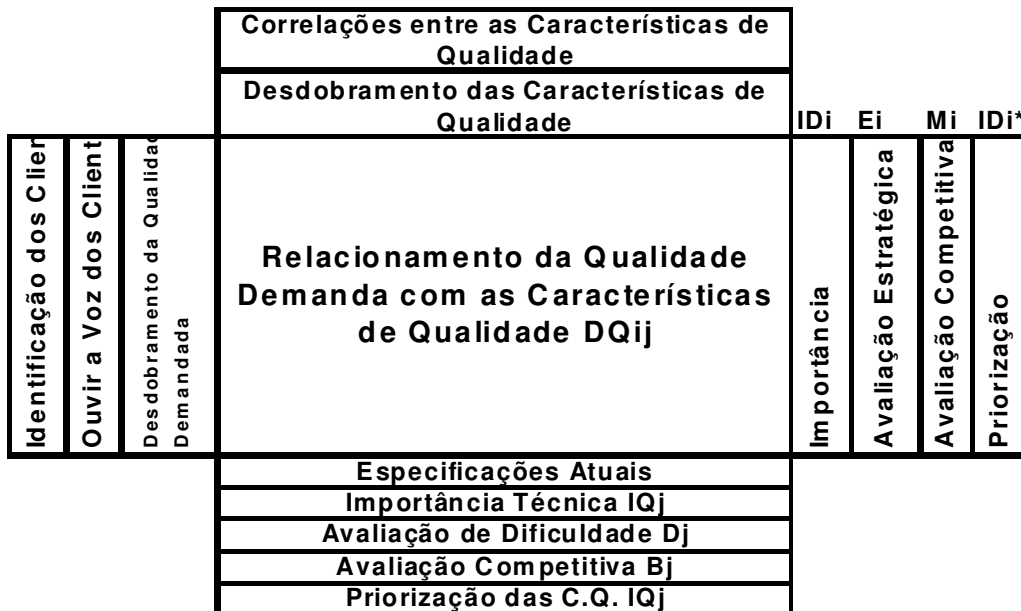


Figura 3 – Matriz da Qualidade

Fonte: Ribeiro (2000)

Onde:

- IDi - Índice de importância da qualidade demandada;
- IDi* - Índice de importância corrigida da qualidade demandada;
- Ei - Avaliação estratégica dos itens de qualidade demandada;
- Mi - Avaliação competitiva dos itens da qualidade demandada.

Existe um grande esforço das empresas no sentido de implementar as ações gerenciais para manter a qualidade de seus produtos e serviços para melhor atender o mercado consumidor (EUREKA, 1992).

Clark (1991) cita que o desempenho de uma empresa no desenvolvimento de produtos pode ser avaliado por três parâmetros básicos: produtividade, qualidade e tempo, que devem ser otimizados para capacitar uma empresa na sua habilidade de

atrair e satisfazer seus clientes, aumentando a competitividade de seus produtos, tendo como foco exceder as expectativas de seus clientes.

Ribeiro (2001) cita algumas breves definições para o QFD, como segue:

- É uma técnica de gestão auxiliando no gerenciamento de projetos simples e complexos;
- É um método de planejamento deslocando os esforços da engenharia para a fase de planejamento;
- É um método de resolução de problemas, abordando “O QUÊ” necessita ser feito e “COMO”;
- Facilidade na modelagem do conhecimento;
- Documentação e transporte de informações por meio de matrizes;
- Estimula a criatividade e inovações em um ambiente de Engenharia Simultânea.

O principal objetivo do desdobramento da função qualidade é assegurar que o projeto final de um produto ou serviço realmente atenda as necessidades de seus clientes. Os clientes podem não ter sido considerado explicitamente desde a etapa de geração do conceito, e por isso é adequado verificar se o projeto do produto ou serviço atenderá a essas necessidades (SLACK et al., 2002).

O QFD busca as necessidades dos clientes e conduz as informações ao longo de todo o processo e devolve para o cliente um produto ou serviço expressado por meio de requisitos de qualidade (RIBEIRO, 2001).

Segundo Liphaut (2006), a aplicação da metodologia QFD em pequenas empresas é viável, aumentando a satisfação do cliente quando do surgimento de um novo produto capaz de satisfazer os objetivos e expectativas dos clientes.

Slack et al., (2002) adverte que os detalhes do QFD podem variar, mas o princípio é comum, a saber, identificar os requisitos do consumidor para um produto ou serviço

(juntamente com sua importância relativa) e relacioná-los às características de projeto que traduzem tais requisitos na prática. Esse princípio pode ser estendido ao fazer o “como” de um estágio tornar-se o “que” do próximo (Figura 4).

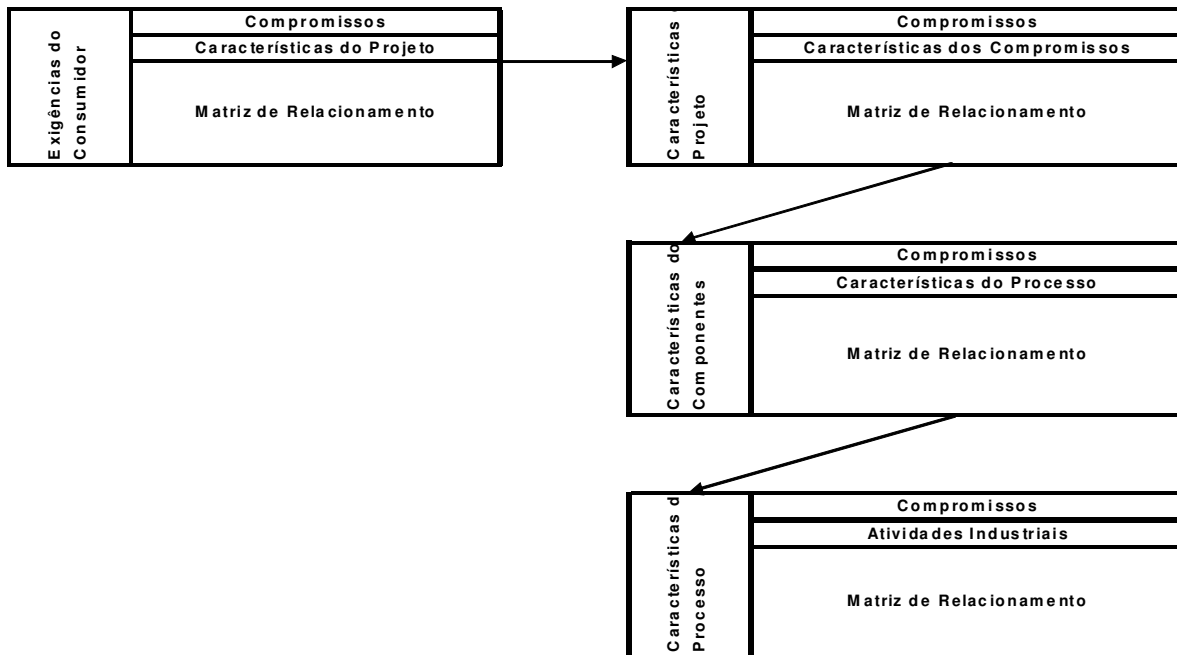


Figura 4 – Associação das Matrizes QFD

Fonte: Slack et al. (2002)

1.3.4 Abordagens do QFD

O desdobramento das necessidades dos clientes até o estágio de produção é feito através do uso de matrizes de relacionamento e de priorização. Este *modelo conceitual* representa o caminho por onde o estudo deve percorrer para alcançar o objetivo desejado, ou seja, é um plano de trabalho que direciona todo o processo de desdobramento (FIATES, 1995).

O Desdobramento da Função Qualidade é um método conhecido e utilizado nos Estados Unidos e Europa, porém esta denominação define em termos de conteúdo é restrito ao Desdobramento da Qualidade (DQ). Nas empresas japonesas a denominação é *Hinshitsu Kino Tenkai* a qual é subdividida em QD e QFD_r, onde o entendimento está

ligado ao planejamento da qualidade e ao sistema de garantia da qualidade (CHENG et al., 1995).

Nos Estados Unidos, existem duas versões distintas adotadas por instituições diferentes. A primeira versão é a adotada pelo *American Supplir Institute (ASI)*, sendo representada por quatro desdobramentos principais: planejamento do produto, desdobramento dos componentes, planejamento do processo e planejamento da produção. Este método é uma versão simplificada incluindo somente os desdobramentos da qualidade de acordo com a divisão do professor Akao (CHENG et al., 1995).

Conforme King (1989), a segunda versão é difundida pelo *Goal/QPC* e se originou nos trabalhos do professor Akao. Esta versão contempla também somente o desdobramento da qualidade através do desdobramento sistemático de matrizes ao invés de tabelas.

1.3.5 Abordagem de Akao

A abordagem do QFD por Akao é a mais abrangente, sendo aplicada em várias situações de desdobramento, em que cada caso terá seu *modelo conceitual* próprio (SPINOLA, 2000).

Segundo Spinola (2000), os desdobramentos são:

- Desdobramento da qualidade;
- Desdobramento das funções;
- Desdobramento dos mecanismos;
- Desdobramento da produção;
- Desdobramento da tecnologia;
- Desdobramento da confiabilidade;
- Desdobramento do custo.

Conforme Fiates (1995), a abordagem de Akao contempla quatro perspectivas distintas de desdobramento: desdobramento da qualidade. Desdobramento da tecnologia, custo e da confiabilidade. A Figura 5 resume o modelo proposto por Akao.

	Qualidade		Tecnologia	Custos	Confiabilidade
Produto	Qualidade Demanda X Características do Produto		Requisitos do Consumidor X Desdobramentos do Mecanismo	Avaliação do Mercado Preço / Participação Lucro	Qualidade Demandada X Árvore de Falhas
Sistemas	Desdobramentos da Função X Características do Produto	Desdobramentos da Função X Características Demandada	Desdobramento da Função X Desdobramento das Características X Desdobramento do Mecanismo	Estudo de Gargalo	Desdobramento da Função X Árvore de Falhas X Desdobramento das Características
Partes	Componentes X Características do Produto		Componentes X Desdobramento do Mecanismo	Desdobramento do Custo das Partes	Álise de Falhas
	Produção		Métodos de Desdobramento / Cartas de Controle e Garantia da Qualidade		

Figura 5 – Modelo de QFD de Akao (Adaptado de AKAO, 1990)

Fonte: Akao (1990)

1.3.6 Abordagem de Bob King

Bob King reorganizou o sistema de Akao agrupando todas as matrizes em uma única matriz, chamada de Matriz das Matrizes e esquematizou os desdobramentos de maneira ordenada seguindo uma seqüência. Finalmente, introduziu o fator inovação na metodologia QFD. No modelo de King, estão dispostas 30 matrizes. A Figura 6 ilustra o modelo conceitual de Bob King (FIATES, 1995).

	A Características da Qualidade		B Custos		C Mecanismos		D Modo de Falha do Produto		E Novas Concepções	F Engenharia de Valor
Demanda dos Clientes	A1	Demanda dos Clientes	B1	Novas Tecnologias	C1	Demanda dos Clientes	D1	Demanda dos Clientes	E1	F1
	Características da Qualidade		Custos		Mecanismos		Modo de Falha do Produto		Novas Concepções	FMEA
Funções	A2	Análise dos Componentes	B2	Funções	C2	Funções	D2	Funções	E2	F2
	Características da Qualidade		Alvos de Rompimento		Mecanismos		Modo de Falha do Produto		Novas Concepções	Análise de Valor
Características da Qualidade	A3	Características da Qualidade	B3	Características da Qualidade	C3	Características da Qualidade	D3	Características da Qualidade	E3	F3
	Características da Qualidade		Características da Qualidade		Mecanismos		Modo de Falha do Produto		Novas Concepções	Desdobramento
Partes	A4	Partes Críticas	B4	Partes	C4	Partes	D4	Resumo	E4	F4
	G1		G2		G3		G3		G5	G6
	Planilha da Garantia da Qualidade		Desdobramento		Planejamento		FTA		FEMA	Carta de CQ no Processo

Figura 6 – Modelo conceitual de Bob King (adaptado de King, 1989)

Fonte: King (1989)

Existe uma seqüência de matrizes diferentes pra cada objetivo específico: confiabilidade, custos, inovação, métodos de produção. A Tabela 1 ilustra esta seqüência (KING, 1989).

Tabela 1 – Seqüência de matrizes

OBJETIVO	MATRIZES
Análise das demandas do consumidor	A1,B1,D1,E1
Funções críticas	A2,C2,D2,E2
Definir características de qualidade	A1,A2,A3,A4,B3,B4,C3,D3,E3
Identificar partes críticas	A4,B4,C4,E4

Fonte: King (1989)

1.3.7 Abordagem de Makabe

Esta é a abordagem das quatro matrizes, proposta inicialmente por Makabe, no Japão, através do Instituto de Tecnologia de Tóquio sendo mais utilizado nos Estados Unidos (EUREKA, 1992).

Este método foi difundido devido a sua simplicidade, tendo como principais disseminadores nos Estados Unidos nomes como: Don Clausing, John Hauser e o *American Supplier Institute (ASI)* (HAUSER e CLAUSING, 1988).

Conforme Hauser e Clausing (1988), as quatro fases propostas por seu *modelo conceitual* são:

- Matriz I - Casa da Qualidade: planejamento do produto;
- Matriz II - Desdobramento das partes: projeto do produto;
- Matriz III - Planejamento do processo;
- Matriz IV - Planejamento da produção.

Segundo Fiates (1995), devido a facilidade de acesso a bibliografias americanas no Brasil, este método tem recebido o maior número de adeptos. As quatro fases desta abordagem se constituem em quatro matrizes orientando o desenvolvimento do produto ou serviço, desde os requisitos dos consumidores até a fabricação como mostra a Figura 7.

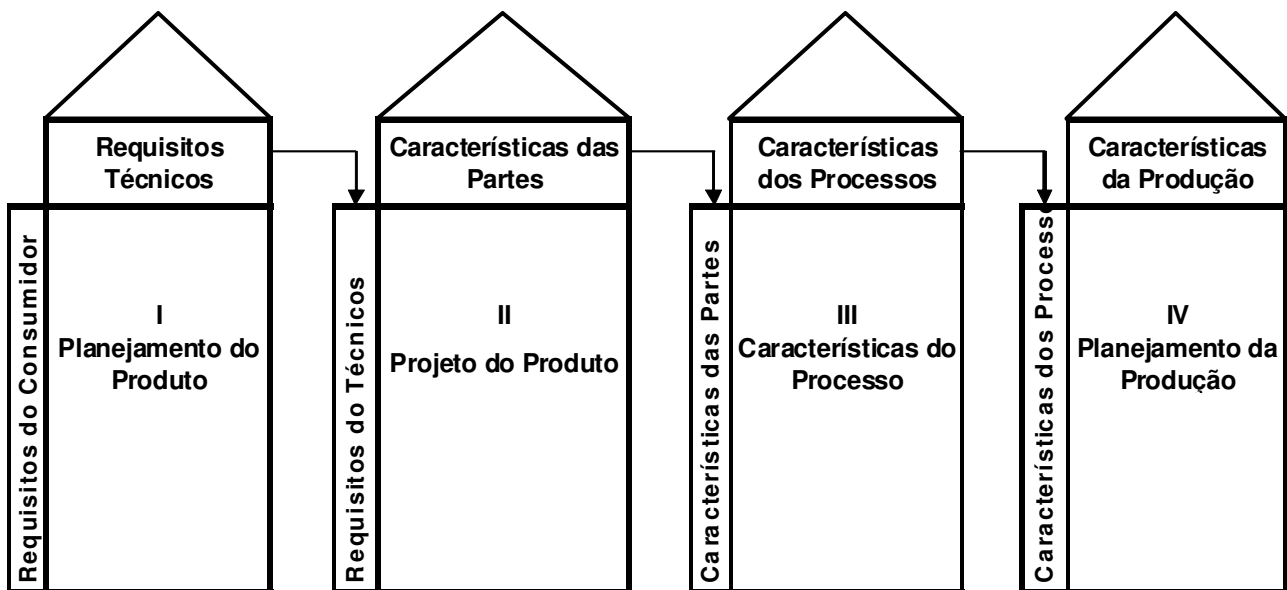


Figura 7 – Representação da abordagem de Makabe

Fonte: Hauser e Clausing (1988)

Conforme Hauser e Clausing (1988), o relacionamento estabelecido entre as matrizes é rígida, devendo seguir da Matriz I até a Matriz IV. Seu desenvolvimento é iniciado com os Atributos do Consumidor e em seguida são desdobradas as Características de Engenharia, formando-se a matriz. Seguindo-se a mesma seqüência para as matrizes seguinte, vai-se completando o modelo.

A abordagem de Makabe é mais indicada para as necessidades específicas de melhoria se um produto já existe ou para o desenvolvimento de um novo produto pouco complexo (MARTORANO, 1993).

1.3.8 A Casa da Qualidade

A casa da qualidade (ou matriz da qualidade) não só está presente nos modelos de QFD, como inicia os desdobramentos. Mais importante ainda, essa matriz é ferramenta básica de projeto do QFD (HAUSER e CLAUSING, 1988).

A casa da qualidade inicia o importante processo de “Tradução da Voz do Cliente”, onde todas as informações pesquisadas são organizadas nas matrizes para fazer a diagramação de priorização de recursos e ações para definição de estratégias e resolução de problemas (AKAO, 1996).

Conforme Akao (1996), obtém-se a casa da qualidade através do cruzamento dos requisitos do cliente, ou da qualidade exigida, ou qualidade demandada com as características da qualidade, sendo o resultado de saída as especificações do produto, ou seja, o conjunto de características técnicas do produto com suas qualidades projetadas (especificados), sendo então, os requisitos dos clientes a entrada da casa da qualidade e as características da qualidade e saída.

A casa da qualidade pode ser definida como a matriz que tem a finalidade de executar o projeto da qualidade, sistematizando as qualidades verdadeiras exigidas pelos clientes por meio de expressões lingüísticas, convertendo-as em características substitutas e mostrando a correlação entre essas características substitutas (características de qualidade) e as qualidades verdadeiras (AKAO, 1996).

Conforme Guinta (1993), a casa da qualidade é a matriz do QFD mais utilizada. A Figura 8 ilustra a matriz da qualidade.

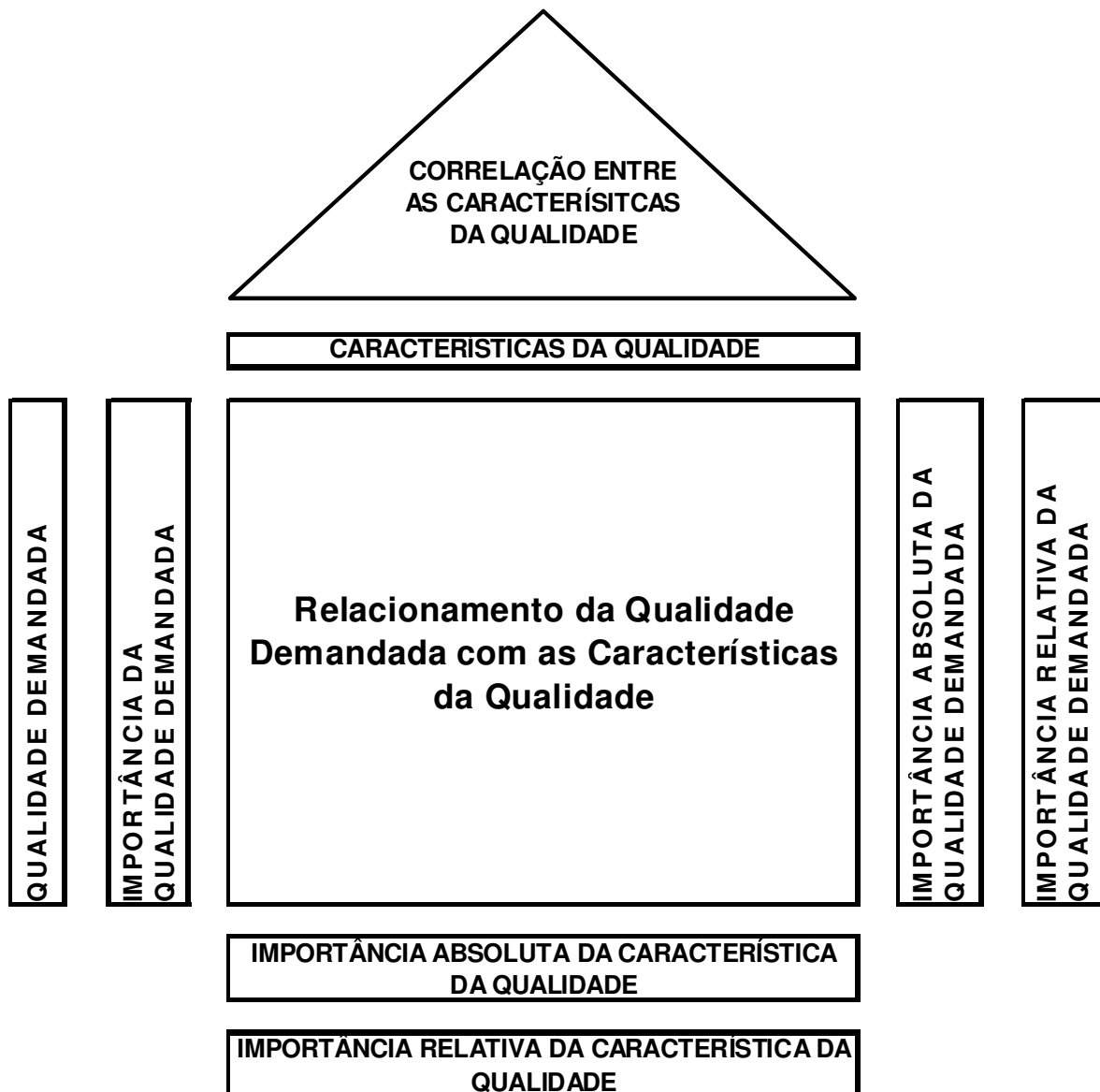


Figura 8 – Matriz da Qualidade Teórica (adaptada de Guinta, 1993)

Fonte: Guinta (1993)

Cheng et al. (1995) diz que a matriz da qualidade é formada por três atividades: sistematização das qualidades verdadeiras exigidas pelos clientes, transformação das qualidades exigidas pelos clientes em características de qualidade ou técnicas e as relações entre a qualidade demandada e as características de qualidade.

Cohen (1991) propõe a construção da casa da qualidade seguindo os seguintes passos:

- Descrever as necessidades dos clientes;
- Montar e analisar a matriz de planejamento;
- Obter as características de qualidade;
- Obter e analisar os relacionamentos;
- Obter e analisar as *correlações*;
- Descrever e analisar a concorrência;
- Definir objetivos;
- Planejar o desenvolvimento.

1.3.9 Elementos da Casa da Qualidade

Conforme Cheng et al. (1995), as etapas para a elaboração da Matriz da Qualidade possuem seus principais componentes descritos a seguir:

- **Qualidades exigidas (Requisitos dos Clientes)**

Os requisitos dos clientes são as expressões lingüísticas dos clientes convertidas, qualitativamente, em necessidades reais.

São obtidas por meio de pesquisas de mercado diretamente pelos clientes. Estes requisitos são organizados em um diagrama de afinidades e dispostos em um diagrama de árvore (MOURA, 1994).

- **Grau de importância (Cliente)**

É a identificação do grau de importância que os clientes atribuem a cada requisito, sendo obtido diretamente com os clientes, que mencionam uma pontuação a cada requisito.

Conforme Akao (1996), essa escala pode ser relativa ou absoluta. A escala é relativa quando o cliente indica a importância de cada requisito em comparação aos demais, ou seja, este requisito é mais importante que aquele. A escala é absoluta quando o cliente analisa a influência de cada requisito em sua decisão de compra do produto, ou seja, sem compará-lo com os demais.

- **Grau de Importância (Interno da Empresa)**

É o enquadramento dos requisitos dos clientes, através da pontuação de uma escala numérica, baseando-se em um dos cinco tipos de qualidade classificados por Kano (1998). Os requisitos dos clientes expressam as qualidades verdadeiras. A empresa deve saber a classificação de cada requisito e considerar que: a comparação entre produtos é obtida através da avaliação das qualidades lineares, as qualidades óbvias só são percebidas quando ausentes e as qualidades excitantes seduzem os clientes e faz com que o produto não seja comparado racionalmente.

- **Grau de Importância (Necessidades Futuras)**

Conforme Akao, (1996), é a previsão da importância dos requisitos quando o produto for lançado no mercado. As pessoas mudam suas necessidades e valores com o passar dos tempos. Produtos que requerem um longo período de desenvolvimento podem não possuir mais o grau de importância obtido com as pesquisas de mercado e desta forma, a empresa deve estimar a importância que os clientes darão, no futuro, a cada requisito.

- **Grau de Importância (Geral)**

É o valor final de cada requisito como resultado da análise dos três itens anteriores. Somente o grau de importância geral será considerado nos cálculos dos pesos relativo e absoluto, sendo que este cálculo é realizado por meio de uma análise qualitativa (AKAO, 1996).

- **Avaliação competitiva do cliente (Empresa x Concorrentes)**

É uma pesquisa quantitativa que identifica como os clientes percebem o desempenho do produto atual da empresa quando comparado aos principais concorrentes. A empresa deve saber com precisão qual é o desempenho do produto e quais as suas características que determinam esse desempenho. A partir desse conhecimento, e da avaliação do cliente para o produto atual da empresa, pode-se estabelecer uma comparação de características do produto e a satisfação do cliente. Isto servirá para a análise da pontuação dos produtos concorrentes e projeção da qualidade do produto em desenvolvimento (AKAO, 1996).

- **Plano de Qualidade (Requisitos)**

É o plano de qualidade que faz a inserção da estratégia da empresa no planejamento do produto. O plano de qualidade deve ser definido após a análise dos seguintes itens: avaliação competitiva do cliente, argumento de vendas e importância do requisito (AKAO, 1996).

- **Melhoria**

É a maneira de pontuar a importância final dos requisitos (peso absoluto e relativo) ao plano estratégico da empresa. Esse índice reflete quantas vezes o produto precisa melhorar seu desempenho, em relação ao produto atual, para alcançar a situação planejada (AKAO, 1996).

- **Ponto de Venda**

Conforme Cheng et al. (1995), são os principais benefícios que o produto fornecerá aos clientes visando o atendimento de suas necessidades.

Para Akao (1996), significa o grau de consonância dos requisitos dos clientes com a política da empresa para o mercado alvo.

- **Peso Absoluto (Requisitos)**

Representa a prioridade de atendimento de cada requisito sob os esforços de melhoria que devem ser concentrados em três pontos: requisitos mais importantes, requisitos que a empresa precisa melhorar (AKAO, 1996).

- **Peso Relativo (Requisitos)**

É a *conversão* do peso absoluto em percentagem, através da divisão do peso absoluto de cada requisito pelo resultado da soma de todos os pesos absolutos. O peso relativo tem por objetivo facilitar a percepção da importância relativa dos requisitos (AKAO,1996).

Conforme Cheng et al. (1995), a Tabela 2 representa os requisitos dos clientes.

Tabela 2 – Requisitos dos Clientes (adaptado de Cheng et al., 1995)

Requisitos dos Clientes	Grau de Importância				Avaliação dos Clientes		Qualidade Planejada				
	Cliente	Interno	Futuro	Geral	Empresa Hoje	Concorrentes	Plano Qualidade	Melhoria	Ponto de Venda	Peso Absoluto	Peso Relativo

Fonte: Cheng et al. (1995)

1.3.10 Tabela das Características de Qualidade

É a tabela das características de qualidade como um arranjo sistemático, baseado em um diagrama de árvore lógico, das características de qualidade que constituem um produto ou serviço (AKAO, 1990).

Cheng et al. (1995) diz que as características técnicas do produto podem ser divididas em “elementos da qualidade” e “características de qualidade”. Os elementos da qualidade são itens não quantificáveis, capazes de avaliar a qualidade do produto. Já as características de qualidade são definidas como os itens que devem ser medidos no produto para verificar se a qualidade exigida está sendo cumprida. A Figura 9 ilustra as características da qualidade.

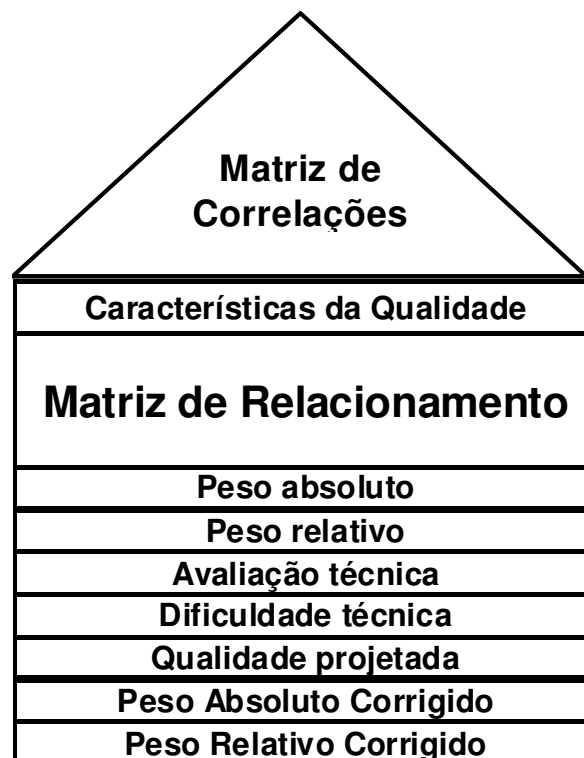


Figura 9 – Características da Qualidade

Fonte: Cheng et al. (1995)

- **Características de Qualidade**

As características de qualidade são características técnicas, ou características substitutas, para o produto final (AKAO, 1996).

As características técnicas são divididas em: elementos da qualidade e características da qualidade. Os elementos da qualidade são itens não quantificáveis, capazes de avaliar a qualidade do produto. As características de qualidade são definidas como itens que devem ser medidos no produto verificando se a qualidade exigida está sendo cumprida (AKAO, 1990).

- **Matriz de Correlações**

É o teto da casa da qualidade e por ela é feito o cruzamento das características da qualidade sempre duas a duas para identificar como elas se relacionam. Estas relações podem ser de apoio mútuo, ou seja, quando o desempenho favorável de uma característica ajuda o desempenho favorável de uma característica prejudica o desempenho favorável de outra característica (CHENG et al., 1995).

- **Matriz de Relações**

Segundo Cheng et al. (1995), matriz de relações é a interseção dos itens da qualidade demandada pelos clientes com as características de qualidade. É composta de células identificando como e quando cada característica de qualidade influencia em cada item da qualidade demandada. A intensidade das relações é indicada nos seguintes níveis: forte, média, fraca e inexistente. Identificadas as relações, preenchida a matriz, é necessário verificar por meio de análise os seguintes aspectos:

- Itens da qualidade demandada, significando que alguma necessidade do cliente não está sendo levada em consideração no desenvolvimento, sendo necessário reformular as características da qualidade.

- Características da Qualidade que não possuem relacionamento com nenhum item da qualidade demandada podem ter sido mal especificadas e não se relacionam corretamente com o produto ou serviço em desenvolvimento.

- **Peso Absoluto**

Conforme Cheng et al. (1995), peso absoluto é a soma dos valores obtidos em cada coluna e colocando o resultado final dessa soma na célula correspondente. Indica a importância de cada característica de qualidade no atendimento do conjunto de requisitos dos clientes.

- **Peso Relativo**

Conforme Cheng et al. (1995), o peso relativo de cada característica da qualidade é obtido através da *conversão* dos valores de peso absoluto em pesos relativos percentuais, ou seja, dividindo o valor de cada coluna pelo somatório da linha de peso absoluto.

- **Avaliação Competitiva**

Segundo Cheng et al. (1995), o estabelecimento das especificações de projeto é feito sem fatos e dados, com base exclusivamente na experiência pessoal da equipe de projetos sem levar em consideração as necessidades de mercado. Deve ser feita uma medição dos valores das características da qualidade dos produtos da concorrência pelo menos nos principais produtos que competem no mercado alvo. Esta pesquisa se dá por meio de notas atribuídas pelos clientes para os produtos concorrentes.

- **Dificuldade Técnica**

Este fator é uma nota em função da dificuldade tecnológica que a empresa terá para obter o valor determinado para a qualidade projetada das características de qualidade, com a confiabilidade projetada e com o objetivo de custo (AKAO, 1996).

Segundo Ribeiro (2000) é importante avaliar os itens de infra-estrutura e recursos humanos com relação a custos com a finalidade de incluir na priorização a análise custo/benefício dos equipamentos ou tecnologias que estão sendo propostas para utilização na empresa.

Segundo Hauser e Clausing (1998), a dificuldade técnica determina quais são as características que exigirão maior comprometimento de esforços e recursos para obtenção da sua qualidade projetada. É usada na matriz da qualidade para corrigir o peso das características de qualidade.

- **Qualidade Projetada**

É a última etapa do modelo se concretizando o planejamento das melhorias da qualidade que reforçará o sistema de qualidade existente e levando em consideração os desenvolvimentos anteriores, a voz do cliente e aspectos referentes à concorrência, custo e dificuldade de implantação (RIBEIRO, 2000).

A Qualidade Projetada pode ser entendida como os valores-meta para as características da qualidade do produto considerando seu peso relativo e a comparação com as características da qualidade dos produtos da concorrência e os objetivos do projeto (CHENG et al., 1995).

- **Peso absoluto corrigido**

É o resultado da multiplicação do peso absoluto de cada característica de qualidade pelo fator de dificuldade técnica, ou seja, indica a importância de cada característica de qualidade no atendimento do conjunto de requisitos dos clientes (AKAO, 1996).

- **Peso relativo corrigido**

É a *conversão* do peso corrigido absoluto em percentual. É calculado dividindo o peso absoluto de cada característica de qualidade pelo resultado da soma dos pesos absolutos de todas as características de qualidade (AKAO, 1996).

A partir da casa da qualidade é possível avaliar e determinar a prioridade da qualidade demandada e as características de qualidade que se correspondem, sendo possível o planejamento da qualidade (AKAO, 1996).

1.3.11 Modelo conceitual do QFD

Modelo conceitual pode ser representado por uma seqüência de tabelas e matrizes as quais permitem a visualização existente entre as relações de componentes, mecanismos, processos, matérias-primas e outros fatores com a Qualidade Projetada. É o caminho por onde o desenvolvimento do projeto deve percorrer para atingir as metas estabelecidas. Para operacionalizar os desdobramentos ou refinamentos, são utilizadas tabelas, matrizes e modelos conceituais que são denominados de unidades básicas de trabalho (*UBTs*), mostradas na Figura 10 (CHENG et al., 1995).

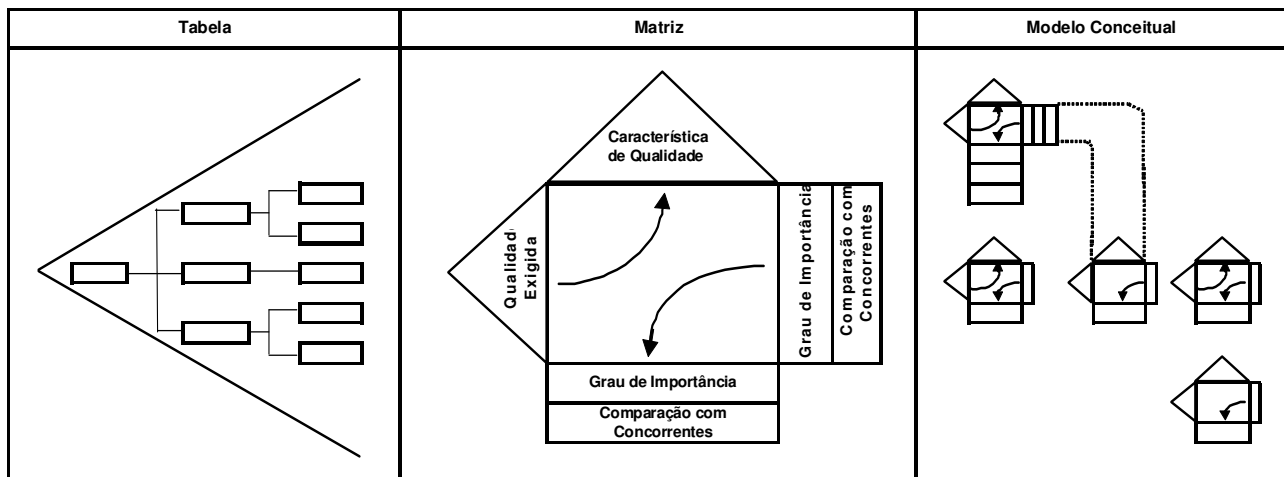


Figura 10 – Unidades Básicas do QFD (adaptado de Cheng et al. 1995)

Fonte: Cheng et al. (1995)

A tabela no QFD é considerada a unidade elementar, onde se registra o detalhamento de algo de forma organizada e ordenada em níveis, semelhante a um diagrama em árvore. Essa organização hierárquica é representada graficamente por um triângulo. O seu conteúdo e a origem de informação dependem do propósito para o qual é construída. Para confeccionar as tabelas, utilizam-se primeiramente, ferramentas de criatividade e participação como *Brainstorming*. Em seqüência, utiliza-se o Diagrama de Afinidade, de forma a agrupar as contribuições afins sob algum critério de relação (MIZUNO, 1993).

Ilustrado pela Figura 11 Mizuno (1993) explica que a partir de duas tabelas como A e B elabora-se uma matriz, com a finalidade de dar visibilidade às relações entre elas. As relações podem ser de três tipos: qualitativa, quantitativa e de intensidade, cujos processos de definição são denominados *extração* (seta 1), *conversão* (seta 2) e *correlação* (seta 3), respectivamente. Ainda na Figura 11, a tabela C, mostrada, representa a importância dos itens da tabela A. Já a tabela D representa o resultado obtido através do processo de *conversão*.

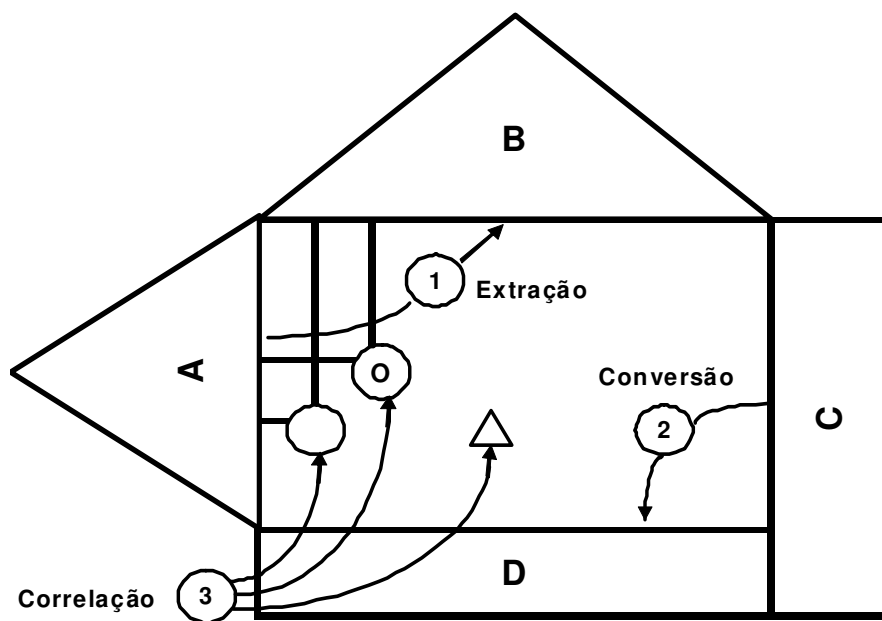


Figura 11 - Elementos de uma Matriz QFD (adaptado de Cheng et al. 1995)

Fonte: Cheng et al. (1995)

A *extração* acontece quando elementos de uma tabela são obtidos a partir de elementos de outra tabela. A *conversão* consiste em transmitir a importância dos elementos de uma tabela para outros elementos de outra tabela. Esse processo é posterior ao processo de *correlação*. A *correlação* visa identificar as relações entre os elementos desdobrados do último nível das tabelas (MIZUNO, 1993).

O *modelo conceitual* é um conjunto de tabelas e matrizes seqüenciadas de forma a permitir a visibilidade das relações existentes entre os componentes, mecanismos, processos, com a qualidade projetada para o produto. Representa o caminho por onde o desenvolvimento do projeto deve percorrer para atingir as metas estabelecidas. Um *modelo conceitual* completo contempla quatro dimensões de desdobramento: desdobramento da qualidade, da tecnologia, do custo e da confiabilidade. Entretanto, o tipo de *modelo conceitual* a ser construído é inteiramente dependente das metas, do tipo de empresa, da natureza do produto e da proximidade aos clientes (MIZUNO, 1993).

1.4 Mensuração em Software

Mensuração é o processo pelo qual números ou símbolos são associados a atributos de entidades no mundo real, com o objetivo de descrevê-la de acordo com um conjunto de regras claramente definidas. O principal propósito da mensuração da qualidade de *software* é fornecer resultados quantitativos referentes aos produtos de *software* para que estes sejam compreensíveis, aceitáveis e confiáveis por qualquer parte interessada. Alguns fatores que necessitam de medição para que sejam efetivos na avaliação de qualidade são a satisfação dos usuários e o retorno econômico (KUMAR, 2001).

Andersson (1990) expõe alguns conceitos relacionados com mensuração da qualidade de *software* e define um conjunto de métricas para os fatores de qualidade:

manutenibilidade e confiabilidade. Também apresenta conceitos gerais de mensuração de *software*, como o processo de identificação dos atributos a serem medidos, a validação das medidas selecionadas.

Viveiros (2006) explica que a norma ABNT (2003) apresenta várias medidas de qualidade de *software* organizadas de acordo com seu modelo de qualidade e as descreve detalhadamente em tabelas. Já o padrão IEEE-982 (1988) define um conjunto de métricas de atributos do produto, processo e recurso para mensurar a característica confiabilidade de *software*.

Pereira (2001) ressalta que além desses aspectos, devem-se mensurar também outras características de um *software*, como a satisfação, tamanho do produto dentre outras. Contudo geralmente é inviável mensurar todas as medidas em um projeto devido ao custo associado as suas coletas. Dessa forma, torna-se necessário priorizar as medidas que sejam adequadas a cada tipo de projeto de *software*.

Pereira (2001) explica que a implantação de um processo de mensuração no desenvolvimento de *software* deve priorizar as medidas a serem coletadas devido ao custo associado a sua coleta. Vários mecanismos podem ser utilizados para guiar o processo de seleção das medidas, destacando-se: o paradigma GQM (*Goal-Question-Metric*), o GDSM (*Goal-Driven Software Measurement*), o QFD (*Quality Function Deployment*) e o padrão IEEE-1061. Em linhas gerais, o que essas propostas pregam é que a seleção das medidas deve tomar como base os objetivos que se pretende atingir com a mensuração (PEREIRA, 2001).

O paradigma GQM seleciona as medidas desdobrando os objetivos de medição em questões quantificáveis e depois em medidas. Esse método é genérico o suficiente para ser utilizado para qualquer objetivo de medição. Porém, a definição de medidas para alguns objetivos, como por exemplo, produzir produtos com alta qualidade, depende da experiência em desdobrá-lo em questões quantificáveis. Dessa forma, para auxiliar o

desdobramento de alguns objetivos da organização seria necessário o auxílio de outro método para definir as medidas (LUIGI, 2006).

Luigi (2006) esclarece que o processo GDSM foi definido para desdobrar as metas da organização em indicadores e depois em objetivos de medição, que possam ser utilizados pelo método GQM. O GDSM define 10 passos para desdobrar as metas da organização até nas medidas para o processo de desenvolvimento. Porém, de acordo com o próprio GDSM, esses passos não são repetíveis e dessa forma, não garante que o processo chegue sempre ao mesmo resultado.

Viveiros (2006) explica que o padrão IEEE-1061 (1998) seleciona as medidas de qualidade através da identificação de fatores e subfatores de qualidade relacionados com o produto. A identificação dos fatores de qualidade mostra a preocupação de que a qualidade do produto possa ser influenciada por mais de uma característica de qualidade. Porém, o levantamento dos fatores de qualidade não é padronizado, dependendo da experiência do engenheiro de *software*.

Viveiros (2006) ressalta que a metodologia QFD pode ser usada para desdobrar a qualidade do produto esperada pelo usuário, no processo de desenvolvimento do *software*, envolvendo os responsáveis por cada fase. Esse desdobramento auxilia identificar os fatores que influenciam a qualidade do produto, e também, como deve ser mensurada essa qualidade durante a construção do *software*. A metodologia QFD pode ser aplicada de maneira formalizada e padronizada para qualquer produto, inclusive, produtos de *software*.

Na construção do método proposta neste trabalho, a metodologia QFD foi utilizada para desdobrar a qualidade esperada pelo usuário em medidas para o desenvolvimento de *software*. Conforme Fenton (1996), as medidas selecionadas por um processo de definição são utilizadas para mensuração de entidades e atributos do *software*. O número

de entidades envolvidas em *software* é grande devido à quantidade e à complexidade dos fatores envolvidos no desenvolvimento de um *software* de grande porte.

Ainda de acordo com Fenton (1996), existem três classes de entidades envolvidas: processo, recurso e produto, mostradas na Figura 12. Essas classes são subdivididas em outras e o conjunto de todas essas classes forma o *Framework de Atributos*.

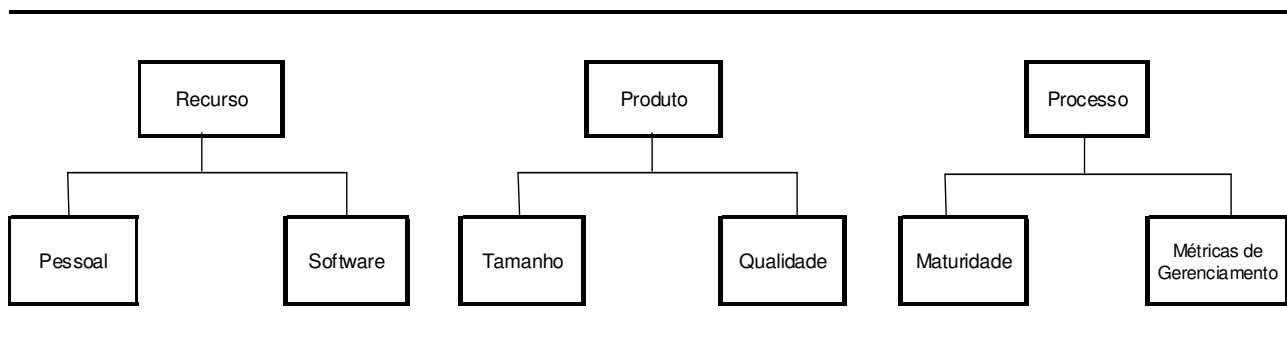


Figura 12 – *Framework* de atributos de Fenton

Fonte: Fenton (1996)

1.5 Norma ABNT (2003) - Engenharia de *Software* Qualidade de Produto

A “*International Organization for Standardization*” (ISO), em português “Organização Internacional para Normalização”, é uma entidade não governamental criada em 1947 e sediada em Genebra, na Suíça, cujo objetivo é o desenvolvimento da normalização e atividades relacionadas à intenção de facilitar o intercâmbio internacional de bens e serviços e desenvolver o cooperativismo intelectual, científico, tecnológico e econômico (RICCI, 1995).

A norma ABNT (2003) define um padrão de qualidade de produtos de *software* composto de duas partes: qualidade interna e externa e qualidade em uso. A primeira parte do modelo especifica seis características para qualidade interna e externa, as quais por sua vez são subdivididas em subcaracterísticas. Estas subcaracterísticas são

manifestadas exatamente, quando *software* é utilizado como parte de um sistema computacional, e são resultantes de atributos internos do *software*. A segunda parte do modelo especifica quatro características de uso, mas não apresenta o modelo de qualidade em uso além do nível de característica. Qualidade em uso é, para o usuário, o efeito combinado das seis características de qualidade dos produtos de *software* (ABNT, 2003).

O objetivo da norma ABNT (2003) é de garantir que a qualidade dos produtos de *software* seja especificada e avaliada em diferentes perspectivas pelos envolvidos com aquisição, requisitos, desenvolvimento, uso, avaliação apoio, manutenção, garantia de qualidade e auditoria de *software*. Alguns usos do modelo de qualidade definido pela norma ABNT (2003) são para:

- Validar a completude de uma definição de requisitos;
- Identificar requisitos de *software*;
- Identificar objetivos de projeto de *software*;
- Identificar objetivos para teste de *software*;
- Identificar critérios para garantia da qualidade;
- Identificar critérios de aceitação para produtos finais de *software*.

A Figura 13 ilustra segundo a norma ABNT (2003), que existem diferentes visões da qualidade do produto e de suas métricas em diferentes estágios do ciclo de vida do *software* (ABNT, 2003).

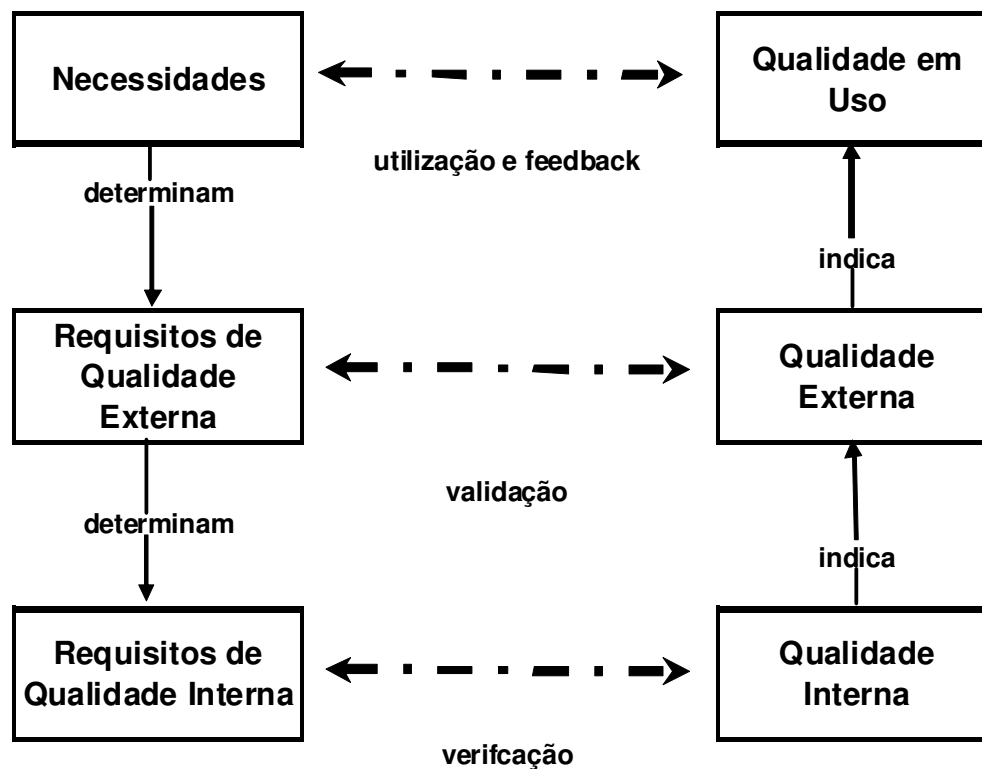


Figura 13 – Qualidade no ciclo de vida do *software*

Fonte: ABNT (2003)

1.5.1 Modelo de qualidade para qualidade externa e interna

Este modelo categoriza os atributos de qualidade de *software* em seis características (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade) as quais são, por sua vez, subdivididas em subcaracterísticas ilustradas na Figura 14. As subcaracterísticas podem ser medidas por meio de métrica externas e internas (ABNT, 2003).

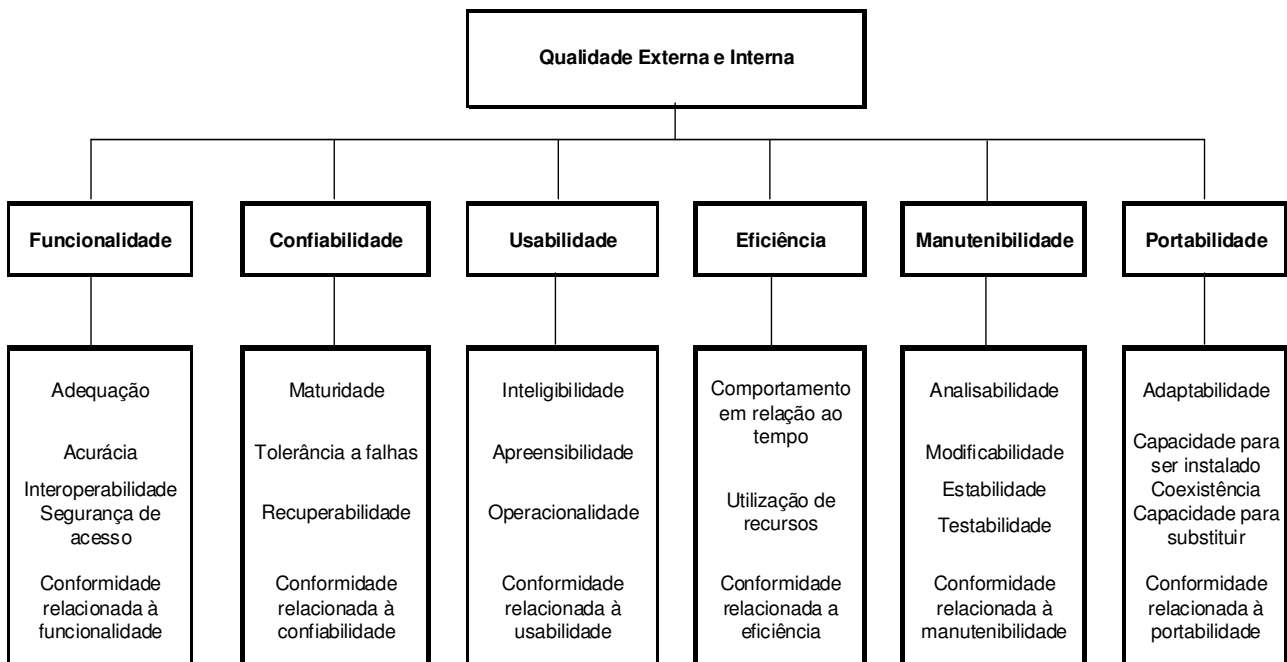


Figura 14 – Modelo de qualidade para qualidade externa e interna

Fonte: ABNT (2003)

Uma definição é atribuída para cada característica e para cada subcaracterística do *software* que influencia a característica de qualidade. A capacidade do *software* é determinada por um conjunto de atributos internos que podem ser medidos, para cada característica e subcaracterística. As características e subcaracterísticas podem ser medidas externamente pelo grau da capacidade do sistema contendo o *software*. Existe uma subcaracterística de conformidade para todas as características, já que os princípios são geralmente aplicáveis a todas as características de qualidade externa e interna. (ABNT, 2003).

Segundo a ABNT (2003), as características e suas subcaracterísticas são descritas na Tabela 3.

Tabela 3 – Características e subcaracterísticas de qualidade interna e externa

QUALIDADE INTERNA E EXTERNA DE PRODUTOS DE SOFTWARE ABNT (2003)	
CARACTERÍSTICAS	SUBCARACTERÍSTICAS
<p><u>Funcionalidade:</u></p> <p>Refere-se à existência de um conjunto de funções que satisfazem necessidades explícitas ou implícitas e suas propriedades específicas.</p>	<p>Adequação: atributos de <i>software</i> que influenciam na adequação de um conjunto de funções para tarefas específicas e objetivos de uso;</p> <p>Acurácia: atributos do <i>software</i> que evidenciam a presença de resultados ou efeitos corretos ou conformidade acordada;</p> <p>Interoperabilidade: atributos de <i>software</i> que influenciam na habilidade de interagir com um ou mais sistemas específicos;</p> <p>Conformidade: atributos do <i>software</i> que influenciam na aderência e padrões relativos a convenções ou regulamentações legais e prescrições similares;</p> <p>Segurança de acesso: atributos de <i>software</i> que influenciam na habilidade de prevenir acessos não intencionados e resistir a ataques intencionados para se ter acesso não autorizado à informação confidencial, ou fazer modificações não autorizadas em informações ou em programa.</p>
<p><u>Confiabilidade:</u></p> <p>Refere-se à capacidade do <i>software</i> manter seu nível de desempenho, sob condições estabelecidas, por um período de tempo.</p>	<p>Maturidade: atributos de <i>software</i> que influenciam na frequência de erros devido às falhas no <i>software</i>.</p> <p>Tolerância à falhas: atributos de <i>software</i> que influenciam na habilidade de um nível específico de desempenho em casos de falhas do <i>software</i> ou por violação de sua interface específica;</p> <p>Recuperabilidade: atributos de <i>software</i> que influenciam sua capacidade de restabelecer seu nível de desempenho e recuperar os dados diretamente afetados no caso de ocorrer uma falha e no tempo e esforços necessários.</p> <p>Conformidade: A capacidade dos produtos de <i>software</i> a aderir a padrões, convenções e regulamentações relacionadas com confiabilidade.</p>
<p><u>Usabilidade:</u></p> <p>Refere-se ao esforço necessário ao uso dos produtos de <i>software</i>, bem como o julgamento individual de tal uso, por um conjunto explícito ou implícito de usuários.</p>	<p>Inteligibilidade: atributos de <i>software</i> que influenciam na capacidade do usuário entender se o <i>software</i> é adequado, e como ele pode ser usado para tarefas e condições de uso particulares;</p> <p>Aprensibilidade: atributos de <i>software</i> que influenciam a facilidade com a qual o usuário pode aprender suas aplicações;</p> <p>Operacionalidade: atributos de <i>software</i> que influenciam no esforço necessário para o usuário poder operar e manter o controle da operação;</p> <p>Atratividade: atributos de <i>software</i> que influenciam o interesse do usuário pelo uso do <i>software</i>;</p> <p>Conformidade: A capacidade dos produtos de <i>software</i> a aderir a padrões, convenções e regulamentações relacionadas com usabilidade.</p>
<p><u>Eficiência:</u></p> <p>Refere-se ao relacionamento entre o nível de desempenho do <i>software</i> e a quantidade de recursos utilizada, sob condições estabelecidas.</p>	<p>Comportamento em relação ao tempo: atributos do <i>software</i> que influenciam no tempo de resposta de processamento e desempenho na execução de suas funções;</p> <p>Utilização de recursos: atributos de <i>software</i> que influenciam na quantidade de recursos usados e a duração de tal uso na execução de suas funções.</p> <p>Conformidade: A capacidade dos produtos de <i>software</i> a aderir a padrões, convenções e regulamentações relacionadas com eficiência.</p>
<p><u>Manutenibilidade:</u></p> <p>Refere-se ao esforço necessário para fazer modificações específicas no <i>software</i>.</p>	<p>Analisabilidade: atributos de <i>software</i> que influenciam na necessidade de recursos para diagnóstico de deficiências ou causas de falhas, ou para identificação de partes a serem modificadas;</p> <p>Modificabilidade: atributos de <i>software</i> que influenciam na necessidade de recursos para implementar as modificações específicas;</p> <p>Estabilidade: atributos de <i>software</i> que influenciam nos riscos de efeitos inesperados das modificações;</p> <p>Testabilidade: atributos de <i>software</i> que influenciam na necessidade de recursos necessários para validação do <i>software</i> modificado;</p> <p>Conformidade: A capacidade dos produtos de <i>software</i> a aderir a padrões, convenções e regulamentações relacionadas com manutenibilidade.</p>
<p><u>Portabilidade:</u></p> <p>Refere-se à habilidade do <i>software</i> para ser transferido de um ambiente para outro.</p>	<p>Adaptabilidade: atributos de <i>software</i> que influenciam na capacidade e esforço necessário para sua adaptação em ambientes diferentes especificados, sem aplicar outros meios ou ações para atingir o propósito do <i>software</i>;</p> <p>Capacidade de instalação: atributos de <i>software</i> que influenciam nos esforços necessários para instalar o <i>software</i> no ambiente especificado;</p> <p>Capacidade de substituição: atributos de <i>software</i> que proporcionam a oportunidade e influenciam no esforço requerido para usá-lo em lugar de outro <i>software</i> específico no ambiente de tal <i>software</i>;</p> <p>Coexistência: atributos de <i>software</i> que requerem esforços para existir no mesmo ambiente que outros softwares sem sofrer influências dos mesmos ou influenciá-los;</p> <p>Conformidade: A capacidade dos produtos de <i>software</i> a aderir a padrões, convenções e regulamentações relacionadas com portabilidade.</p>

Fonte: ABNT (2003)

1.5.2 Modelo de qualidade para qualidade em uso

Os atributos de qualidade em uso são categorizados em quatro características: eficácia, produtividade, segurança e satisfação. Conforme ilustrado na Figura 15 (ABNT, 2003).

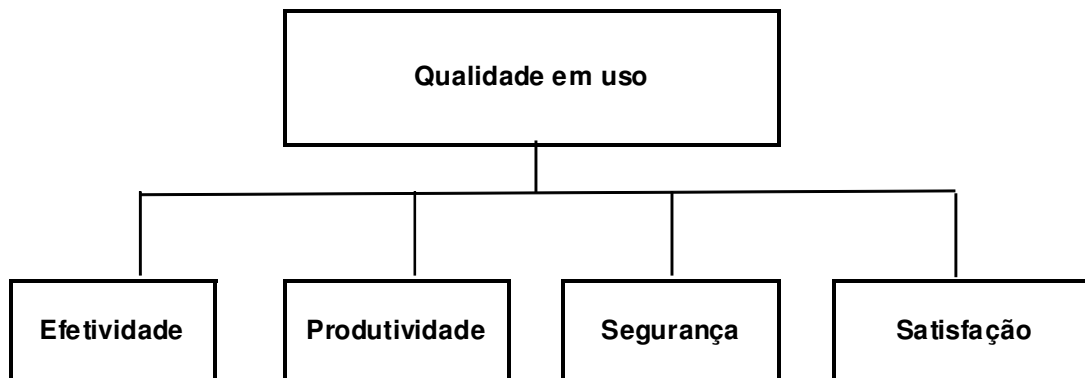


Figura 15 - Modelo de qualidade para qualidade em uso

Fonte: ABNT (2003)

Qualidade em uso é a visão da qualidade sob a perspectiva do usuário. A obtenção de qualidade em uso é dependente da obtenção da necessária qualidade externa, a qual, por sua vez, é dependente da obtenção da necessária qualidade interna. Normalmente, são necessárias medidas em todos os três níveis, pois atender aos critérios para medidas internas em geral não é suficiente para garantir o atendimento aos critérios para medidas externas e atender aos critérios para medidas externas de subcaracterísticas em geral não é suficiente para garantir o atendimento aos critérios para qualidade em uso.

Segundo a ABNT (2003), as características relacionadas com o desempenho do produto no seu uso pelo usuário são chamadas de características de qualidade em uso, descritas na Tabela 4.

Tabela 4 – Características de qualidade em uso

QUALIDADE EM USO DE PRODUTOS DE SOFTWARE ABNT (2003)	
CARACTERÍSTICAS	DESCRIÇÃO
<u>Efetividade</u>	Refere-se à capacidade dos produtos de <i>software</i> em possibilitar aos usuários atingir metas especificadas com acurácia e completeza, num contexto de uso especificado.
<u>Produtividade</u>	Refere-se à capacidade dos produtos de <i>software</i> em possibilitar aos usuários utilizar uma quantidade adequada de recursos em relação à efetividade alcançada, num contexto de uso especificado.
<u>Segurança</u>	Refere-se à capacidade dos produtos de <i>software</i> em propiciar níveis aceitáveis de risco de danos a pessoas, negócios, <i>software</i> , propriedade ou ambiente, num contexto de uso especificado.
<u>Satisfação</u>	Refere-se à capacidade dos produtos de <i>software</i> em satisfazer usuários, num contexto de uso especificado.

Fonte: ABNT (2003)

1.6 Metodologia de desenvolvimento de produtos de *software*

Koscianski et al. (2007) diz que uma metodologia de desenvolvimento de produtos de *software* é um conjunto de atividades que auxiliam a produção de produtos de *software*. O resultado dessas atividades é um produto que reflete a forma como todo o processo foi conduzido. Muitas organizações desenvolvem *software* sem utilizar nenhuma metodologia. O resultado desta falta de sistematização na produção de produtos de *software* é a baixa qualidade do produto final, além de dificultar a entrega do *software* nos prazos e custos predefinidos e inviabilizar futura evolução do *software*.

Dentre as várias metodologias, Pádua et al. (2003) citam o Praxis (Processo para Aplicativos Extensíveis e Interativos) como uma metodologia completa de desenvolvimento de *software*. Ele abrange tanto métodos técnicos, como requisitos, análise, desenho, testes e implementação, quanto os métodos gerenciais, como gestão de projetos, gestão da qualidade e engenharia de processos.

Ainda segundo Pádua et al. (2003), o Praxis é baseado na tecnologia orientada a objetos e seus métodos gerenciais cobrem as áreas de processos do CMMI (*Capability Maturity Model Integration*), que é um modelo de capacitação adotado por centenas das principais organizações produtoras de *software*. Os padrões do Praxis são conformes com os padrões de engenharia de *software* do IEEE (*Institute of Electrical and Electronic Engineers*), os mais abrangentes e respeitados da área. O Praxis foi experimentado em cursos de graduação, pós-graduação e treinamento industrial, e usado em processos reais de desenvolvimento de *software*.

O Praxis propõe um ciclo de vida composto por fases, divididas em uma ou mais interações e fluxos que produzem um conjunto precisamente definido de artefatos (documentos e modelos). A Tabela 5 apresenta a divisão em fases e interações, enquanto a organização em fluxos é descrita na Tabela 6 (PÁDUA et al., 2003).

Tabela 5 – Fase e interações do Praxis

FASE	INTERAÇÃO	SIGLA	DESCRIÇÃO
Concepção	Ativação	AT	Levantamento e análise das necessidades dos usuários e conceitos da aplicação, em nível de detalhe suficiente para justificar a especificação de um produto de <i>software</i> .
	Levantamento dos Requisitos	LR	Levantamento das funções, interfaces e requisitos não funcionais desejados para o produto em nível de detalhe suficiente para determinar a viabilidade de desenvolvimento de um produto.
Elaboração	Análise dos Requisitos	AR	Modelagem conceitual dos elementos relevantes do domínio do problema, validação dos requisitos e definição da arquitetura, em detalhe suficiente para o planejamento da fase de Construção.
	Desenho Implementável	DI	Implementação de um subconjunto crítico de funções do produto, para validar a arquitetura e tecnologias escolhidas, determinando a produtividade de desenvolvimento.
Construção	Liberação 1	L1	Implementação de um subconjunto de funções do produto que será avaliado pelos usuários.
	Liberação ...	Ln	Implementação de um subconjunto de funções do produto que será avaliado pelos usuários.
	Testes Alfa	TA	Realização dos testes de aceitação, no ambiente dos desenvolvedores, juntamente com elaboração da documentação de usuário e possíveis planos de Transição.
Transição	Testes Beta	TB	Realização dos testes de aceitação, no ambiente dos usuários.
	Operação Piloto	OP	Operação experimental do produto em instalação piloto do cliente, com a resolução de eventuais problemas por meio de processo de manutenção.

Fonte: Pádua et al., (2003)

Tabela 6 – Fluxos técnicos e gerenciais do Praxis

NATUREZA	FLUXO		DESCRIÇÃO
TÉCNICOS	Requisitos		Fluxo que visa a obter um conjunto de requisitos de um produto, acordado entre cliente e fornecedor.
	Análise		Fluxo que visa a detalhar, estruturar e validar os requisitos, em termos de um modelo conceitual do problema, de forma que estes possam ser usados como base para o planejamento e acompanhamento detalhados da construção do produto.
	Desenho		Fluxo que visa a formular um modelo estrutural do produto que sirva de base para a implementação, definindo os componentes a desenvolver e a reutilizar, assim como as interfaces entre si e com o contexto do produto.
	Implementação		Fluxo que visa a detalhar e implementar o desenho por meio de componentes de código e de documentação associada.
	Testes		Fluxo que visa a verificar os resultados da implementação, através do planejamento, desenho e realização de baterias de testes.
	Engenharia de Sistemas		Fluxo que abrange atividades relativas ao desenvolvimento do sistema no qual o produto de <i>software</i> está contido; por exemplo, modelagem de processos de negócio, implantação, usabilidade e criação de conteúdo.
GERENCIAIS	FLUXO	SUBFLUXO	DESCRIÇÃO
	Gestão de Projetos	Gestão de requisitos	Controle das alterações e rastreamento dos requisitos.
		Planejamento de projetos	Elaboração de planos de projetos, por meio de estimativas de tamanho, esforço, prazo e riscos.
		Controle de projetos	Acompanhamento do progresso e dos riscos dos projetos, com execução de procedimentos corretivos, quando necessários.
		Contratação de projetos	Especificação e acompanhamento de contratos de desenvolvimento por parte de terceiros.
	Gestão da Qualidade	Garantia da qualidade	Conjunto planejado e sistemático de ações necessárias para estabelecer um nível adequado de confiança na qualidade de um produto.
		Gestão de revisões	Planejamento, convocação e avaliação de revisões técnicas e inspeções.
		Gestão de configurações	Conjunto de procedimentos técnicos e gerenciais para identificação de artefatos e gestão de alterações destes.
		Gestão da manutenção	Conjunto de procedimentos para a manutenção dos produtos em Transição e Produção.
	Engenharia de Processos	Gestão de processos	Guarda, manutenção e personalização do patrimônio de processos da organização.
		Gestão do treinamento	Gestão das atividades de treinamento relacionadas com processos de <i>software</i> .
		Melhoria de tecnologia	Execução das atividades de evolução tecnológica relacionadas com processos de <i>software</i> .
		Melhoria de processos	Aferição, controle quantitativo e evolução dos processos de <i>software</i> .

Fonte: Pádua et al., (2003)

Segundo Pádua et al. (2003) as atividades desenvolvidas pelos fluxos técnicos e gerencias do Praxis geram e consomem artefatos (documentos e modelos), que são descritos na Tabela 7.

Tabela 7 – Artefatos do Praxis

ESPÉCIE	SIGLA	SIGNIFICADO
Modelos	CRSw	Cadastro dos requisitos
	MASw	Modelo de análise
	MPPSw	Memória de planejamento de projeto
	MDSw	Modelo de desenho
	BTRSw	Bateria de testes de regressão
	CFSw	Código fonte
	CESw	Código executável
Relatórios	LCSw	Listas de conferência
	RRSw	Relatórios de revisão
	RTSw	Relatório dos testes
	RAPSw	Relatório de acompanhamento de projeto
	RAQSw	Relatório de auditoria da qualidade
Documentos para o cliente	PESw	Proposta de especificação
	ERSw	Especificação dos requisitos
	PDSw	Plano de desenvolvimento
	PQSw	Plano da qualidade
	DDSw	Descrição do desenho
	DTSw	Descrição dos testes
	MUSw	Manual do usuário

Fonte: Pádua et al., (2003)

O Praxis originalmente foi destinado à utilização em projetos didáticos em disciplinas de Engenharia de *Software*, o processo atualmente vem sendo usado com sucesso também em projetos maiores, alguns envolvendo equipes de até dezenas de pessoas em grandes empresas de desenvolvimento de produtos de *software* (PÁDUA et al., 2003).

1.7 Medidas de qualidade para produtos de *software*

A classificação dos atributos de *software* possibilita organizar e priorizar as várias propriedades que, potencialmente, podem ser medidas no desenvolvimento de *software*. O número de propriedades envolvidas na construção de um *software* de grande porte é consideravelmente alto, devido à quantidade e a complexidade dos fatores envolvidos no seu processo de desenvolvimento. A coleta de medidas para cada fator envolve um custo e torna-se necessário restringir as medições a um conjunto limitado. Uma classificação de atributos é útil na seleção do subconjunto de referência de medidas a ser coletada para, de forma eficiente diminuir os custos e a complexidade do processo de desenvolvimento de *software* (JACOBSON, 1998).

Segundo Jacobson (1998), uma classificação de atributos de *software* padronizada auxilia na organização das medidas a serem utilizadas no processo de mensuração. A escolha adequada das medidas a serem coletadas é um fator importante para o sucesso de uma política de mensuração. Além disso, a classificação dos atributos de *software* possibilita que outras medidas, ainda não utilizadas, sejam organizadas e potencialmente utilizadas na mensuração dos atributos de um projeto.

Para classificar as medidas de *software* é preciso categorizar os atributos do desenvolvimento em grupos bem definidos. O *Framework de Atributos* definido por Fenton (1996) descrito no capítulo 1, identifica três classes de atributos de *software*: processo, recurso e produto. Cada uma dessas classes é subdividida em outras classes, por exemplo, a classe de produto é subdividida em qualidade, tamanho, complexidade entre outras. Essas classes têm uma descrição clara que possibilita classificar cada medida de acordo com seu propósito e suas entradas.

A classificação das medidas em uma estrutura de atributos de *software* possibilita a padronização entre os vários projetos de desenvolvimento de *software* de uma empresa. Dessa forma, a classificação serve de agente facilitador na criação da base

histórica de projetos da instituição. Os dados coletados referentes às medidas auxiliam na avaliação dos atributos de *software* durante um projeto, mas também, podem servir para a construção de uma base histórica para a organização. A base histórica possibilita, uma melhor estimativa de preço de novos projetos (JACOBSON, 1998).

1.7.1 Norma ABNT (2003) – Medidas de qualidade de software (I)

Existem vários atributos do desenvolvimento de *software* que precisam ser mensurados para avaliar a qualidade desejada pelo usuário. A qualidade do *software* é influenciada por vários fatores que variam desde a adequação das funcionalidades às necessidades do usuário até a facilidade de uso do produto. As informações coletadas sobre a qualidade do *software* podem ser organizadas por meio de um modelo de qualidade padronizado como o modelo descrito na norma ABNT (2003) (ABNT, 2003).

Segundo ABNT (2003), cada medida exemplificada na norma é descrita conforme os seguintes campos: *nome, propósito, método de aplicação, fórmula, interpretação do valor, tipo de escala, tipos, entradas e audiência alvo*. A descrição das medidas com esses campos permite documentá-las de forma clara e objetiva, e assim, melhorar a compreensão de tal medida pelo engenheiro de *software* que desejar utilizá-las. Estes campos são descritos a seguir:

- O campo nome é uma referência para medida que possibilita uma fácil comunicação entre as pessoas do projeto;
- O campo propósito descreve em poucas palavras o objetivo da coleta a medida;
- O campo método de aplicação define o processo de coleta da medida;
- O campo fórmula descreve a equação matemática para calcular o valor da medida;

- O campo interpretação do valor serve de referência para leitura do valor coletada da medida;
- O campo tipo de escala classifica a medida de forma que ela possa ser comparada com medidas de outros projetos;
- O campo tipos define o tipo de cada valor para o cálculo da medida;
- O campo entradas descreve os artefatos do processo de desenvolvimento envolvidos no cálculo da medida;
- O campo audiência alvo define as pessoas que devem utilizar o valor coletado da medida.

Entre as medidas internas da norma ABNT (2003) (ABNT-2003), são apresentadas algumas relevantes para este trabalho, sendo elas:

- Response Time (Tempo de Resposta);
- Functional Adequacy (Adequação Funcional);
- Fault Removal (Remoção de Falhas);
- Input Validity Checking (Checagem de Entrada de Dados);
- User Operation Cancellability (Operações Canceláveis).

As medidas de qualidade interna selecionadas estão descritas no Anexo 1 de acordo com o padrão de campos para classificação (ABNT, 2003).

1.7.2 Padrão IEEE-982 (1988) – Medidas de qualidade de software (II)

O padrão da IEEE (*Institute of Electrical and Electronic Engineers*) tem o objetivo de fornecer um conjunto de medidas utilizadas como indicador da característica Confiabilidade da qualidade de um *software*. As medidas descritas nesse padrão são organizadas conforme uma classificação própria. Algumas medidas do padrão IEEE-982 (1988) são apresentadas e demonstram a sua classificação, sendo elas:

- Cumulative Failure Profile (Confiabilidade do Perfil de Falhas);
- Requirements Traceability. (Ratreabilidade de Requisitos).

A medida “*Cumulative Failure Profile*” pode ser aplicada para prever a confiabilidade de um produto através do uso de perfis de defeitos, ou para identificar os módulos ou os subsistemas que requerem testes adicionais. Dessa forma, essa medida pode ser utilizada para prever a maturidade ou para mensurar a testabilidade de um produto. Essa medida é classificada na subcaracterística maturidade da característica confiabilidade e também classificada na subcaracterística testabilidade da característica manutenibilidade (IEEE-982, 1988).

A medida “*Requirements Traceability*” pode ser aplicada para identificar os requisitos que estão faltando ou fora do escopo dos requisitos definidos inicialmente. Podendo ser utilizada como indicativo da adequação das funcionalidades de um produto. Dessa forma, essa medida é classificada na subcaracterística Adequação da característica Funcionalidade do modelo de qualidade interna e externa do *Framework de Atributos* (IEEE-982, 1988).

As medidas do padrão IEEE-982 (1988) selecionadas estão descritas no Anexo 2 de acordo com o padrão de campos para classificação.

1.7.3 Metodologia Praxis – Medidas de qualidade de software (III)

Na metodologia Praxis, existem três situações típicas em que é possível coletar medidas de defeitos: procedimentos de revisão e auditoria e relatórios de teste. Nessas atividades, os defeitos são registrados nos relatórios RRSw (*Relatório de Revisão de Software*), RISw (*Relatório de Inspeção de Software*) e RTSw (*Relatório de Teste de Software*). Esses relatórios do Praxis são consolidados no relatório RAPS w (*Relatório de Acompanhamento de Projeto de Software*), visando o acompanhamento da medida de

defeito do projeto. A contagem de defeitos constitui a base de diversos indicadores de qualidade comumente utilizados, como confiabilidade, correção, completeza, eficiência e usabilidade. Dessa forma, a medida de defeito do Praxis influencia vários atributos do *Framework de Atributos* envolvidos com a qualidade de produto de *software* (PÁDUA et al., 2003).

Algumas medidas da Metodologia Praxis segundo Pádua et al., (2003) são apresentadas e demonstram a sua classificação de medidas de defeitos, sendo elas:

- Defeitos de Testes de aceitação;
- Defeitos de Avaliação pelo usuário.

A medida “Defeitos de Testes de Aceitação” é coletada na atividade de teste de aceitação do Praxis. Os testes de aceitação têm por objetivo validar o produto, ou seja, verificar se ele atende aos requisitos especificados. O produto deve atender tanto aos requisitos funcionais quanto aos requisitos não-funcionais. A medição em testes de aceitação pode envolver praticamente todas as características de qualidade do produto, dependendo de o que o cliente especificou como requisito a ser atendido. Uma vez que os testes de aceitação são realizados a partir do código executável do produto, a medição de defeitos desses testes é considerada como uma medida de qualidade externa ou qualidade em uso (PÁDUA et al., 2003).

A medida “Defeitos de Avaliação pelo Usuário” é coletada na atividade de avaliação de uso, que formaliza a avaliação do produto por parte dos usuários. De acordo com a lista de conferência utilizada pela atividade de avaliação de uso, ela tem por objetivo verificar os aspectos que influenciam a característica de usabilidade da qualidade do produto (PÁDUA et al., 2003).

As medidas de defeitos da metodologia Praxis selecionadas estão descritas no Anexo 3 de acordo com o padrão de campos para classificação (PÁDUA et al., 2003).

2 OBJETIVO DO TRABALHO

O objetivo deste trabalho é o desenvolvimento de um método de aferição da qualidade de produtos de *software*, por meio da aplicação da metodologia QFD, no desdobramento da qualidade esperada pelo usuário de produtos de *software*. O método visa contribuir na definição, organização, priorização e padronização de medidas de qualidade.

3 REFERENCIAL METODOLÓGICO

3.1 Tipo de pesquisa

Este capítulo trata do *modelo conceitual* para ser aplicado na aferição a qualidade de produtos de *software*. O trabalho trata-se de uma pesquisa aplicada, pois objetiva a aplicação de conhecimentos básicos na geração de novos produtos. Neste contexto, a pesquisa utilizou a metodologia QFD e a norma ABNT (2003) na criação de um método para a aferição da qualidade de produtos nos processos de desenvolvimento de *software*.

O método proposto neste trabalho é de caráter explicativo, pois visa ampliar a utilização da metodologia QFD na área de Engenharia de *Software* e utiliza-se de procedimento experimental.

3.2 Limites do trabalho

Tão importante quanto enumerar os objetivos do trabalho é esclarecer os seus limites, visando delimitar mais precisamente o escopo que está sendo tratado.

Este trabalho trata especificamente da criação de um modelo para garantir a qualidade do produto e não a qualidade do processo de desenvolvimento de *software*, desdobrando a qualidade esperada pelo usuário em medidas de qualidade interna no desenvolvimento de produtos de *software*

Além da qualidade de produto objeto desta pesquisa, vale ressaltar a necessidade em se aferir a qualidade do processo de desenvolvimento de produtos de *software*, e assim, garantir e alcançar outros objetivos como produtividade, lucratividade, à melhoria contínua do processo através da avaliação da maturidade do processo de desenvolvimento.

Não faz parte do escopo a definição da ferramenta para aplicação do modelo desta pesquisa, portanto, o método proposto neste trabalho é de caráter explicativo, pois tenta ampliar a utilização da metodologia QFD na área de desenvolvimento de produtos de *software*. A metodologia QFD já foi utilizada para o levantamento de requisitos dos produtos de *software*, porém pouco utilizada na definição de medidas para acompanhar a qualidade durante o desenvolvimento de produtos de *software*.

Portanto, esta dissertação utiliza-se de procedimento experimental na definição de um método para mensurar a qualidade esperada pelo usuário durante o desenvolvimento de produtos de *software*.

3.3 Modelo conceitual do método de aferição da qualidade de produtos de *software* utilizando a metodologia QFD

Para atingir o objetivo desta pesquisa, o primeiro passo foi conduzir um estudo da literatura especializada, buscando conhecer a metodologia QFD e processos e normas de aferição da qualidade produtos de *software*.

A partir deste estudo foi desenvolvido e proposto um método de aferição da qualidade *software* (*modelo conceitual*) que se aplica a metodologia QFD no mapeamento da qualidade do produto em medidas utilizadas para o desenvolvimento de *software*. O desdobramento da qualidade esperada pelo usuário foi definido através da construção de um *modelo conceitual* para aferição da qualidade de produto de *software*. O *modelo conceitual* proposto utiliza quatro matrizes correlacionadas: Matriz de Qualidade do Usuário, Matriz de Qualidade em Uso, Matriz de Qualidade Externa e Matriz de Qualidade Interna. O encadeamento das matrizes, acima citadas, define o *modelo conceitual* que guia o desdobramento da qualidade esperada pelo cliente permitindo a

definição de medidas para o controle e o acompanhamento dessa qualidade durante o desenvolvimento do *software*.

O método também se utiliza do modelo de qualidade definido na norma ABNT (2003), para transferir uma padronização à qualidade do produto de *software* em dois modelos: qualidade interna e externa e qualidade em uso, utilizados na construção do *modelo conceitual*. De acordo com a norma ABNT (2003), as qualidades internas e externas possuem as mesmas características de qualidade, porém, a qualidade interna é mensurada através dos resultados intermediários do produto e a qualidade externa é mensurada através da execução do seu código. O *modelo conceitual* é apresentado e detalhado no capítulo 4 e sendo ilustrado pela Figura 13 na seção 4.1.1.

3.3.1 Extração, conversão e correlação das matrizes QFD

A extração acontece quando elementos de uma tabela são obtidos a partir de elementos de outra tabela. A conversão consiste em transmitir a importância dos elementos de uma tabela para outros elementos de outra tabela realizados a posterior ao processo de correlação. A correlação visa identificar as relações entre elementos desdobrados do último nível das tabelas. O grau ou a intensidade da correlação é indicado com Forte, Fraca e Possível. Os valores utilizados neste trabalho para indicar estes critérios são \odot ou A - Forte = 9, \circ ou M - Fraca = 3 e \triangle ou B- Possível = 1.

3.3.2 Nível de Avaliação

O nível de avaliação está relacionado com a importância dada para uma característica da norma ABNT (2003). A importância de cada uma das características influencia a profundidade e completude de sua avaliação. No método proposto neste trabalho, o nível de avaliação é utilizado para ajustar o grau de importância, utilizado para

priorizar as medidas para a Matriz de Qualidade em Uso, da Matriz de Qualidade Externa e da Matriz de Qualidade Interna. O ajuste do grau de importância é feito através dos pesos associados aos níveis de avaliações, sendo que os pesos considerados para o nível A, B, C, e D são respectivamente 4, 3, 2 e 1.

3.3.3 Medidas de qualidade

Foram classificadas algumas medidas da norma ABNT (2003), do padrão IEEE-982 (1988) e da metodologia de desenvolvimento Praxis. A classificação proposta no *modelo conceitual* segue o modelo de qualidade da norma ABNT (2003) e o padrão de campos para descrever suas medidas internas encontram-se descritas no Anexo 8.1.

As medidas selecionadas do padrão IEEE-982 (1988) que estão descritas no Anexo 8.2 são utilizadas como indicador da característica Confiabilidade da qualidade de um produto de *software*.

Da metodologia Praxis foram selecionadas medidas que estão descritas no Anexo 8.3 que tratam fundamentalmente da mensuração do produto.

3.3.4 Instanciação¹ do método para metodologia Praxis

A última etapa na elaboração do *modelo conceitual* foi a sua aplicação a uma metodologia de desenvolvimento de produtos de *software*. A aplicação foi realizada por meio da sua instanciação para a metodologia Praxis.

O processo de instanciação do método de aferição de qualidade a metodologia Praxis é descrito e apresentado detalhadamente no Capítulo 4.3 deste trabalho.

3.4 Aplicação do método

O método foi validado por meio de um experimento de engenharia de *software* em um ambiente real de desenvolvimento em uma concessionária de Motocicletas e Peças de um grande fabricante nacional.

Por questão de sigilo a concessionária e o fabricante de motocicletas não terão seus nomes revelados. Também não foi autorizada pela concessionária a inclusão no escopo deste trabalho a divulgação da metodologia utilizada para os levantamentos de dados e informações. As informações utilizadas neste trabalho para a validação do método foram fornecidas e formatadas pela concessionária. A validação do método dependeu da disponibilidade de recursos e pessoas chaves, envolvidas nos projetos. O método foi aplicado a um produto de *software* desenvolvido internamente nesta concessionária chamado MotorcyclesControl.

O MotorcyclesControl é um produto destinado a gestão e controle de compras, vendas, estoques e financeiro que possui sub-módulos respectivamente.

¹Instanciação: em tecnologia da informação, devido a importação do idioma inglês, significa a concretização de um “molde”, ou seja, dar vida a um processo por meio de uma instância.

3.4.1 Requisitos funcionais

Para preencher a Tabela de Requisitos Funcionais foi utilizada a lista de CUA (Casos de Uso de Análise) do modelo CRSw e MASw. Essa lista é composta por:

- Gestão Manual de Estoque;
- Gestão de Mercadorias;
- Gestão de Fornecedores;
- Gestão de Pedidos de Compra;
- Operação de Venda;
- Relatório de Estoque Baixo;
- Relatório de Mercadorias;
- Relatório de Fornecedores;
- Relação de Pedidos de Compra;
- Emissão de Nota Fiscal.

3.4.2 Atributos de qualidade

No MotorcyclesControl, a Tabela de Características de Qualidade é obtida usando os atributos de qualidade obtidos a partir do modelo MASw e do documento PESw (Proposta de Especificação de *Software*). O documento PESw fornece uma lista dos benefícios esperados pelo usuário, utilizados para fazer a *extração* dos atributos de qualidade. No caso do MotorcyclesControl, os atributos de qualidade foram completados pela lista dos requisitos não-funcionais e as restrições de memória, obtidos a partir do modelo MASw. A junção das informações desses dois artefatos compõe a lista de atributos de qualidade descritos a seguir:

- Economia de mão-de-obra na operação de venda;
- Agilidade na compra e venda de mercadorias;
- Diminuição de erros nas vendas;

- Diminuição do tempo de venda;
- Diminuição dos prejuízos na operação de venda;
- Diminuição dos erros nas notas fiscais;
- Eliminação da duplicidade de pedidos;
- Tempo de operação de venda deverá gastar no máximo 2 segundos;
- Tempo de operação de pesquisa de 10 segundos;
- Desenhado de forma que possa ser expandido para mais de um terminal de caixa;
- Leiaute do relatório Nota Fiscal deve seguir padrão da Secretaria da Receita;
- Um operador de caixa deverá ser capaz de aprender a operar o MotorcyclesControl com 1 dia de treinamento;
- Restringir o acesso dos usuários às funções por meio de senhas, conforme o respectivo grupo;
- O produto deve ocupar no máximo 200 MB em disco (sem considerar as bases de dados);
- O produto deve executar em 128 MB de memória RAM;

3.4.3 Matrizes de qualidade

As matrizes de qualidade de usuário, qualidade em uso, qualidade externa e qualidade interna assim como os gráficos apresentados no estudo de caso possuem o detalhamento de suas elaborações descritas no Capítulo 5 deste trabalho.

4 PROPOSTA DO MÉTODO DE AFERIÇÃO DA QUALIDADE DE PRODUTOS DE SOFTWARE (MODELO CONCEITUAL)

O desenvolvimento do método para aferição da qualidade de produtos de *software* foi baseado no modelo de qualidade da norma ABNT (2003) e na metodologia QFD. Utilizou-se o QFD para o mapeamento da qualidade do produto em medidas utilizadas para o desenvolvimento do *software*. O desdobramento da qualidade esperada pelo usuário foi definido através da construção de um *modelo conceitual* para aferição de qualidade de produtos de *software*. O *modelo conceitual* utiliza quatro matrizes: Matriz de Qualidade do Usuário, Matriz de Qualidade em Uso, Matriz de Qualidade Externa e Matriz de Qualidade Interna. As matrizes são utilizadas para mapear a qualidade esperada pelo usuário em medidas para o controle e o acompanhamento dessa qualidade durante o desenvolvimento do *software*.

O método desenvolvido também se utilizou do modelo de qualidade definida na norma ABNT (2003) para transferir uma padronização à qualidade do produto esperada pelo usuário. A norma ABNT (2003) divide a qualidade dos produtos de *software* em dois modelos: qualidade interna e externa, e qualidade em uso, utilizados na construção do *modelo conceitual*. De acordo com a norma ABNT (2003), as qualidades internas e externas possuem as mesmas características de qualidade, porém a qualidade interna é mensurada através dos resultados intermediários do produto e a qualidade externa é mensurada através da execução do seu código.

A partir da padronização da qualidade definida pela norma ABNT (2003) são selecionadas e classificadas medidas de qualidade sugeridas pelo padrão IEEE-982 (1988), pela metodologia Praxis (Pádua et al., 2003) e pela norma ABNT (2003).

Para que o método proposto ilustrado pela Figura 16 possa ser aplicado a um processo de desenvolvimento concreto foi instanciado à metodologia de desenvolvimento de produtos de software Praxis (Pádua et al., 2003)

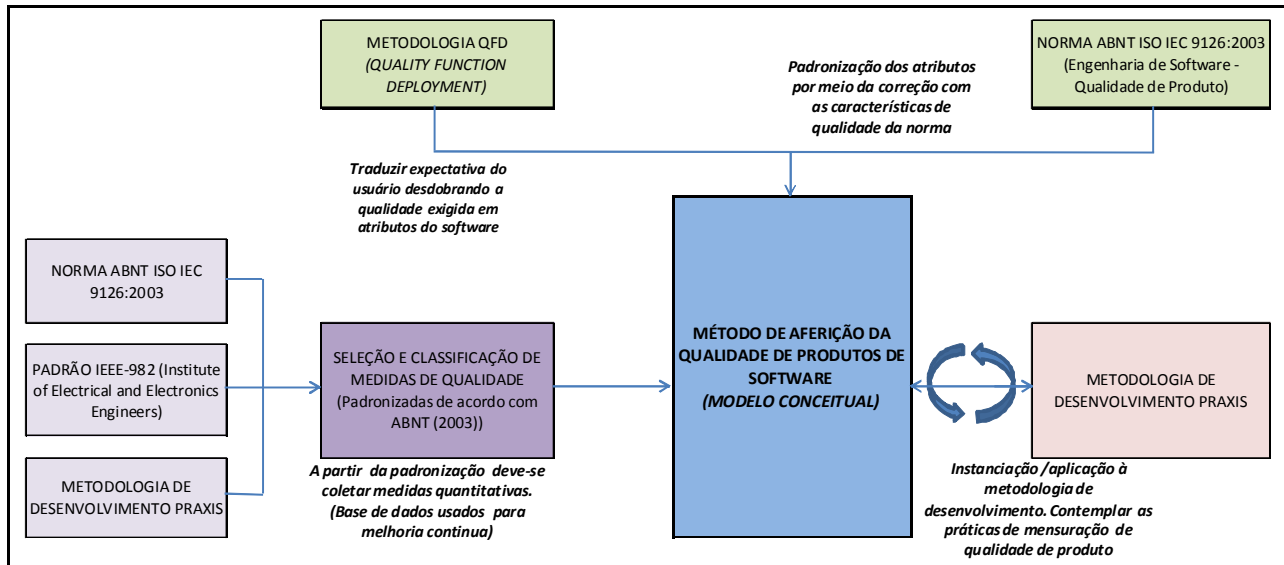


Figura 16 – Método proposto – (Modelo conceitual)

4.1 Importância da metodologia QFD e da norma ABNT (2003) no desenvolvimento do método

O modelo de qualidade definido na norma ABNT (2003) pode ser aplicado para todo tipo de produtos de *software*. O modelo de qualidade proposto pela norma é dividido em duas partes: a primeira parte qualidade interna e qualidade externa e a segunda parte qualidade em uso. A qualidade interna representa a qualidade de produtos intermediários do desenvolvimento de *software* e pode ser um indicativo da qualidade externa do produto. A qualidade externa é a qualidade do produto em execução, por exemplo, o produto sendo executado na fase de testes de aceitação e pode ser um indicativo da qualidade em uso. A qualidade em uso representa a qualidade que o usuário percebe em determinados contextos de uso do produto (ABNT, 2003).

O modelo de qualidade da norma ABNT (2003) foi escolhido para este trabalho por ser um padrão estabelecido, pela qualidade e clareza em sua definição e sua abrangência, pois se trata de um padrão internacional. Contudo, outro modelo, possivelmente algum já definido em um processo de desenvolvimento de *software*, pode ser utilizado, desde que apresente categorias de qualidade bem definidas.

A qualidade interna, coletada durante o desenvolvimento, pode fornecer uma indicação da qualidade externa esperada do produto. Contudo, as medidas de qualidade interna, para serem utilizadas como indicativo da qualidade externa, devem ser relacionadas com a qualidade externa do produto. Esse relacionamento da qualidade externa com a qualidade interna é realizado no método definido neste trabalho através da metodologia QFD (AKAO, 1996).

A metodologia QFD foi escolhida para guiar o desdobramento da qualidade de *software*, por se tratar de um método consolidado, flexível e comprovadamente eficiente para desdobrar a qualidade exigida pelo usuário nos processos de desenvolvimento de produtos.

4.1.1 Desenvolvimento do método para aferição da qualidade em produtos de *software*

O método para aferição da qualidade de produtos de *software* tem o objetivo de prover medidas para mensuração da qualidade durante o desenvolvimento dos produtos de *software*, por meio do desdobramento da qualidade esperada pelo usuário. A mensuração da qualidade durante o desenvolvimento dos produtos de *software* antecipa a descoberta de falhas e quanto mais cedo elas são identificadas, menor o custo de sua correção. Através da correção dessas falhas, espera-se que o desenvolvimento do produto possa obter a qualidade esperada pelo usuário, contribuindo para o custo e o prazo sob controle.

O método tem por objetivo também auxiliar a padronização da qualidade levantada com o usuário de acordo com a norma ABNT (2003). Essa padronização auxilia a criação de uma base histórica de projetos da empresa, visando à reutilização de conhecimento, como por exemplo, medidas, soluções técnicas, arquitetura dentre outros. O reuso de conhecimento é um passo em direção a melhoria contínua da qualidade do produto e também do aumento da produtividade nas atividades do processo de desenvolvimento.

A Figura 17 apresenta o *modelo conceitual*, conforme proposto na metodologia QFD, que utiliza quatro matrizes correlacionadas, sendo elas, respectivamente, Matriz de Qualidade do Usuário, Matriz de Qualidade em Uso, Matriz de Qualidade Externa e Matriz de Qualidade Interna. O modelo é utilizado para desdobrar a qualidade esperada pelo usuário em medidas e também para conferir uma padronização aos atributos de qualidade de acordo com a norma ABNT (2003).

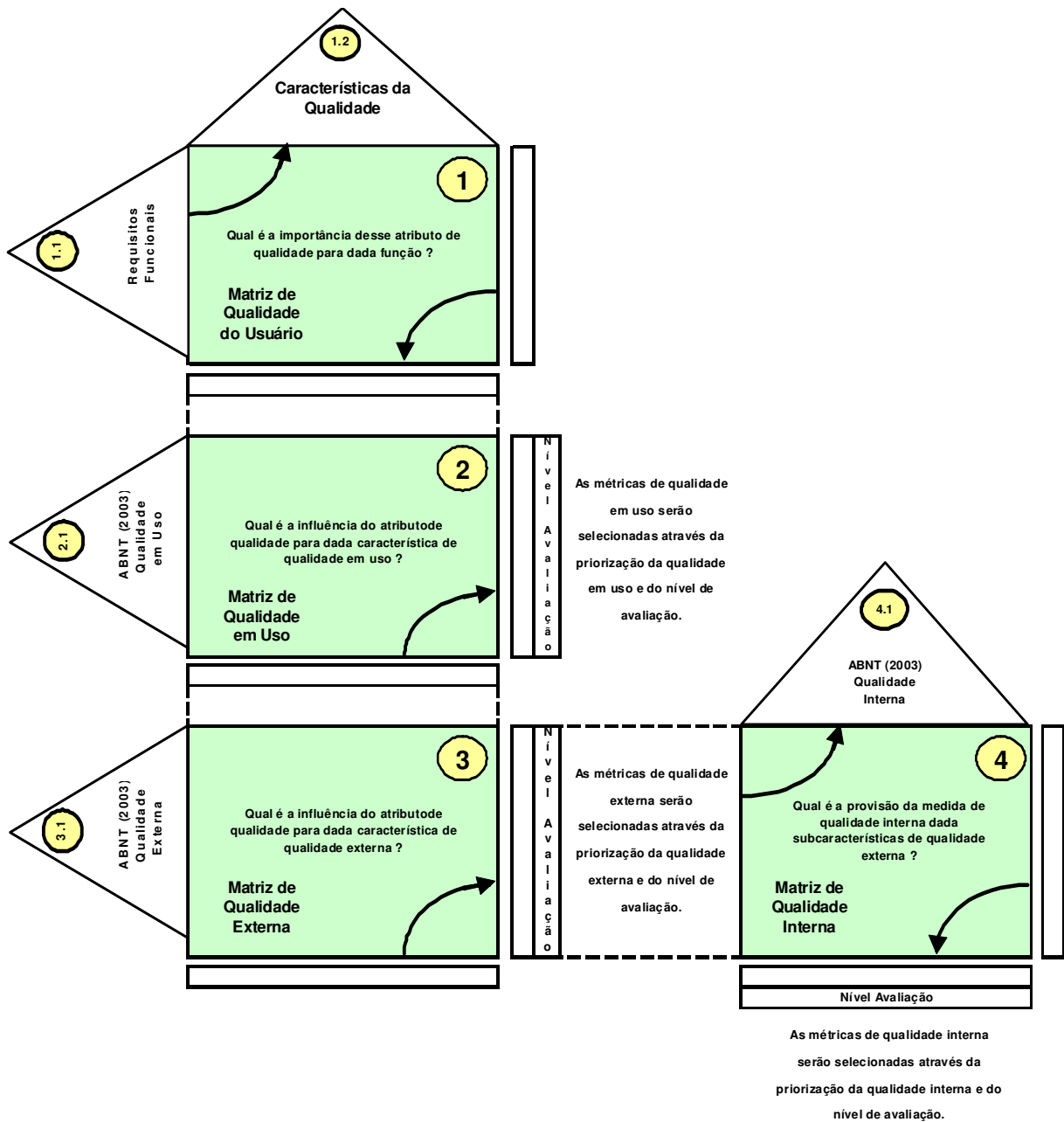


Figura 17 - Modelo conceitual para aferição de qualidade para produtos de software

Fonte: Adaptada de Cheng et al. (1995)

A Matriz de Qualidade do Usuário (Matriz 1 – Figura 17) correlaciona as funções do produto com os atributos de qualidade esperados pelo usuário. Os atributos de qualidade são também conhecidos como requisitos não-funcionais na área de Engenharia de *Software*. O objetivo da matriz de qualidade do usuário é determinar, desde o início do projeto, os atributos de qualidade mais importantes para o usuário, o que é feito por meio de sua correlação com as funções do produto. A Matriz de Qualidade em Uso (Matriz 2 – Figura 17) correlaciona os atributos de qualidade esperados pelo usuário com as características de qualidade em uso da norma ABNT (2003). O objetivo dessa atividade é priorizar as características de qualidade em uso e transferir a padronização de qualidade em uso do produto de *software* aos atributos de qualidade, do ponto de vista do usuário. A Matriz de Qualidade Externa (Matriz 3 – Figura 17) correlaciona os atributos de qualidade esperados pelo usuário com as características de qualidade externa da norma ABNT (2003), visando à transferência da padronização da qualidade externa do produto de *software* aos atributos de qualidade levantados com o usuário. A Matriz de Qualidade Interna (Matriz 4 – Figura 17) correlaciona as características de qualidade externa da ABNT (2003) com as medidas de qualidade interna do desenvolvimento do produto. O objetivo dessa matriz é priorizar as medidas internas que possam auxiliar na previsão da qualidade externa do produto.

O encadeamento das matrizes, acima citadas, define o *modelo conceitual* de aferição de qualidade proposto neste trabalho. Esse *modelo conceitual* irá guiar o desdobramento da qualidade esperada pelo usuário auxiliando a definição de medidas para o desenvolvimento do produto de *software*. Além da definição de medidas, o desdobramento da qualidade pode auxiliar no planejamento das atividades de construção do produto de *software*.

A definição das medidas para cada característica de qualidade, definida na norma ABNT (2003), é auxiliada pelo seu Nível de Avaliação. Cada característica de qualidade é

priorizada de acordo com a qualidade esperada pelo usuário e pode ser mensurada através da coleta de medidas durante o desenvolvimento do produto. O Nível de Avaliação foi combinado com o *modelo conceitual* proposto para permitir uma melhor priorização das medidas da qualidade.

As próximas seções apresentam o conceito de Nível de Avaliação utilizado para seleção de medidas no método, o processo de construção das matrizes e o relacionamento entre essas matrizes, visando à criação do *modelo conceitual* de aferição de qualidade.

4.1.2 Nível de Avaliação

O método proposto utiliza o Nível de Avaliação em três matrizes: Qualidade em Uso, Qualidade Externa e Qualidade Interna, com o objetivo de auxiliar a priorização das medidas para o processo de desenvolvimento. A seleção das medidas nessas matrizes considera o grau de importância, calculado através do *modelo conceitual*, e o Nível de Avaliação de cada característica de qualidade da norma ABNT (2003).

O Nível de Avaliação está relacionado com a importância dada para uma característica de qualidade da norma ABNT (2003). A importância de cada característica influencia a profundidade e completude da sua avaliação. As técnicas de avaliação são utilizadas para coletar as medidas relacionadas com as características de qualidade do modelo da norma ABNT (2003).

O método propõe quatro níveis de avaliação, designados: A, B, C e D. Esses níveis constituem uma hierarquia, sendo A o nível mais alto e D o mais baixo. No nível A, as técnicas de avaliação mais estritas (levando em consideração uma quantidade razoável de esforço e tempo) são aplicadas dando uma maior confiança nas medidas

coletadas. Até o nível D, gradualmente, técnicas menos estritas são utilizadas, e conseqüentemente, menos esforço é dedicado para a avaliação das medidas.

O Nível de Avaliação é selecionado conforme a conseqüência de um erro no *software* para os aspectos pertinentes ao seu uso

A seleção do Nível de Avaliação deve considerar o nível mais estrito de cada aspecto quando são necessários vários aspectos para o *software*. As técnicas de avaliação devem ser escolhidas de acordo com a característica de qualidade da norma ABNT (2003) e o Nível de Avaliação.

Com as técnicas de avaliação é possível avaliar a qualidade do produto, para cada característica de qualidade, através da coleta de medidas. A qualidade do produto, de acordo com a norma ABNT (2003), é categorizada em características de qualidade bem distintas, que podem ser avaliadas de forma independente.

No método proposto neste trabalho, o Nível de Avaliação é utilizado para ajustar o grau de importância, utilizado para priorizar as medidas para a Matriz de Qualidade em Uso, da Matriz de Qualidade Externa e da Matriz de Qualidade Interna. O ajuste do grau de importância é feito através dos pesos associados aos Níveis de Avaliações, sendo que os pesos considerados para o nível A, B, C e D são, respectivamente, 4, 3, 2 e 1. Os pesos dos níveis de avaliação são valores sugeridos, que podem ser alterados de acordo com a percepção dos envolvidos. Dessa forma, as características de qualidade que, estão associadas ao nível A tem seu grau de importância aumentado.

A unificação do Nível de Avaliação e o grau de importância das características, obtida nas matrizes, são apresentados a seguir.

4.1.3 Matriz da Qualidade do Usuário

A Matriz de Qualidade do Usuário, mostrada na Figura 18, relaciona as funções do produto e os atributos de qualidade esperados pelo cliente. As funções do produto são levantadas juntamente com o usuário e registradas na Tabela de Requisitos Funcionais, (Ilustração 1.1 - Figura 18). Os atributos de qualidade esperados pelo cliente são extraídos também do usuário e registrados na Matriz de Características de Qualidade, (Ilustração 1.2 - Figura 18). As duas tabelas são relacionadas, para a construção dessa matriz, com o objetivo de priorizar os atributos de qualidade de acordo com o entendimento do usuário do produto.

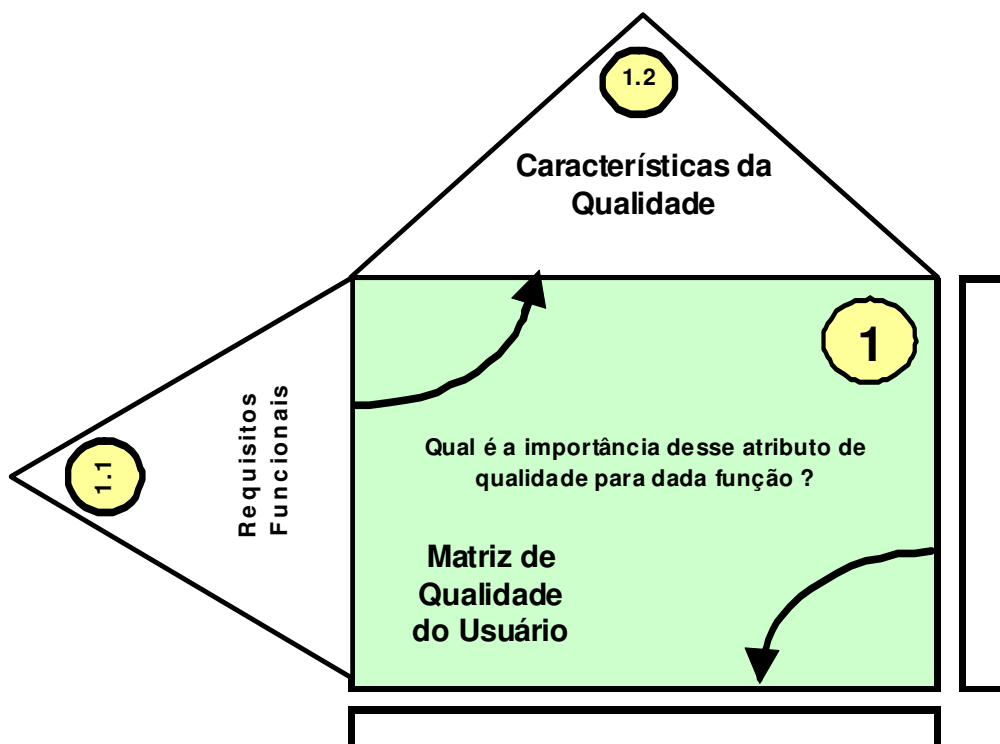


Figura 18 - Matriz da Qualidade do Usuário

A Tabela de Requisitos Funcionais é formada por uma lista priorizada das funções do produto de *software*. Com as funções identificadas é preciso priorizá-las conforme a importância que essas representam para o usuário em relação ao produto.

O usuário é responsável por priorizar as funções críticas para o produto de *software*. Essa priorização deve seguir critérios, como as funções mais utilizadas ou àquelas que representam uma atividade importante para o usuário. Cada função deve receber um valor numérico que representa o seu grau de importância em relação às outras funções.

A Tabela de Características de Qualidade é formada pelos atributos de qualidade que o usuário espera do produto de *software*. Esses atributos são obtidos usando um processo de extração, das expectativas do usuário a partir das funções listadas na tabela de requisitos funcionais. As expectativas do usuário expressas nas Características de Qualidade devem buscar contemplar suas necessidades intrínsecas, observar seus valores, visando maximizar a satisfação do usuário. O termo “usuário” neste contexto representa não somente os usuários diretos do produto de *software*, mas também todos os interessados no produto incluindo representantes da organização cliente e acionistas.

Depois da construção da Tabela de Requisitos Funcionais e da Tabela de Características de Qualidade, o relacionamento dessas permitirá priorizar os atributos de qualidade esperados pelo usuário. O cliente do produto é consultado para realizar a correlação dessas tabelas da Matriz de Qualidade do Usuário através da pergunta “Qual é a importância desse atributo de qualidade para dada função?”.

A correlação da Tabela de Requisitos Funcionais com a Tabela de Características de Qualidade na Matriz de Qualidade do Usuário pode trazer benefícios ao desenho da solução do produto. O primeiro benefício é que o relacionamento possibilita desenhar as funções do produto conforme os atributos de qualidade prioritários. Cada função é correlacionada com os atributos de qualidade e, dessa forma, o arquiteto do *software*

pode desenhar uma solução mais adequada às expectativas de qualidade do usuário. O segundo benefício é priorizar os atributos de qualidade que mais preocupam os usuários, e assim, os programadores podem desenvolver o produto focando nos atributos prioritários.

Com a *conversão* realizada na Matriz de Qualidade do Usuário os atributos de qualidade esperados pelo cliente são priorizados, sendo que eles serão utilizados para construção da Matriz de Qualidade em uso e da Matriz de Qualidade Externa, as próximas matrizes do *modelo conceitual*.

4.1.4 Matriz da Qualidade em Uso

A Matriz de Qualidade em Uso do *modelo conceitual*, exibida na Figura 19, tem por objetivo conferir padronização aos atributos de qualidade esperados pelo usuário de acordo com o modelo de qualidade em uso definido na norma ABNT (2003). Os atributos de qualidade utilizados na Matriz de Qualidade em Uso são provenientes da matriz de Características de Qualidade (Ilustração 1.2 - Figura 19), da Matriz de Qualidade do Usuário. O modelo de qualidade em uso define um conjunto de características de qualidade, sendo utilizado para formar a matriz de Qualidade em Uso (Ilustração 2.1 - Figura 19). As informações dessas tabelas são correlacionadas pelo engenheiro de *software* para construir a Matriz de Qualidade em Uso do *modelo conceitual*.

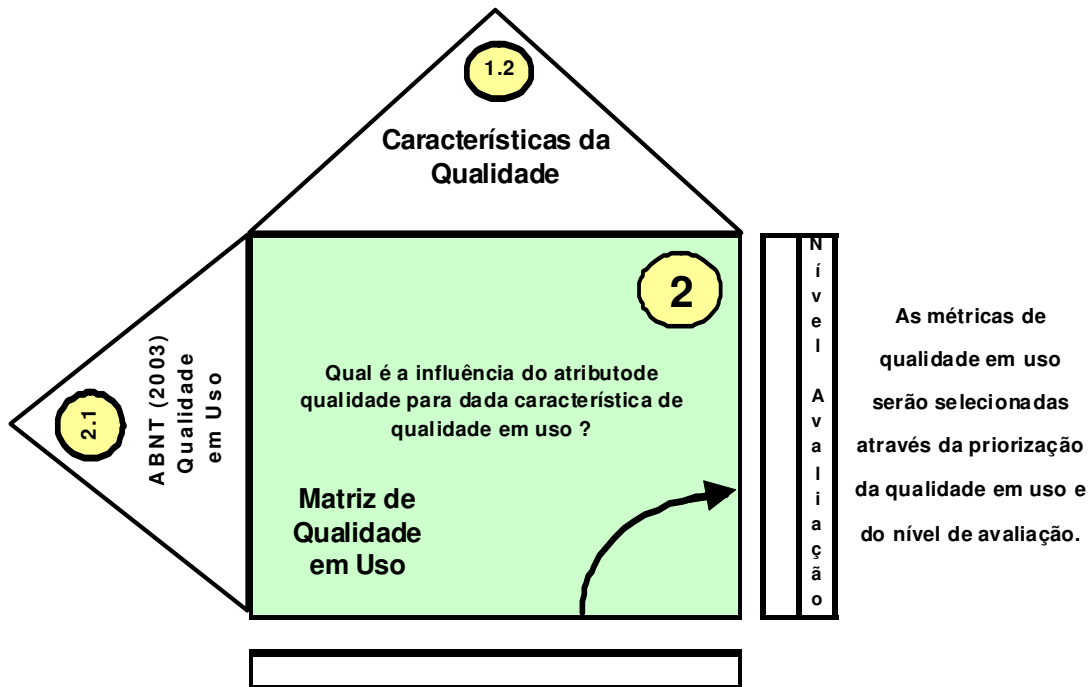


Figura 19 - Matriz da Qualidade em Uso

A Tabela de Características de Qualidade contém os atributos de qualidade levantados e priorizados na construção da Matriz de Qualidade do Usuário, ilustrada na Figura 18. Esses atributos foram priorizados através do relacionamento com as funções do produto e receberam seus graus de importância. Essa mesma priorização é utilizada para calcular o grau de importância da Tabela de Qualidade em Uso.

A utilização da Matriz de Qualidade em Uso nesta parte do modelo possibilita avaliar a capacidade do produto em alcançar os objetivos dos usuários em determinados contextos de uso. O modelo definido na norma categoriza a qualidade em uso, em quatro características: eficácia, produtividade, segurança e satisfação. Essas características são priorizadas através do relacionamento com os atributos de qualidade da Tabela de Características de Qualidade.

As informações da Tabela de Características de Qualidade e da Tabela de Qualidade em Uso são correlacionadas para construção da Matriz de Qualidade em Uso.

A *correlação* dessas tabelas é realizada pelo engenheiro de *software* nessa matriz através da pergunta “Qual é a influência do atributo de qualidade para dada característica de qualidade em uso?”.

Depois do relacionamento das características de qualidade em uso e os atributos de qualidade na Matriz de Qualidade em Uso, o grau de importância de cada característica de qualidade em uso é calculado através da *conversão*. Cabe observar que essa *conversão* nessa matriz é realizada no sentido horizontal, diferente da primeira matriz do *modelo conceitual*.

O relacionamento realizado na Matriz de Qualidade em Uso auxilia na priorização das medidas para as características de qualidade definidas na norma ABNT (2003). De acordo com o grau de importância e o Nível de Avaliação das características de qualidade em uso, o engenheiro de *software* pode selecionar as medidas de qualidade adequadas para a qualidade esperada pelo usuário.

Essa seleção das medidas deve considerar quais características prioritárias para usuário, porém não deve deixar de selecionar medidas para outras características. Com essa seleção das medidas, realizada na Matriz de Qualidade em Uso, o engenheiro de *software* consegue mensurar a qualidade em determinados contextos de uso do produto.

Além da priorização das medidas de qualidade em uso, a *correlação* realizada na Matriz de Qualidade em Uso proporciona padronização aos atributos de qualidade esperados pelo usuário de acordo com a norma ABNT (2003). Os atributos de qualidade são levantados com o usuário na sua linguagem própria, dificultando comparações com os atributos de qualidade de outros produtos desenvolvidos. A padronização da qualidade de acordo com a norma possibilita a empresa construir uma base histórica de projetos. A construção dessa base histórica possibilita aproveitar e até reutilizar alguns conhecimentos, como medidas, listas de conferência, soluções de desenho dentre outros.

Essa base histórica também pode ser utilizada para comparar a qualidade de um produto com outros. A comparação entre projetos permite a empresa alcançar uma melhoria contínua na qualidade dos seus produtos. A melhoria dos produtos mantém os clientes satisfeitos e possibilita a captação de novos clientes.

A próxima matriz do *modelo conceitual*, a Matriz da Qualidade Externa, exibida na Figura 20, é semelhante à Matriz de Qualidade em Uso, porém utiliza as características de qualidade externa, ao invés, das características de qualidade em uso. Nessa próxima matriz, a qualidade esperada pelo usuário é padronizada de acordo com o modelo de qualidade externa definido na norma ABNT (2003).

4.1.5 Matriz da Qualidade Externa

A Matriz de Qualidade Externa do *modelo conceitual*, exibida na Figura 20, relaciona os atributos de qualidade esperados pelo usuário às características de qualidade definidas no modelo de qualidade externa da norma ABNT (2003). Os atributos de qualidade utilizados na Matriz de Qualidade Externa são provenientes da Tabela de Características de Qualidade (Ilustração 1.2 – Figura 20), da Matriz de Qualidade do Usuário. As características de qualidade externa são definidas no modelo de qualidade externa da norma ABNT (2003) e formam a Matriz de Qualidade Externa (Ilustração 3.1 - Figura 20). As informações dessas tabelas são relacionadas pelo engenheiro de *software* para transferir padronização à qualidade esperada pelo cliente de acordo com o modelo de qualidade externa da norma ABNT (2003).

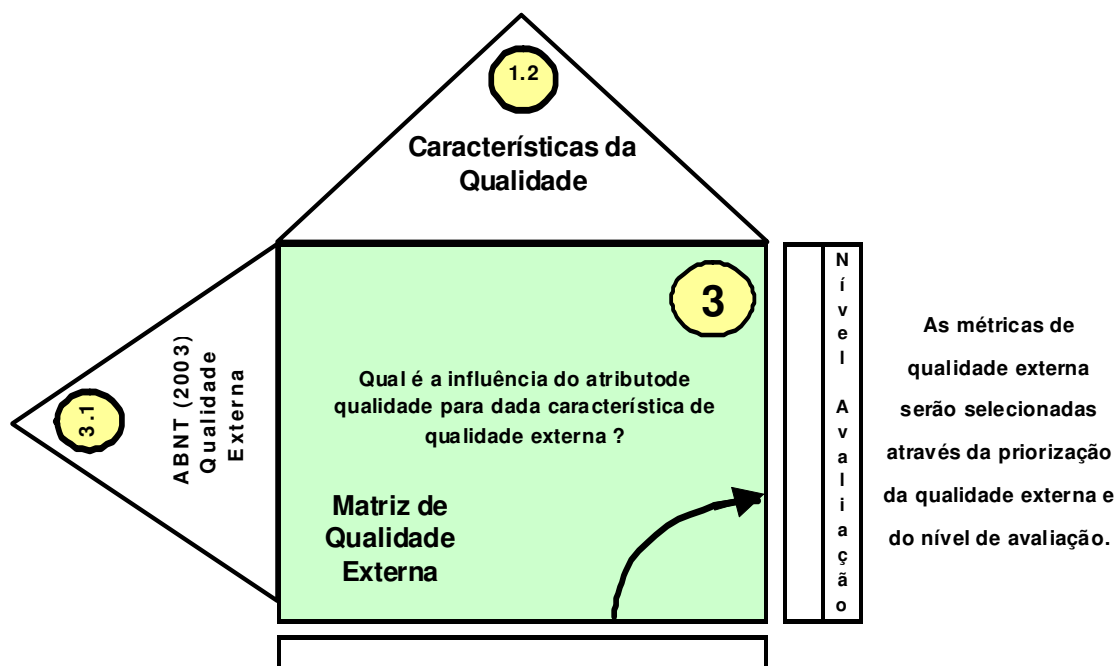


Figura 20 - Matriz da Qualidade Externa

A Tabela de Características de Qualidade contém os atributos de qualidade levantados e priorizados na construção da Matriz de Qualidade do Usuário, ilustrada na Figura 21. Esses atributos foram priorizados através do relacionamento desses com as funções do produto e receberam seus graus de importância. A priorização dos atributos de qualidade é utilizada para calcular o grau de importância da Tabela de Qualidade Externa.

A Tabela de Qualidade Externa é formada pelas características definidas no modelo de qualidade externa da norma ABNT (2003). O modelo de qualidade externa possibilita avaliar a capacidade de um produto de *software* alcançar a qualidade esperada da execução do produto. Esse modelo de qualidade externa é formado por seis características: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Sendo que cada característica é subdividida em subcaracterísticas que padronizam a qualidade do produto e podem ser mensuradas por medidas externas.

As informações da Tabela de Características de Qualidade e da Tabela de Qualidade Externa são correlacionadas para construção da Matriz de Qualidade Externa. A *correlação* dessas tabelas é realizada pelo engenheiro de *software* nessa matriz através da pergunta “Qual é a influência do atributo de qualidade para dada característica da qualidade externa?”.

Depois do relacionamento das subcaracterísticas de qualidade externa com os atributos de qualidade, o grau de importância de cada subcaracterística de qualidade externa é calculado através da *conversão*.

O relacionamento realizado na Matriz de Qualidade Externa auxilia na priorização das medidas para as subcaracterísticas de qualidade externa da norma ABNT (2003). Essa priorização é proveniente do relacionamento realizado nessa matriz e do grau de importância dos atributos de qualidade esperados pelo usuário. De acordo com as subcaracterísticas de qualidade externa priorizadas e os seus níveis de avaliação, o

engenheiro de *software* pode selecionar as medidas de qualidade externa adequadas para o usuário.

As medidas de qualidade externa são utilizadas para controlar e acompanhar a qualidade do produto do código executável do *software*. Essas medidas podem ser coletadas em várias fases do desenvolvimento do projeto, principalmente na fase de testes, para verificar a qualidade externa antes de o produto ser entregue ao usuário. Essa verificação da qualidade permite o gerente do projeto antecipar a descoberta de falhas do produto.

Além da priorização das medidas de qualidade externa, o relacionamento realizado na matriz confere padronização aos atributos de qualidade esperados pelo cliente de acordo com a qualidade externa da norma ABNT (2003). Essa padronização da qualidade esperada pelo usuário de acordo com a norma possibilita a empresa construir uma base histórica dos projetos realizados. A base histórica da Matriz de Qualidade Externa é unida com a base histórica da Matriz de Qualidade em Uso para completar a padronização da qualidade esperada pelo usuário.

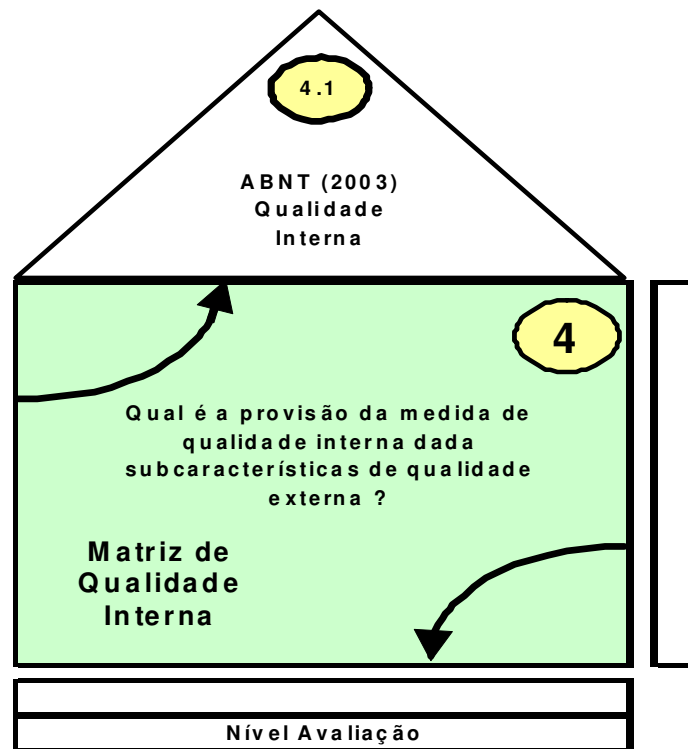
A priorização das subcaracterísticas de qualidade externa, realizada através *conversão* realizada na Matriz de Qualidade Externa, será utilizada na construção da Matriz de Qualidade Interna, a próxima matriz do *modelo conceitual*.

4.1.6 Matriz da Qualidade Interna

A Matriz de Qualidade Interna do *modelo conceitual*, exibido na Figura 21, relaciona as subcaracterísticas de qualidade externa com as medidas de qualidade interna do produto de *software*. As subcaracterísticas de qualidade externa são provenientes da Tabela de Qualidade Externa (Tabela 3.1 – Figura 21), da Matriz de Qualidade Externa. As medidas de qualidade interna do desenvolvimento de *software* da

empresa devem ser listadas na Tabela de Qualidade Interna (Tabela 4.1 – Figura 21). Essas tabelas são correlacionadas pelo engenheiro de *software* com objetivo de verificar quais medidas, e em que grau, de qualidade interna auxilia na previsão da qualidade externa do produto.

A Matriz de Qualidade Interna do *modelo conceitual*, exibido na Figura 21, relaciona as subcaracterísticas de qualidade externa com as medidas de qualidade interna do produto de *software*. As subcaracterísticas de qualidade externa são provenientes da Tabela de Qualidade Externa (Ilustração 3.1 – Figura 20), da Matriz de Qualidade Externa. As medidas de qualidade interna do desenvolvimento de *software* da empresa devem ser listadas na Tabela de Qualidade Interna (Ilustração 4.1 – Figura 21). Essas tabelas são correlacionadas pelo engenheiro de *software* com objetivo de verificar quais medidas, e em que grau, de qualidade interna auxilia na previsão da qualidade externa do produto.



As métricas de qualidade interna
serão selecionadas através da
priorização da qualidade interna e do
nível de avaliação.

Figura 21 - Matriz da Qualidade Interna

A Tabela de Qualidade Externa contém as subcaracterísticas de qualidade externa priorizadas, proveniente da Matriz de Qualidade Externa da Figura 20. As subcaracterísticas de qualidade externa, definidas no modelo da norma ABNT (2003), foram priorizadas através da *correlação* com os atributos de qualidade esperados pelo usuário. Essas subcaracterísticas receberam seus respectivos graus de importância e são utilizados para calcular o grau de importância das medidas de qualidade interna do produto de *software*.

A Tabela de Qualidade Interna é formada pelas medidas de qualidade interna do produto de *software* da empresa. Essas medidas são coletadas com objetivo de acompanhar a qualidade durante a construção desse produto. Essa coleta das medidas tem por objetivo auxiliar a previsão da qualidade externa esperada pelo usuário. Para alcançar esse objetivo, essas medidas internas são correlacionadas com as subcaracterísticas de qualidade externa da Matriz de Qualidade Externa, ilustrada na Figura 20.

As informações da Tabela de Qualidade Externa e da Tabela de Qualidade Interna são correlacionadas para construção da Matriz de Qualidade Interna. A *correlação* dessas tabelas é realizada pelo engenheiro de *software* nessa através da pergunta “Qual é a previsão da medida de qualidade interna dada subcaracterística de qualidade externa?”.

Depois do relacionamento das subcaracterísticas de qualidade externa com as medidas de qualidade interna, o grau de importância de cada medida de qualidade interna é calculado através da *conversão*. Essa *conversão* é realizada no sentido vertical, semelhante à Matriz de Qualidade do Usuário, para priorizar as medidas conforme a qualidade esperado pelo usuário do produto.

O relacionamento realizado na Matriz de Qualidade Interna possibilita priorizar as medidas para acompanhar as atividades do processo de desenvolvimento do *software*. Essa priorização é proveniente da *correlação* das medidas interna e dos graus de importância das subcaracterísticas da qualidade externa. Com as medidas priorizadas e o Nível de Avaliação dessas, o engenheiro de *software* pode mensurar as atividades do processo que influenciam a qualidade externa esperada pelo usuário. Essas medidas de qualidade interna podem ser coletadas em várias fases do desenvolvimento do projeto, desde o início da sua construção. A coleta dessas medidas auxilia a previsão da qualidade externa do produto final desenvolvido.

Além da seleção das medidas para acompanhar a qualidade interna, a priorização dessas medidas possibilita o gerente dimensionar melhor o esforço nas atividades de qualidade do projeto. Essas atividades podem ser dimensionadas focando alcançar as medidas de qualidade interna do produto. Dessa forma, o esforço das atividades do projeto estará priorizado de acordo com a qualidade interna e conseqüentemente com a qualidade final do produto esperada pelo usuário.

A seleção e a classificação das medidas de qualidade interna para mensuração da qualidade de produto serão detalhadas no capítulo seguinte. Essa classificação deve ser realizada com intuito de possibilitar a reutilização das medidas de projetos anteriores da mesma organização.

4.2 Seleção e classificação das medidas de qualidade do *Modelo conceitual*

No método de aferição de qualidade proposto neste trabalho, é utilizada uma adaptação do *Framework de Atributos de Fenton* (1996), descrito na seção 1.4 para padronizar a qualidade esperada pelo usuário. Essa classificação permite organizar as medidas de qualidade do produto utilizadas em projeto anteriores. O uso dessa classificação das medidas possibilita uma uniformização na aferição da qualidade. Por meio dessa uniformização, a empresa pode comparar a qualidade de um projeto com outros, e assim, ter informações sobre a evolução do seu processo de qualidade.

A classificação aqui proposta foi construída através da unificação da árvore de atributos de produtos descrita por Fenton (1996), com o modelo de qualidade da norma ABNT (2003). A classificação definida por Fenton (1996) prevê uma subclasse de qualidade associada à classe de produto. A subclasse de qualidade foi dividida em qualidade interna e externa e qualidade em uso de acordo com o modelo da norma ABNT (2003), conforme mostrado na Figura 22. Esta divisão é interessante no escopo

deste trabalho, pois classifica as medidas para mensurar a qualidade do produto de *software*, selecionadas nas matrizes do método de aferição da qualidade.

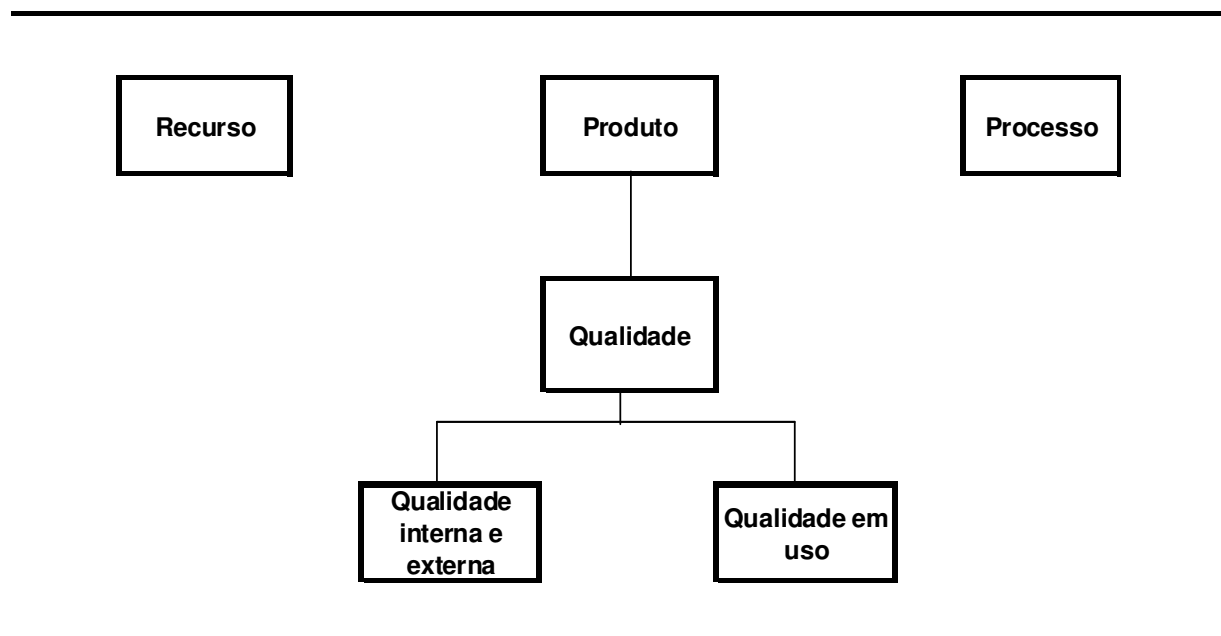


Figura 22 - Classificação de atributos de qualidade adaptada (*Modelo conceitual*)

Além do modelo de qualidade, o padrão dos parâmetros das tabelas de medidas da norma ABNT (2003) foi utilizado para descrever as medidas classificadas.

Foram classificadas algumas medidas da norma ABNT (2003) (Anexo 1), do padrão IEEE-982 (1988) (Anexo 2) e da metodologia Praxis (Anexo 3) de acordo com a classificação proposta. Essas medidas foram escolhidas para demonstrar como classificar e descrever uma medida de qualidade a ser utilizada na metodologia usada neste trabalho. Como existe um número muito grande de medidas na literatura, essa classificação não tem o objetivo de ser extensiva e completa. Além disso, pode-se utilizar medidas próprias que não estariam classificadas neste trabalho.

O padrão IEEE-982 (1988) também foi utilizado para exemplificar a classificação de diferentes medidas de acordo com a classificação adaptada de Fenton (1996). O padrão da *Institute of Electrical and Electronic Engineers* tem o objetivo de fornecer um

conjunto de medidas utilizadas como indicador da característica Confiabilidade da qualidade de um *software*.

Algumas medidas da metodologia Praxis foram escolhidas para exemplificar a classificação de medidas de acordo com o *Framework de Atributos*. Como este trabalho se preocupa fundamentalmente com a mensuração do produto, as medidas escolhidas do Praxis foram àquelas relacionadas ao produto. Essas medidas são utilizadas na aplicação do método proposto.

4.3 Instanciação do método de aferição de qualidade para metodologia Praxis

O método de aferição de qualidade proposta pode ser aplicado a uma metodologia de desenvolvimento concreta, isso é feito através da sua instanciação para a metodologia Praxis.

Dentre os fatores principais que motivaram a instanciação do método proposto para a metodologia Praxis está o fornecimento de subsídios necessário para a elaboração de uma personalização da metodologia Praxis que contemple as práticas de mensuração de qualidade de produto, já que o processo atualmente não possui uma política de medição bem definida. Demonstrar como as matrizes do método de aferição da qualidade de produtos de *software* proposto e descrito na seção 4.1 deste trabalho, podem ser definidas para adequá-las ao contexto de um processo real.

A instanciação do método proposto de aferição da qualidade de produtos de *software* através das matrizes de qualidade à metodologia Praxis é descrita nas seções a seguir.`

4.3.1 Instanciação da Matriz de Qualidade do Usuário ao Praxis

A instanciação da Matriz de Qualidade do Usuário é relacionada no Praxis com os artefatos e as atividades da fase de Elaboração. Esta relação existe, pois esta fase tem como objetivo entender o problema do cliente e criar artefatos que expressem as necessidades do cliente. A Matriz de Qualidade de Usuário tem, por sua vez, exatamente o objetivo de auxiliar a padronização da qualidade do produto esperada pelo usuário. Os artefatos e as atividades envolvidos no levantamento da qualidade do produto no processo Praxis apresentados na seção 1.6, são: CRSw (Cadastro de Requisitos do *Software*) e MASw (Modelo de Análise do *Software*).

O CRSw é o artefato responsável por registrar os casos de uso identificados juntamente com o usuário do produto. Esses casos de uso são utilizados para formar a lista de funções na Tabela de Requisitos Funcionais, da Matriz de Qualidade do Usuário. Essa tabela irá auxiliar na *extração* dos atributos de qualidade da Tabela de Características de Qualidade.

Os atributos de qualidade da Tabela de Características de Qualidade são obtidos a partir da *extração* das funções em atributos de qualidade e a partir dos requisitos não-funcionais registrados no MASw e no CRSw. No Praxis, os requisitos não-funcionais são levantados com os usuários durante o fluxo de Análise no sub-fluxo Oficina de detalhamento dos requisitos. Após a identificação dos atributos de qualidade é possível fazer a *correlação* na Matriz de Qualidade do Usuário.

A *correlação* da Tabela de Requisitos Funcionais com a Tabela de Características de Qualidade tem por objetivo priorizar os atributos de qualidade. Essa priorização dos atributos deve ser realizada na Iteração de Análise de Requisitos do Praxis com a participação do usuário. O fluxo de análise visa detalhar, estruturar e validar os requisitos de um produto, em termos de um *modelo conceitual* do problema. Desta forma os

requisitos podem ser usados como base para o planejamento e controle detalhado do respectivo projeto de desenvolvimento. Nessa iteração a lista de funções do produto está completa e o usuário pode identificar os atributos de qualidade desejados. Sendo assim, a identificação e a priorização dos atributos de qualidade se encaixam como atividades do fluxo de Análise e devem ser registradas utilizando o artefato MASw.

Os atributos de qualidade da Matriz de Qualidade do Usuário servem de entrada para o planejamento e o controle das atividades da fase de Construção e Transição do Praxis. A identificação e a documentação dos atributos de qualidade melhoram o entendimento do analista sobre a qualidade, e dessa forma, possibilitam a construção um produto mais adequado ao usuário. Além disso, a documentação dos atributos pode ser utilizada no planejamento e controle mais detalhado das atividades de garantia da qualidade.

Os atributos de qualidade, registrados no MASw, serão utilizados para construir a próxima matriz do método de aferição da qualidade, que é a Matriz de Qualidade em Uso.

4.3.2 Instanciação da Matriz de Qualidade em Uso ao Praxis

A Matriz de Qualidade em Uso tem o objetivo de conferir a padronização, utilizando a norma ABNT (2003) à qualidade esperada pelo usuário. A qualidade esperada pelo usuário foi priorizada pela *conversão* da Matriz de Qualidade de Usuário.

A Matriz de Qualidade em Uso é formada pela *correlação* da Tabela de Características de Qualidade com a Tabela de Qualidade em Uso. A Tabela de Características de Qualidade é proveniente da Matriz de Qualidade do Usuário. A Tabela de Qualidade em Uso é formada pelas características de qualidade em uso definidas na norma ABNT (2003). Essa duas tabelas são relacionadas com objetivo de identificar quais atributos de qualidade têm influência na qualidade em uso do produto.

A Tabela de Características de Qualidade contém os atributos de qualidade priorizados através da *conversão* na Matriz de Qualidade de Usuário. Essa priorização é utilizada para fazer o cálculo da *conversão* das características de qualidade em uso.

As características de qualidade em uso, definidas na norma ABNT (2003), são usadas para construir a Tabela de Qualidade em Uso, sendo elas: efetividade, produtividade, segurança e satisfação.

Com as duas tabelas construídas é possível correlacionar seus itens na Matriz de Qualidade em Uso. Essa correlação deve seguir os passos do método de aferição de qualidade descritos na seção 4.1.4. Para executar esses passos foi inserida uma atividade no fluxo de Desenho do Praxis. A atividade deve ser realizada em uma reunião com a participação de, pelo menos, um responsável de cada fluxo do Praxis, permitindo a interação de várias visões na *correlação* da matriz. Nessa reunião, os atributos de qualidade e as características de qualidade em uso são correlacionados utilizando a experiência dos participantes e a definição da norma ABNT (2003).

Depois da *correlação*, realizada na Matriz de Qualidade em Uso, deve ser obtido o grau de importância das características de qualidade em uso conforme descrito na seção 4.1.4. As características de qualidade em uso priorizadas devem ser armazenadas em uma nova seção intitulada “Qualidade em Uso” do documento PQSw (Plano de Qualidade do *Software*) do processo Praxis apresentados na seção 1.6 deste trabalho. As informações contidas na seção 4.1.4 podem ser utilizadas como insumo para as atividades da fase Construção e Transição do Praxis. As informações podem, por exemplo, auxiliar no planejamento dos casos de testes, que é uma das primeiras atividades do fluxo de Teste da fase de Construção.

Com o grau de importância das características de qualidade em uso e o Nível de Avaliação descrito na seção 4.1.2, é possível priorizar as medidas de qualidade em uso mais adequadas para mensurar a qualidade esperada pelo usuário. A seleção das

medidas deve ser realizada no sub-fluxo Planejamento do fluxo de Gestão de Projeto e devem ser registradas no artefato MPPSw (Memória de Planejamento de Projeto de *Software*) apresentados na seção 1.6 deste trabalho. Dessa forma, a qualidade em uso esperada pelo usuário pode ser acompanhada durante a execução do projeto de desenvolvimento.

4.3.3 Instanciação da Matriz de Qualidade Externa ao Praxis

A Matriz de Qualidade Externa tem por objetivo proporcionar padronização à qualidade esperada pelo usuário de acordo com a norma ABNT (2003) como referência. A qualidade esperada pelo usuário está representada com uma lista de atributos priorizada, proveniente da Matriz de Qualidade do Usuário.

A Matriz de Qualidade Externa é formada pela *correlação* da Tabela de Características de Qualidade, proveniente da Matriz de Qualidade do Usuário, com a Tabela de Qualidade Externa. A Tabela de Qualidade Externa é formada pelas subcaracterísticas de qualidade externa definidas na norma ABNT (2003). Essas subcaracterísticas são correlacionadas com os atributos de qualidade para construir a Matriz de Qualidade Externa.

Para construir a Tabela de Qualidade Externa são utilizadas as subcaracterísticas de qualidade externa definidas na norma ABNT (2003), que estão associadas às características de qualidade.

A *correlação* realizada na Matriz de Qualidade Externa deve seguir os passos do método de aferição da qualidade. Para executar esses passos deve-se inserir uma atividade no fluxo de Desenho do Praxis. A atividade deve seguir o processo semelhante ao realizado na Matriz de Qualidade em Uso.

Depois de realizada a *correlação* na Matriz de Qualidade Externa deve-se calcular o grau de importância das subcaracterísticas de qualidade externa. As subcaracterísticas de qualidade externa priorizadas são armazenadas em uma nova seção intitulada “Qualidade Externa” no documento PQSw do Praxis apresentados na seção 1.6 deste trabalho. As informações contidas nessa seção podem auxiliar o planejamento das atividades das fases de Construção e Transição do Praxis. Essas informações podem guiar o planejamento dos casos de testes de unidade, identificando os testes adequados às necessidades do usuário.

Com o grau de importância das subcaracterísticas de qualidade externa e o Nível de Avaliação é possível selecionar medidas de qualidade externa adequadas para mensurar a qualidade esperada pelo usuário. A seleção das medidas deve ser realizada no sub-fluxo Planejamento do fluxo de Gestão de Projeto do Praxis e devem ser registradas no artefato MPPSw apresentados na seção 1.6 deste trabalho. Dessa forma, a qualidade em uso esperada pelo usuário pode ser acompanhada durante a execução do projeto de desenvolvimento

Depois da *conversão* das subcaracterísticas de qualidade externa da Matriz de Qualidade Externa é possível construir a próxima matriz do método de aferição de qualidade, que é a Matriz de Qualidade Interna.

4.3.4 Instanciação da Matriz de Qualidade Interna ao Praxis

A Matriz de Qualidade Interna tem por objetivo auxiliar na priorização das medidas interna de qualidade, que são coletadas durante o desenvolvimento de um projeto de *software*. A priorização dessas medidas será realizada através da sua *correlação* com as subcaracterísticas de qualidade externa definidas na norma ABNT (2003).

A Matriz de Qualidade Interna correlaciona a Tabela de Qualidade Externa com a Tabela de Qualidade Interna. A Tabela de Qualidade Externa é formada pelas subcaracterísticas de qualidade externa priorizadas, proveniente da Matriz de Qualidade Externa. Já a Tabela de Qualidade Interna é formada pelas medidas internas da qualidade do produto.

As subcaracterísticas de qualidade externa estão priorizadas e registradas na seção “Qualidade Externa” do PQSw do Praxis apresentados na seção 1.6 deste trabalho. Essa priorização será utilizada para realizar a *conversão* das medidas internas de qualidade.

Para construir a Tabela de Qualidade Interna são utilizadas as medidas de qualidade interna de produto. A qualidade interna é o total das características de um produto de *software* a partir da visão interna. No Praxis, as características da qualidade interna são formadas pelos artefatos produzidos durante a fase de Elaboração e Construção. As atividades dessas fases têm por objetivo modelar e implementar um produto que atenda à qualidade esperada pelo usuário. Para mensurar a qualidade o Praxis prevê algumas medidas internas, como por exemplo, defeitos de revisões, defeitos de testes de aceitação e defeitos de avaliação pelos usuários. Geralmente, essas medidas não englobam todas as características esperadas do produto pelo usuário; sendo assim, a Tabela de Qualidade Interna deve ser completada com medidas de qualidade interna, julgadas importantes. A tabela pode ser completada, por exemplo, com as medidas classificadas de acordo com o modelo de qualidade da norma ABNT (2003).

A construção da Matriz de Qualidade Interna deve seguir os passos do método de aferição da qualidade, descritos na seção 4.1.6 deste documento. A priorização das medidas de qualidade interna deve ser realizada no sub-fluxo Planejamento do fluxo de Gestão de Projeto do Praxis e devem ser registradas no artefato MPPSw apresentados na seção 1.6 deste trabalho. Essas medidas serão coletadas durante o desenvolvimento

do produto para o acompanhamento da qualidade interna dos artefatos produzidos pelo processo Praxis.

A coleta das medidas de qualidade deve ser realizada no processo Praxis pelo sub-fluxo Garantia da Qualidade do fluxo de Gestão da Qualidade. Os resultados da coleta devem ser armazenados em uma nova seção intitulada “Medidas de Qualidade Interna” no relatório RAPS_{Sw} (Relatório de Acompanhamento de Projeto de *Software*) do Praxis apresentados na seção 1.6 deste trabalho. As informações contidas nessa seção são utilizadas para acompanhar a qualidade interna do produto e para contemplar a qualidade externa esperada pelo usuário.

5 ESTUDO DE CASO E DISCUSSÃO: Aplicação do método de aferição da qualidade proposto a um produto de software

Os resultados obtidos com a aplicação do método de aferição da qualidade proposto em um produto de *software* serão apresentados a seguir. Durante a aplicação foram realizadas algumas avaliações qualitativas dos benefícios esperados do método.

Nas seções seguintes será apresentada e discutida a aplicação do método proposto ao produto MotorcyclesControl através da utilização da matrizes.

5.1 Método: Aplicação da Matriz de Qualidade do Usuário ao MotorcyclesControl

A Matriz de Qualidade do Usuário tem por objetivo auxiliar a padronização da qualidade do produto esperada pelo usuário. Essa matriz é formada pela Tabela de Requisitos Funcionais e pela Tabela de Características de Qualidade. O preenchimento dessas tabelas com os dados do produto MotorcyclesControl para a construção da Matriz de Qualidade do Usuário, será descrito a seguir.

Para preencher a Tabela de Requisitos Funcionais foi utilizada a lista de CUA (Casos de Uso de Análise) do modelo CRSw e MASw. Essa lista é composta por:

- Gestão Manual de Estoque;
- Gestão de Mercadorias;
- Gestão de Fornecedores;
- Gestão de Pedidos de Compra;
- Operação de Venda;
- Relatório de Estoque Baixo;
- Relatório de Mercadorias;
- Relatório de Fornecedores;
- Relação de Pedidos de Compra;
- Emissão de Nota Fiscal.

Cada caso de uso representa uma função que agrega valor ao produto, do ponto de vista do usuário. Dessa forma, essa lista representa o total de valor esperado pelo usuário do MotorcyclesControl. Esse valor agregado está documentado na Matriz de Qualidade do Usuário e será correlacionado com a Tabela de Características de Qualidade.

No MotorcyclesControl, a Tabela de Características de Qualidade é obtida usando os atributos de qualidade obtidos a partir do modelo MASw e do documento PESw (Proposta de Especificação de *Software*). O documento PESw fornece uma lista dos benefícios esperados pelo usuário, utilizados para fazer a *extração* dos atributos de qualidade. No caso do MotorcyclesControl, os atributos de qualidade foram completados pela lista dos requisitos não-funcionais e as restrições de memória, obtidos a partir do modelo MASw. A junção das informações desses dois artefatos compõe a lista de atributos de qualidade descritos a seguir:

- Economia de mão-de-obra na operação de venda;
- Agilidade na compra e venda de mercadorias;
- Diminuição de erros nas vendas;
- Diminuição do tempo de venda;
- Diminuição dos prejuízos na operação de venda;
- Diminuição dos erros nas notas fiscais;
- Eliminação da duplicidade de pedidos;
- Tempo de operação de venda deverá gastar no máximo 2 segundos;
- Tempo de operação de pesquisa de 10 segundos;
- Desenhado de forma que possa ser expandido para mais de um terminal de caixa;
- Leiaute do relatório Nota Fiscal deve seguir padrão da Secretaria da Receita;

- Um operador de caixa deverá ser capaz de aprender a operar o MotorcyclesControl com 1 dia de treinamento;
- Restringir o acesso dos usuários às funções por meio de senhas, conforme o respectivo grupo;
- O produto deve ocupar no máximo 200 MB em disco (sem considerar as bases de dados);
- O produto deve executar em 128 MB de memória RAM;

A lista de atributos representa a qualidade esperada pelo usuário do produto MotorcyclesControl. Essa qualidade deverá ser acompanhada durante o desenvolvimento do projeto com o objetivo de garantir que o produto final esteja dentro das expectativas do usuário. O acompanhamento da qualidade é auxiliado pelo método, pois documenta a qualidade de forma clara para consulta de todos envolvidos no projeto.

Para correlacionar a Tabela de Características de Qualidade e a Tabela de Requisitos Funcionais foram utilizadas as informações disponíveis nos artefatos do produto MotorcyclesControl. Do artefato MASw foi utilizada a aba “Influências” da planilha “Priorização das funções”, exibida na Figura 23, que representa a relação das funções com os benefícios esperados do MotorcyclesControl. No relacionamento da planilha, os valores 10, 4 e 1 representam, respectivamente, uma influência alta, média e baixa. O valor de cada influência foi utilizado para realizar a *correlação* dos atributos de qualidade extraídos dos benefícios do MotorcyclesControl

INFLUÊNCIAS		Peso do Benefício	Gestão Manual de Estoque	Relatório de Estoque Baixo	Relatório de Mercadorias	Relatório de Fornecedores	Relação de Pedidos de Compras	Gestão de Mercadorias	Gestão de Fornecedores	Gestão de Pedidos de Compras	Operação de Venda	Emissão de Nota Fiscal
Número	Benefício											
1	Diminuição de erros na venda de mercadorias.	10						10	1			
2	Qualidade na emissão da nota fiscal e ticket de venda, em relação à emissão manual.	10									10	10
3	Identificação de distorções entre o vendido e o estoque.	10	10	4				4			10	
4	Agilidade na compra de mercadorias.	4		1	4	4	4	4		4	4	
5	Economia de mão-de-obra.	4	4	1	1	1	1	1	1	4	1	10
6	Diminuição do custo de estocagem.	4	10	4	1			4		1		
7	Identificação de produtos mais e menos vendidos.	4	1				4	4			4	
8	Conhecimento do mercado de fornecedores.	1			1	4		4	10			
9	Indicação de promoções.	1	10		1			4			4	

Figura 23 – Planilha de priorização das funções do MotorcyclesControl

Para realizar a *correlação* dos requisitos não-funcionais verificaram-se quais funções eram influenciadas por esses requisitos. O requisito não-funcional “Tempo de operação de venda deverá gastar no máximo 2 segundos” tem uma forte correlação com a função “Operação de Venda”, sendo, assim, registrado como tendo uma correlação alta na Matriz de Qualidade do Usuário como por exemplo. Para construir o restante da matriz, mostrada na Figura 24, foi considerado a *correlação* de todas as funções com os atributos de qualidade do produto.

Características de Qualidade

Requisitos Funcionais	Características de Qualidade																Total
	Importância	Tempo de operação de pesquisa 10s	Diminuição de erros nas vendas	Diminuição do tempo de venda	Diminuição dos prejuízos na operação de venda	Economia de mão de obra na operação de venda	Tempo de operação de venda deverá gastar no máximo	Um operador de caixa deverá ser capaz de aprender e o MotorcyclesControl com 1 dia de treinamento	Agilidade na compra e venda de mercadorias	Diminuição dos erros nas notas fiscais	Restringir o acesso dos usuários às funções através de conforme o respectivo grupo	O produto deve ocupar no máximo 200 MB (sem considerar as bases de dados)	O produto deve executar em 128 MB	Eliminação da duplicidade de pedidos	Leiaute do relatório Nota Fiscal deve ser aprovado pela Secretaria da Receita Federal	Desenhado de forma que possa ser expandido para mais de um terminal de caixa	
Gestão de Mercadorias	4	A	M	B	B				M	B	B	B	B				88
Gestão Manual de Estoque	4	A									B	B	B				48
Operação de Venda	4		A	A	A	A	A	A	M	M	B	B	B				252
Emissão de Nota Fiscal	2		A	A	A	A	M	B		A	B	B	B		A		122
Gestão de Fornecedores	1	A							B		B	B	B	M			16
Gestão de Pedidos de Compra	1	A							M		B	B	B	A			24
Relação de Pedidos de Compra	1								M		B	B	B	B			7
Relatório de Estoque Baixo	1								B		B	B	B				4
Relatório de Fornecedores	1								B		B	B	B	B			5
Relatórios de Mercadorias	1								M		B	B	B	B			7
Total		90	66	58	58	54	42	38	36	34	20	20	20	19	18	0	

Figura 24 – Matriz de Qualidade do Usuário para o MotorcyclesControl

Nessa matriz, o requisito não-funcional “Desenhado de forma que possa ser expandido para mais de um terminal de caixa” não foi relacionado, pois no produto MotorcyclesControl esse requisito está adiado, como mostrado na seção “Requisitos Adiados” do MASw.

Depois da *correlação* realizada na Matriz de Qualidade do Usuário, é calculado o grau de importância dos atributos de qualidade. Na matriz exibida na Figura 24, os atributos de qualidade já estão ordenados pelo grau de importância calculado.

Através da *correlação* das funções com os atributos de qualidade podem-se verificar quais os atributos de qualidade são prioritários para o usuário do produto. O gráfico de importância obtido para todos os atributos de qualidade do MotorcyclesControl está mostrado na Figura 25. Nesse gráfico do MotorcyclesControl, o atributo de qualidade mais importante é “Tempo de operação de pesquisa de 10 segundos” e o menos importante é “Leiaute do relatório Nota Fiscal deve seguir padrão da Secretaria da Receita”. Isso demonstra que o usuário do MotorcyclesControl está mais preocupado com a rapidez que os resultados das pesquisas são retornados, e assim, esse atributo de qualidade deve ser priorizado na fase de Construção do produto.

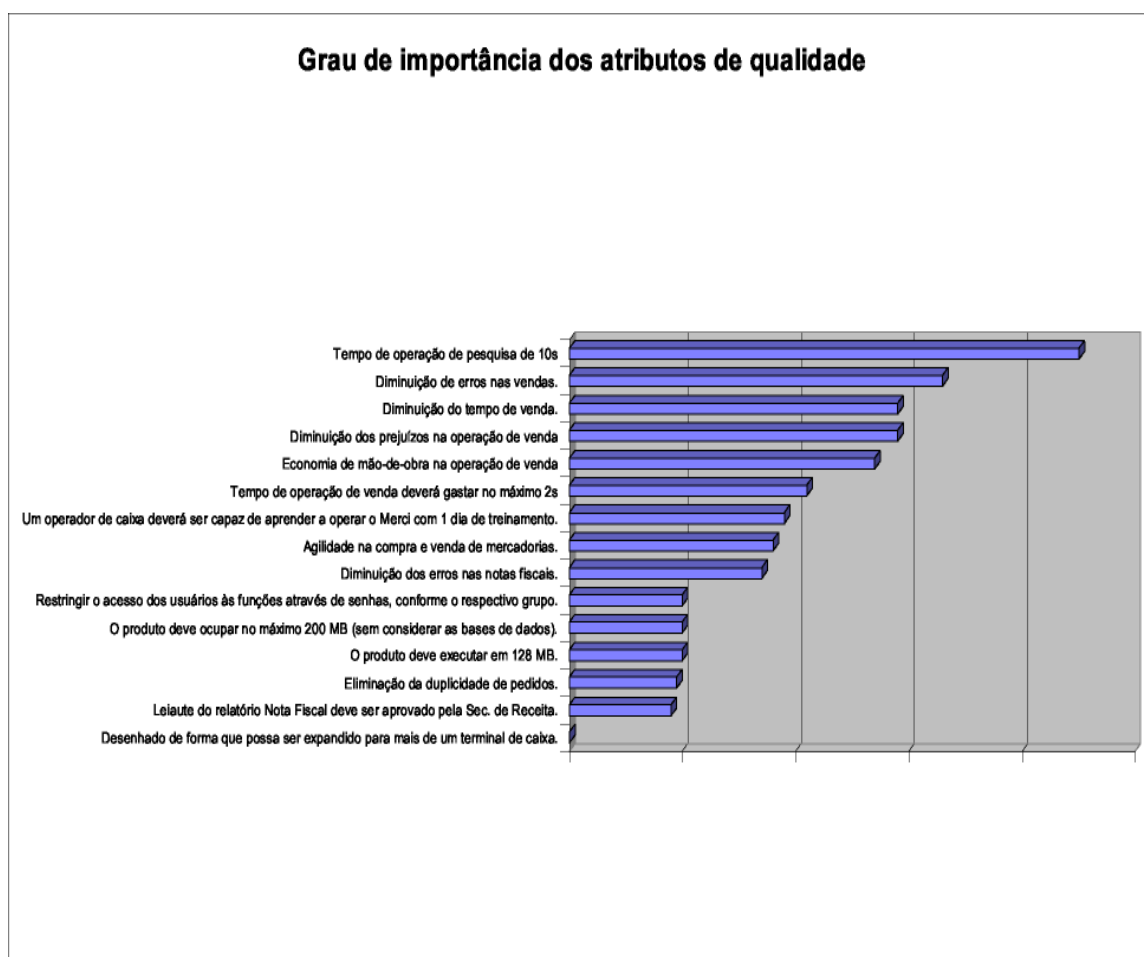


Figura 25 – Gráfico de importância dos atributos de qualidade

A priorização dos atributos de qualidade esperados pelo usuário é utilizada no planejamento e no controle das atividades do desenvolvimento de um produto. Por exemplo, nas atividades do fluxo de Gestão da Qualidade do Praxis pode ser considerada a priorização dos atributos no dimensionamento do esforço e da cobertura desejada para a qualidade do produto. No Praxis, o esforço e cobertura planejados da qualidade são registrados no documento PQSw e são acompanhados durante o desenvolvimento do produto.

Os atributos de qualidade priorizados serão utilizados para construir a Matriz de Qualidade em Uso e a Matriz de Qualidade Externa.

5.2 Método: Aplicação da Matriz de Qualidade em Uso ao MotorcyclesControl

A Matriz de Qualidade em Uso foi construída para o MotorcyclesControls com o objetivo de conferir padronização à qualidade esperada pelo usuário. A Tabela de Características de Qualidade e a Tabela de Qualidade em Uso foram correlacionadas na construção da Matriz de Qualidade em Uso para o produto.

A Tabela de Características de Qualidade é composta pela lista priorizada dos atributos de qualidade, proveniente da Matriz de Qualidade do Usuário. Já a Tabela de Qualidade em Uso é formada pelas características de qualidade em uso definidas na norma ABNT (2003), sendo elas: efetividade, produtividade, segurança e satisfação.

A *correlação* dos atributos de qualidade do MotorcyclesControl com as características de qualidade em uso é apresentada na Figura 26.

Alguns atributos de qualidade do produto não estão relacionados com a qualidade em uso de acordo com a norma ABNT (2003) por não se relacionarem com nenhuma das características de qualidade em uso da norma. Portanto, não é necessário que todos os atributos estejam relacionados com a qualidade em uso, pois alguns desses estão relacionados com a qualidade externa definida na norma.

Características de Qualidade	Características Qualidade em Uso					Total
	Importância	Produtividade	Efetividade	Segurança	Satisfação	
Tempo de operação de pesquisa 10s	90	B				90
Diminuição de erros nas vendas	66	B	M	B		330
Diminuição do tempo de venda	58	M				174
Diminuição dos prejuízos na operação de venda	58		M	B		232
Economia de mão de obra na operação de venda	54	A				486
Tempo de operação de venda deverá gastar no máximo 2s	42	M				126
Um operador de caixa deverá ser capaz de aprender e operar o MotorcyclesControl com 1 dia de treinamento	38					0
Agilidade na compra e venda de mercadorias	36	A				324
Diminuição dos erros nas notas fiscais	34		M	B		136
Restringir o acesso dos usuários às funções através de senhas, conforme o respectivo grupo	20			A		180
O produto deve ocupar no máximo 200 MB (sem considerar as bases de dados)	20					0
O produto deve executar em 128 MB	20					0
Eliminação da duplicidade de pedidos	19		M	B		76
Leiaute do relatório Nota Fiscal deve ser aprovado pela Secretaria da Receita Federal	18					0
Desenhado de forma que possa ser expandido para mais de um terminal de caixa	0					0
Total		1266	531	357	0	

Figura 26 - Matriz de Qualidade em Uso para o MotorcyclesControl

A construção da Matriz de Qualidade em Uso possibilita verificar quais características de qualidade em uso da norma ABNT (2003) são mais importantes para o usuário. Com o resultado da matriz, exibido no gráfico da Figura 27, percebe-se que a construção do MotorcyclesControl deve priorizar a Produtividade e Efetividade. Mas isso não quer dizer que, a característica de segurança deve ser desconsiderada no desenvolvimento do produto.

No MotorcyclesControl não há nenhum atributo de qualidade relacionado explicitamente com a característica Satisfação. Isso também pode acontecer com outros produtos, pois a satisfação geralmente faz parte das expectativas do usuário que estão na curva básica, do modelo de Kano (1996). Dessa forma, o usuário não levanta claramente um atributo de qualidade, mesmo esperando que a característica de qualidade esteja contemplada no produto. Portanto, é importante a empresa fornecedora avaliar as características de qualidade zeradas e, se necessário, selecionar medidas de qualidade.

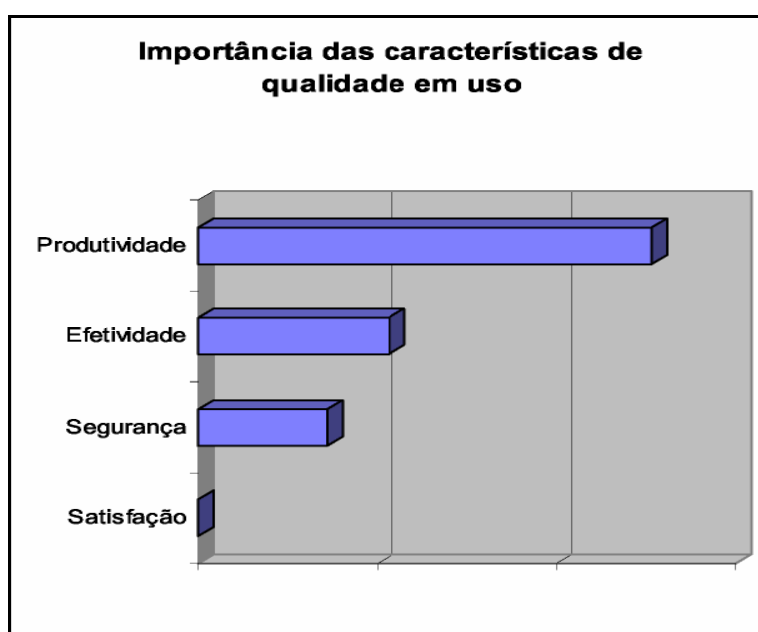


Figura 27 - Gráfico de importância das características de qualidade em uso

A priorização das características da Tabela de Qualidade em Uso também é utilizada para selecionar as medidas da qualidade esperada pelo usuário. Com o grau de importância calculado pela *conversão* da Matriz de Qualidade em Uso e o Nível de Avaliação dessas características é possível priorizar as medidas de qualidade em uso. Destaca-se a característica Produtividade que tem o grau de importância maior que a característica Segurança, sendo assim, devem-se selecionar mais medidas para sua avaliação.

Para identificar o Nível de Avaliação das características de qualidade em uso deve-se avaliar o impacto de uma consequência de uma falha em cada característica. Como o MortocyclesControl é um produto que controla a movimentação do caixa de uma loja, a característica Produtividade tem o nível A por interagir diretamente com questões de tempo de atendimento enquanto Efetividade e Segurança tem o nível B Figura 28.

Características Qualidade em Uso	Nível de avaliação
Efetividade	B
Produtividade	A
Segurança	B
Satisfação	D

Figura 28 – Níveis de avaliação para características de qualidade em uso

Os atributos de qualidade priorizados, da Tabela de Características de Qualidade, também serão utilizados na construção da Matriz de Qualidade Externa, descrita na seção seguinte.

5.3 Método: Aplicação da Matriz de Qualidade Externa ao MotorcyclesControl

A Matriz de Qualidade Externa é construída para o produto MotorcyclesControl com o objetivo de padronizar a qualidade esperada pelo usuário. Essa matriz é composta pela Tabela de Características de Qualidade, proveniente da Matriz de Qualidade do Usuário, e pela Tabela de Qualidade Externa.

A Tabela de Qualidade Externa é formada pelas subcaracterísticas de qualidade externa definidas na norma ABNT (2003). Essa tabela é organizada colocando a letra inicial da característica na frente de cada subcaracterística na tabela para mostrar o relacionamento das subcaracterísticas com sua característica.

A *correlação* das tabelas foi realizada executando os passos da instanciação do método, descritos na seção 4.3.3. deste documento. A *correlação* de todas subcaracterísticas de qualidade para o produto está ilustrada na Figura 29.

Características de Qualidade	Características Qualidade Externa																		Total										
	Importância	E - Comportamento em relação ao tempo	U - Operacionalidade	F - Acurácia	E - Comportamento em relação aos recursos	U - Apreensabilidade	C - Maturidade	F - Segurança de acesso	F - Conformidade	U - Iteigibilidade	U - Atratividade	U - Conformidade	C - Tolerância a falha	F - Adequação	C - Recuperabilidade	C - Conformidade	E - Conformidade	M - Analisabilidade		M - Modificabilidade	M - Estabilidade	M - Testabilidade	M - Conformidade	P - Adaptabilidade	P - Capacidade para ser instalado	P - Co-existência	P - Capacidade para substituir	P - Conformidade	
Tempo de operação de pesquisa 10s	90	A																											810
Diminuição de erros nas vendas	66		B	B			M																						330
Diminuição do tempo de venda	58	M	M																										348
Diminuição dos prejuízos na operação de venda	58			M																									174
Economia de mão de obra na operação de venda	54	B	M																										216
Tempo de operação de venda deverá gastar no máximo 2s	42	A	B																										420
Um operador de caixa deverá ser capaz de aprender e operar o MotorcyclesControl com 1 dia de treinamento	38		M			A				M	B	B																	646
Agilidade na compra e venda de mercadorias	36	M	B																										144
Diminuição dos erros nas notas fiscais	34		B				M																						136
Restringir o acesso dos usuários às funções através de senhas, conforme o respectivo grupo	20				A																								180
O produto deve ocupar no máximo 200 MB (sem considerar as bases de dados)	20			A																									180
O produto deve executar em 128 MB	20					A																							180
Eliminação da duplicidade de pedidos	19			A			B					B																	209
Layout do relatório Nota Fiscal deve ser aprovado pela Secretaria da Receita Federal	18								A																				162
Desenhado de forma que possa ser expandido para mais de um terminal de caixa	0																						M	A	B				0
Total		1524	628	411	360	342	319	180	162	114	38	38	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figura 29 - Matriz de Qualidade Externa para o MotorcyclesControl

O atributo de qualidade “Desenhado de forma que possa ser expandido para mais de um terminal de caixa” está relacionado com três subcaracterísticas de qualidade, porém seu grau de importância é zero, por ser um requisito adiado. Dessa forma, as subcaracterísticas de qualidade correlacionadas com esse atributo ficam com seu grau de importância zero também e não são priorizadas para qualidade externa do produto.

Na Matriz de Qualidade Externa, nenhum dos atributos de qualidade está relacionado com a subcaracterística Interoperabilidade, apesar de que o MotorcyclesControl possui integração com o software Sistema Financeiro. Nos artefatos disponíveis no MortorcyclesControl, não fica claro se faltou levantar com o usuário os atributos de qualidade para a subcaracterística de interoperabilidade. Porém, a identificação da falta desse atributo correlacionado com a subcaracterística possibilita o arquiteto de um produto como o MortocyclesControl solicitar esclarecimento dessa dúvida com o usuário. Dessa forma, o método auxiliaria o desenvolvedor na indicação de características de qualidade importantes para o usuário do produto.

Observa-se que na Figura 29 que nenhum atributo de qualidade é relacionado com as subcaracterísticas da característica Manutenibilidade. Normalmente, usuários de produtos de software não levantam os atributos de qualidade relacionados com essa característica. Essa subcaracterística está mais relacionada com a qualidade desejada pela empresa desenvolvedora do produto, pois problemas com essa característica impactam o custo de um projeto de manutenção. Nesse caso, o método poderia ser estendido para incorporar a qualidade do produto esperada pela empresa desenvolvedora. Para a incorporação da qualidade esperada pela empresa desenvolvedora no método, no entanto, seria necessário ponderar os atributos de qualidade esperados pelo usuário com os atributos de qualidade esperados pela empresa.

Pode-se observar ainda na Figura 29 que vários atributos de qualidade do produto MotorcyclesControl não foram relacionados com a subcaracterística Adequação da característica Funcionalidade, semelhante com que aconteceu com a característica Satisfação na Matriz de Qualidade em Uso. Portanto, devem-se ser selecionadas medidas para avaliar a subcaracterística Adequação, mesmo que não tenha nenhum relacionamento na Matriz de Qualidade Externa.

As subcaracterísticas de qualidade foram priorizadas de acordo com sua relação com os atributos de qualidade de software. A priorização dessas subcaracterísticas é mostrada no gráfico apresentado na Figura 30 que é utilizada para priorizar as medidas de qualidade coletadas no desenvolvimento. Contudo, é importante salientar que a priorização não significa que as subcaracterísticas de qualidade menos importantes não devam ser mensuradas.

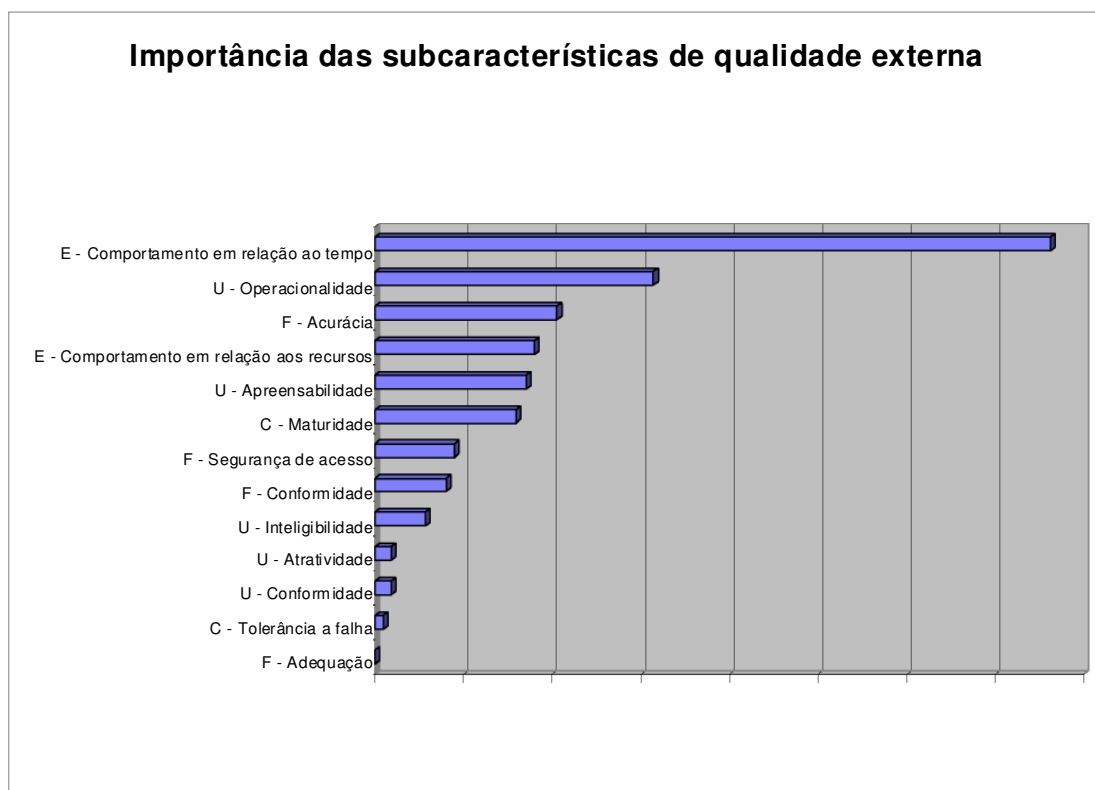


Figura 30 - Gráfico de importância das subcaracterísticas de qualidade externa

A priorização das subcaracterísticas da Tabela de Qualidade Externa também é utilizada para selecionar as medidas da qualidade esperada pelo usuário. Com o grau de importância calculado pela *conversão* da Matriz de Qualidade Externa e o Nível de Avaliação dessas subcaracterísticas é possível escolher as medidas de qualidade externa. A característica Eficiência tem o maior grau de importância para o MotorcyclesControl, e assim, deve-se selecionar um maior número de medidas para essa característica. Essas medidas são utilizadas para verificar a qualidade externa do produto no seu desenvolvimento.

Como o MotorcyclesControl é um produto com uma grande interação com seu usuário, as características Funcionalidade, Usabilidade e Eficiência estão associadas com o Nível de Avaliação A como mostrado na Figura 31.

Características Qualidade Externa	Nível de avaliação
Funcionalidade	A
Confiabilidade	B
Usabilidade	A
Eficiência	A
Manutenabilidade	D
Portabilidade	D

Figura 31 - Níveis de avaliação para características de qualidade externa

A Tabela de Qualidade Externa é utilizada para a construção da Matriz de Qualidade Interna do método, descrita na seção seguinte.

5.4 Método: Aplicação da Matriz de Qualidade Interna ao MotorcyclesControl

A Matriz de Qualidade Interna tem o objetivo de auxiliar na priorização das medidas de qualidade interna, que serão coletadas durante o desenvolvimento do produto. Essa matriz é composta pela Tabela de Qualidade Externa, proveniente da Matriz de Qualidade Externa, e a Tabela de Qualidade Interna.

A Tabela de Qualidade Interna é formada pelas medidas de qualidade do produto MotorcyclesControl e pelas medidas de qualidade interna da norma ABNT (2003). As medidas de qualidade do produto são obtidas da aba “Defeitos ” do relatório RAPS w. As medidas de qualidade interna da norma ABNT (2003) foram selecionadas de forma que cada subcaracterística de qualidade externa possuísse, pelo menos, uma medida de qualidade interna.

As medidas de qualidade interna selecionadas do produto MotorcyclesControl são:

- Defeitos de Revisões;
- Defeitos de Testes de aceitação;
- Defeitos de Avaliação pelo usuário;
- Defeitos de auditoria de qualidade.

Algumas medidas do produto MotorcyclesControl são acompanhadas durante o desenvolvimento, porém não estão ligadas diretamente com a qualidade interna, como medidas de tamanho e esforço. As medidas de tamanho são utilizadas para normalizar os dados das medidas de qualidade coletados entre projetos, não sendo utilizadas para construção da Matriz de Qualidade Interna.

Para completar a Tabela de Qualidade Interna foi escolhida pelo menos uma medida de qualidade interna da norma ABNT (2003) para cada subcaracterística. O nome

dessas medidas foi deixado em inglês para facilitar a busca dessas nas tabelas de definição de medidas da norma, sendo elas:

- E - *Response time*;
- U - *Input Validity checking*;
- F - *Computational Accuracy*;
- E - *Memory utilization*;
- U - *Completeness of description*;
- C - *Fault removal*;
- F - *Access auditability*;
- F - *Functional compliance*;
- U - *Completeness of user documentation and/or help facility*;
- U - *Attractive interaction*;
- U - *Usability compliance*;
- C - *Failure avoidance*;
- F - *Functional adequacy*.

As medidas de qualidade interna e as subcaracterísticas de qualidade externa foram correlacionadas para construir a Matriz de Qualidade Interna. As medidas selecionadas do MotorcyclesControl foram relacionadas com as subcaracterísticas de qualidade externa utilizando as informações das suas listas de conferência. Essas listas possibilitam perceber o que foi verificado durante as revisões realizadas no desenvolvimento do MotorcyclesControl, sendo elas: LCIDESw (Lista Conferência de Inspeção do Desenho Externo de Software), LCIDTSw (Lista Conferência de Inspeção dos Testes de Software), LCIDISw (Lista Conferência de Inspeção do Desenho Interno de Software), LCIIISw (Lista Conferência de Inspeção da Implementação e Desenho Detalhado de Software), LCAUSw (Lista Conferência de Inspeção da Avaliação do Usuário de Software), LCAQSw (Lista Conferência da Auditoria de Qualidade de

Software). O relacionamento das medidas de qualidade interna da norma ABNT (2003) com as subcaracterísticas de qualidade externa foi realizado através da sua classificação, descrita na norma.

A relação entre as medidas de qualidade interna e as subcaracterísticas de qualidade externa está demonstrada na Matriz de Qualidade Interna, ilustrada na Figura 32.

Características Qualidade Externa	Métricas de Qualidade Interna																		
	Importância	E - Response time	Defeitos de avaliação pelo usuário	Defeitos de testes de aceitação	U - Input Validity checking	F - Computational accuracy	E - Memory utilization	U - Completeness of description	C - Fault removal	F - Access audibility	F - Function compliance	U - Completeness of user doc. And/or facility	Defeitos de revisões	U - Attractive interaction	U - Usability compliance	C - Failure avoidance	F - Function adequacy	Defeitos de auditoria de qualidade	Total
E - Comportamento em relação ao tempo	1524	A		M															41148
U - Operacionalidade	628		A		A														50868
F - Acurácia	411			M		A							B						11097
E - Comportamento em relação aos recursos	360			M			A												9720
U - Apreensibilidade	342		A					A											27702
C - Maturidade	319			B					A										2871
F - Segurança de acesso	180			M						A				B					4860
F - Conformidade	162			M							A			B					4374
U - Iteligibilidade	114		A									A							9234
U - Atratividade	38		A											A					3078
U - Conformidade	38		A												A				3078
C - Tolerância a falha	19			B												A			171
C - Conformidade	0																		0
C - Recuperabilidade	0																		0
E - Conformidade	0																		0
F - Adequação	0			M									M				A		0
F - Iteroperabilidade	0			B															0
M - Analisabilidade	0												B						0
M - Modificabilidade	0												B						0
M - Estabilidade	0																		0
M - Testabilidade	0																		0
M - Conformidade	0																		0
P - Adaptabilidade	0																		0
P - Capacidade para ser instalado	0																		0
P - Co-existência	0			B															0
P - Capacidade para substituir	0			B															0
P - Conformidade	0																		0
Total		13716	10440	8249	5652	3699	3240	3078	2871	1620	1458	1026	753	342	171	0	0	0	

Figura 32 - Matriz de Qualidade Interna para o MotorcyclesControl

A *conversão*, realizada na Matriz de Qualidade Interna, e o Nível de Avaliação auxilia na priorização das medidas de qualidade interna. Por simplificação, foram utilizados os mesmos Níveis de Avaliação da Matriz de Qualidade Externa. Essa priorização, ilustrada no gráfico da Figura 33, permite o acompanhamento e controle da qualidade com objetivo de alcançar a qualidade externa do produto esperada pelo usuário. Porém, a seleção das medidas internas deve considerar também outros aspectos da organização, como por exemplo, o custo envolvido na coleta das medidas

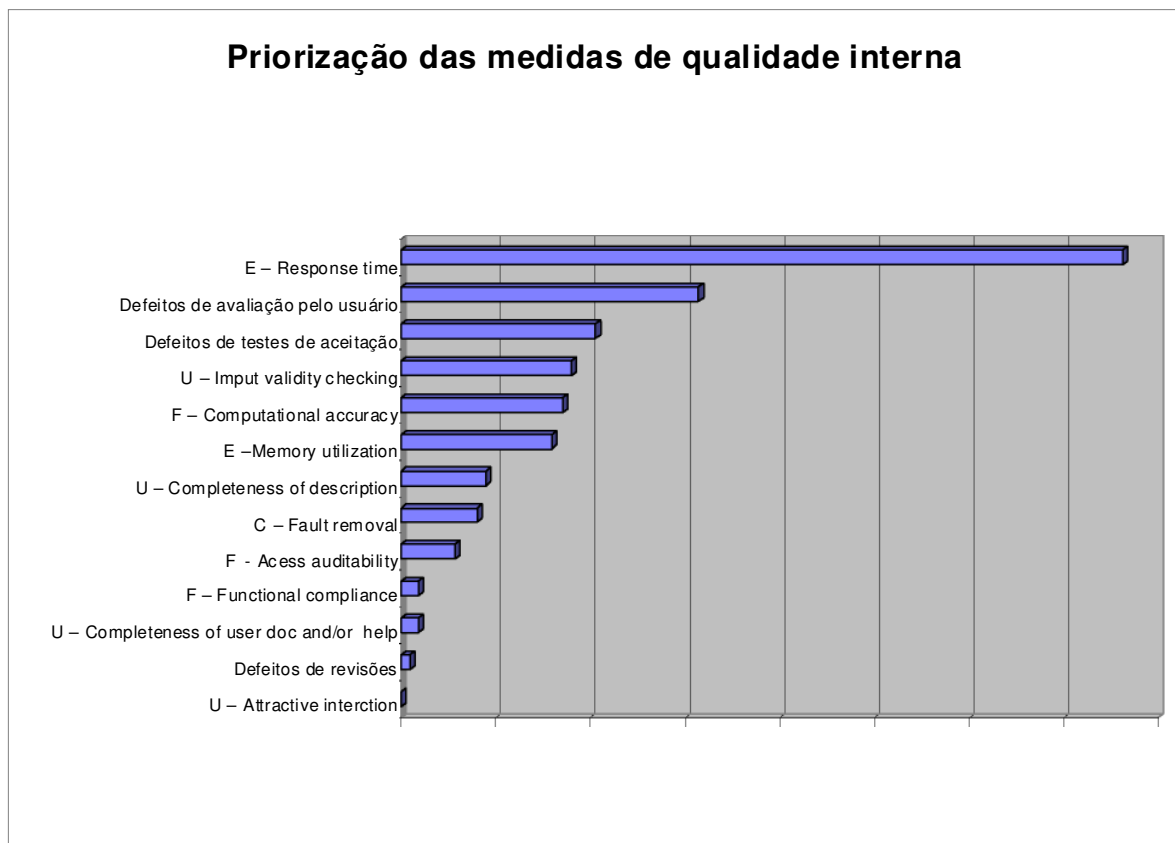


Figura 33 – Priorização das medidas internas de qualidade

A *conversão* realizada na Matriz de Qualidade Interna também pode auxiliar a confecção das listas de conferências dos artefatos do MotorcyclesControl. As listas de conferência são utilizadas para verificar padrões de qualidade dos artefatos confeccionados durante a fase de construção do produto. Dessa forma, as listas também

devem incluir itens que verifiquem a qualidade interna dos artefatos, visando alcançar a qualidade esperada pelo usuário.

Além da confecção das listas de conferência, essa *conversão* pode auxiliar na criação dos casos de testes de unidade. Esses testes de unidade são utilizados para verificar a implementação dos casos de uso do produto. Com a *conversão* realizada na matriz, os testes de unidade podem priorizar a característica de qualidade Eficiência.

A medida “Defeitos de auditoria de qualidade” não foi relacionada com nenhuma subcaracterística de qualidade, pois ela é uma medida de processo e não uma medida de produto. Contudo, medidas de processo também devem ser coletadas, pois a qualidade do produto depende da qualidade do processo. Porém, neste trabalho de pesquisa o foco de mensuração é a qualidade do produto e não qualidade do processo.

6 CONCLUSÕES

O presente estudo permite concluir que o método de aferição da qualidade de produtos de software utilizando QFD desdobra a qualidade esperada pelo usuário em medidas de qualidade.

Após definir medidas adequadas da qualidade esperada pelo usuário foi possível uma melhor priorização das atividades de garantia da qualidade.

O método permitiu mensurar e identificar falhas no produto desde o início da sua construção até a entrega ao cliente por meio do acompanhamento de medidas de qualidade do produto de software e por conseqüência aumentar a satisfação do seu usuário.

O estudo de caso demonstrou por meio da aplicação do método que o tempo de resposta do *software* é a característica de qualidade mais relevante para o usuário e o motivo de sua maior insatisfação com o MotorcyclesControl.

Através da mensuração e identificação de falhas no produto desde o início da sua construção é possível planejar os investimentos em qualidade para não extrapolar custos e alcançar seus objetivos de satisfazer as expectativas do usuário.

A aplicação do modelo de qualidade da norma ABNT (2003) possibilitou a padronização da qualidade exigida pelo usuário permitindo a criação de uma base histórica de projetos desenvolvidos na empresa que poderá auxiliar na previsão e planejamento de esforços de qualidade para desenvolvimento de projetos futuros.

A aplicação da norma ABNT (2003) na adaptação do Framework de Atributos de Fenton (FENTON, 1996) utilização pelo método, permitiu a classificação das medidas utilizadas pelo Praxis e também outras medidas encontradas na literatura.

A instanciação do método de aferição de qualidade ao processo Praxis, demonstrou que o método proposto pode ser facilmente incorporado e aplicado a qualquer processo de desenvolvimento de software.

6.1 Sugestões para trabalhos futuros

Sugerem-se como possíveis trabalhos futuros a serem realizados para completarem esta pesquisa:

- Especificação e construção de uma ferramenta automatizada para o método de aferição da qualidade com objetivo de facilitar a manipulação e acompanhamento de dados e informações em projetos de grande porte.
- Pesquisa de outros métodos de classificação de medidas visando aumentar a abrangência de medidas de qualidade de produtos de *software*.
- Como ocorre em processos de desenvolvimento de *software*, o método de aferição de qualidade proposto deve ser experimentado, avaliado, ajustado e melhorado continuamente.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ABNT – NBR ISO/IEC 9126:2003, **Engenharia de Software - Qualidade de Produto.**

ABRAN,A., **Guide to the Software Engineering Body of Knowledge (SWEBOK).** The Institute of Electrical and Electronics Engineers, Inc.(IEEE), 210 pages, 2008. Disponível em <<http://www.swebok.org/>>. Acesso em: 11 de outubro de 2008

AKAO, Y., **Introdução ao desdobramento da qualidade.** Belo Horizonte/MG: Fundação Christiano Ottoni, 1996.

AKAO, Y., **Quality Function Deployment: Integrating Customer Requirements into Product Design.** Cambridge: Productivity Press, 1990.

ANDERSSON, T., **A survey of software quality metrics.** Slides of course presentation, Department of Computer Science, 1990. Disponível em <citeseer.ifi.unizh.ch/andersson90survey.html> Acesso em: 30 de outubro de 2008.

APRIL, A., **Software Maintenance Maturity Model (SMMM): The software maintenance process model.** Journal of Software Maintenance and Evolution: Research and Practice, vol. 17(3), pp. 197-223, 2005.

BASIL, V. R., **Software Modeling and Measurement: The Goal/Question/Metric Paradigm.** Technical Report UMIACS-TR-92-96. University of Maryland USA., 2000.

BASIL, V.R., **The Future Engineering of Software: A Management Perspective.** IEEE Computer Society Press Los Alamitos, CA, USA,2006.

BROOKS, F., **Quality and Software Engineering.** Proceedings of the IFIP Tenth World Computing Conference, New Jersey,2006.

CAPPS, B., **Software Engineering Processes.** Slides of course presentation, Department of Computer Science and Engineering of Oregon Health & Science University. Disponível em < <http://www.cse.ogi.edu/class/cse503/>>. Acesso em 15 de outubro de 2008.

CASTANHO(1), F.R.; CARDOSO, A.A.; CHAVES, C.A., **Método QFD na Gestão de Projetos de Aplicações de Baixa Plataforma: Uma Abordagem para Gestão e Aferição da Qualidade de Produtos de Software.** IV Simpósio de Excelência em Gestão e Tecnologia – SEGeT, Associação Educacional Dom Bosco, Resende – RJ, 2007.

CASTANHO(2), F.R.; CARDOSO, A.A.; CHAVES, C.A., **DLFSS AND QFD: Application in a Service Company.** 19th Congress of Mechanical Engineering – Cobem. Brasília, 2007.

CASTANHO(3), F.R., **Lean Six Sigma através do Método QFD aplicado na Gestão de Projetos: Uma abordagem para a Gestão e Aferição da Qualidade de Produtos de Software.** Internacional TCS Iberoamericana Technical Architects Conference 2007 – TATA Group. Indian, 2007.

CHENG, L.C., FILHO, L.D.R.M. et al. **QFD: Desdobramento da Função Qualidade na Gestão de Desenvolvimento de Produtos**. São Paulo. Editora Blucher, 2007.

CHENG, L.C. et. al., **QFD: Planejamento da Qualidade**. Belo Horizonte: UFMG, Escola de Engenharia, Fundação Christiano Ottoni. Editora Littera Maciel Ltda., 1995.

CLARK, K. B., FIJIMOTO, T., **Product development performance: strategy, organization and management in the word auto industry**. Boston – Massachusetts: Harvard Business School Press, 1991.

COHEN, L., **Quality Function Deployment: how to make QFD work for you**. Massachusetts: Harvard Business School Press, 1991.

DEMING, W.E., **Quality, productivity, and competitive position**. Massachusetts Institute of Technology, Center for Advanced Engineering Study Cambridge, MA, 1982.

EMAM, E., **Software Engineering Process**. Chapter 9 of Guide to the Software Engineering Body of Knowledge (SWEBOK), pp 9-1 to 9-14. Disponível em: <<http://www.swebok.org/>>. Acesso em: 14 de outubro de 2008

EUREKA, W.H.; KYAN, N., **QFD – perspectivas gerenciais do desdobramento da função qualidade**. Rio de Janeiro: Qualitymark, 1992.

FALBO, R. A., **Integração de Conhecimento em Ambiente de Desenvolvimento de Software**. Tese de Doutorado. COPPE/UFRJ, 2002.

FARIA, F., **Um método para seleção de processos de software em programas de melhoria: caso de uma pequena empresa do setor de telecomunicações**. Monografia de conclusão do curso de Pós-Graduação Latu Sensu em Melhoria de Processo de Software da Universidade Federal de Lavras U FLA, 2005.

FEHLMANN, T., **Combinatory metrics for software development**. QFD Institute Transactions from the 2nd Symposium on Quality Function Deployment, QFD Institute, Deutschland, 2002.

FENTON, N., **Software Metrics; A Rigorous and Practical**. Approach. 2nd ed. London: International Thonson, 1996.

FERREIRA, J. S., **Descrevendo os processos do grupo de reuso da ISO/IEC 15504-5 para o padrão do CMMI**. Monografia de conclusão do curso de especialização de Qualidade no Desenvolvimento de Software da Faculdade Senac de Ciências Exatas e Tecnologia, 2005.

FIATES, G.G.S., **a Utilização do QFD como suporte à implementação do TQC em empresas do setor de serviços**. Dissertação de Mestrado em Engenharia de Produção. Florianópolis/SC: UFSC, 1995.

FUGGETTA, A., **Software Process**. A Roadmap, in Proceedings of Future of Software Engineering Workshop, ICSE, Limerick, Ireland, pp. 25-34, 2000.

GARVIN, D. A., **Gerenciando a Qualidade: a visão estratégica e competitiva**. Rio de Janeiro: Qualitymark, 1992.

GUIMARÃES, L. M., **QFD – Quality Function Deployment: Uma ferramenta de Suporte à Estratégia Competitiva**. Revista Controle da Qualidade. N° 56. 1996.

GUINTA, L.R., **Manual de QFD**. Rio de Janeiro. LTC – Livros Técnicos Científicos Ed. 1993.

HAUSER, J.R., CLAUSING, D., **The House of Quality**. Harvard Business Review, v. 66, n.3, p.63-73, may-june 1988.

HERBSLEB, J. D., **Splitting the Organization and Integrating the Code: Conway's Law Revisited**. Proceedings of ICSE. Los Angeles: IEEE, 2007.

IEEE, **Software Engineering Body of Knowledge - SWEBOOK**. Institute of Electrical and Electronics Engineers. IEEE SWEBOOK, 2004.

IEEE-610, **Standard Glossary of Software Engineering Terminology**. Institute of Electrical and Electronics Engineers. IEEE STD 610.12, 1990.

IEEE-982, **Standard Dictionary of Measures to Produce Reliable Software**. Institute of Electrical and Electronics Engineers, IEEE STD 982.1, 1988.

IEEE-1061, **Standard for a Software Quality Metrics Methodology**. Institute of Electrical and Electronics Engineers, IEEE STD 982.1, 1998.

ISO/IEC 8402, **The International Organization for Standardization - ISO/IEC 8402**. Quality management and quality assurance--vocabulary, 1995.

JACOBSON, I., **The Unified Software Development Process**. Reading. EUA, New York: Addison-Wesley, 1998.

JUNG, C. F., **Metodologia para pesquisa & desenvolvimento: aplicada a novas tecnologias, produtos e processos**. Rio de Janeiro: Axcel Books do Brasil Editora, 2004.

JURAN, J. M., **Juran's Quality Handbook**. 5th ed. McGraw-Hill Professional, 1872p.,1998.

JURAN, J. M., **A Qualidade desde o Projeto – Novos Passos para o Planejamento da Qualidade em Produtos e Serviços**. 3ª Edição. São Paulo: Editora Pioneira, 1997.

KANO, N., SERAKU, N., TAKAHASHI, F. AND TSUJI, S., **Attractive quality and must-bequality**. In The best on quality, edited by John D. Hromi. Volume 7 of the BookSeries of the International Academy for Quality. Milwaukee:ASQC Quality Press, 1996.

KASSE, T., **Practical Insight into CMMI**. Artech House Publishers, 281 pages, 2004.

KING, B., **Better Designs in Half the Time: Implementing QFD in America**. Methuen: GOAL/QPC, 1989.

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

KUMAR, A., **Estimating Software Projects**, ACM SIGSOFT, p.60, New York, 2001.

LIPHAUS, E.E., **Desenvolvimento de produto em pequenas empresas com a utilização do método QFD**. Mestrado em Engenharia Mecânica. Taubaté – SP. Universidade de Taubaté, 2006..

LUIGI L.; BARRESI G., **Automated support for process-aware definition and execution of measurement plans**. Boston, EUA: The Rational Edge Press, 2006.

MARTORANO, E., **O QFD no projeto e desenvolvimento de produtos com ênfase nas quatro fases**. Dissertação de Mestrado. Florianópolis: UFSC, 1993.

MCFEELEY B., **IDEAL - A User's Guide for Software Process Improvement**. Handbook CMU/SEI-96-HB-001, 236 pages, 2006.

MIGUEL, P.A.C.; CARNEVALLI, J.A., **Aplicações não convencionais do desdobramento da função qualidade**. São Paulo: Artliber Editora, 2006.

MIZUNO, S., **Gerência para Melhoria da Qualidade: As Sete Novas Ferramentas de Controle da Qualidade**. LTC, RJ, 1993.

MOURA, E.C., **As sete ferramentas gerenciais da qualidade: implementando a melhoria continua com maior eficácia**. São Paulo: Makron Books, 1994.

NAUR, P., **Software Engineering** - Report on a conference sponsored by the NATO Science Committee, Germany, 7-11 October 2006.

O'TOOLE, P., **CMMI Process**. Improvement Yahoo Group. Disponível em: <http://groups.yahoo.com/group/cmmi_process>. Publicado em: 05 de janeiro de 2005.

OLIVEIRA, Junior F.N., **Desdobramento da Função Qualidade: O Pós-Venda Agregando Valor ao Produto em um Segmento de Mercado**. Mestrado em Engenharia Mecânica. Taubaté – SP. Universidade de Taubaté, 2006.

PANDE, P.S., NEUMAN, R.P. e CAVANAGH, R.R., **Estratégia seis sigma: como a GE e a Motorola e outras grandes empresas estão aguçando seu desempenho**. Rio de Janeiro: Qualitymark, 2001.

PÁDUA, W., FILHO, P., **Engenharia de software; fundamentos, métodos e padrões**. Rio de Janeiro: Editora LTC, 2003.

PEREIRA, E., **Um modelo de medição para processos de desenvolvimento de software**. Dissertação de Mestrado. Universidade Federal de Minas Gerais, 2001.

POLLICE, G., **Teaching Software Development and Software Engineering**. Boston – EUA: The Rational Edge Press, 2005.

RIBEIRO, F.M., **Novos processos logísticos decorrentes de profundas transformações físicas funcionais**. Mestrado em Administração – Escola Brasileira de Administração Pública – Centro de Formação Acadêmica e Pesquisa – FGV, Rio de Janeiro, 2000.

RIBEIRO, J.L.D., **Desdobramento da qualidade: um plano de melhorias para retenção de clientes em clubes sociais e esportivo**. In: XXI Encontro Nacional de Engenharia de Produção, 2001, Salvador. Anais do XXI ENEGEP.

RICCI, R., **Guia Prático ISO 9000: tudo o que saber sobre ISO 9000**. Rio de Janeiro: Editora Qualitymark, 1995.

ROCHA, A., **Qualidade de Software: Teoria e Prática**. Prentice Hall, 303 páginas, 2001.

SALVIANO, C. F., **Um Método para Escolha dos Processos para uma Melhoria Alinhada aos Objetivos de Negócio**. Anais do WQS'2001 Workshop de Qualidade de Software do SBES'2001, p. 39-50, Rio de Janeiro, RJ, Outubro, 2001.

SLACK, N., STUART, C., ROBERT, J., **Administração da Produção 2º Ed.**. São Paulo: Atlas, 2002.

SPINOLA, M. M., **Aplicação de QFD para a Especificação de um Sistema de Informações**. In: 2 Congresso Brasileiro de Gestão de Desenvolvimento de Produto. São Carlos, 2000.

VIVEIROS, S. P., **Um estudo para a utilização do método QFD na definição de medidas de qualidade de produtos de software**. Dissertação de mestrado apresentada ao Departamento de Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, MG, 2006.

ZULTNER, R.E., **Software Quality Function Deployment. Applying QFD to Software**. Transactions from the 2nd Symposium on Quality Function Deployment, QFD, Deutschland, 2002.

8 ANEXOS

8.1 Anexo 1 - ABNT (2003) - Medidas internas da norma

8.1.1 Response Time

Nome	Propósito	Método de Aplicação	Fórmula	Interpretação do valor	Tipo de Escala	Tipos	Entradas	Audiência Alvo
Response Time (Eficiência)	Qual o tempo estimado para completar uma tarefa específica?	<p>Avaliar a eficiência das chamadas do sistema operacional e da aplicação.</p> <p>Estimar o tempo de resposta baseada nisso.</p> <p>O que deve ser mensurado:</p> <p>1- toda ou parte das especificações</p> <p>2- testar caminhos de transação completos</p> <p>3- - testar módulos completos/partes do produto de software</p> <p>3- o produto completo durante a fase de testes</p>	X=tempo (calculado ou simulado)	Quanto menor melhor.	Razão	X=tempo	<p>Sistema operacional conhecido</p> <p>Tempo estimado nas chamadas do sistema</p>	Desenvolvedor Analista de requisito

8.1.2 Functional Adequacy

Nome	Propósito	Método de Aplicação	Fórmula	Interpretação do valor	Tipo de Escala	Tipos	Entradas	Audiência Alvo
Functional Adequacy (Funcionalidade)	Quão adequadas são as funções verificadas?	<p>Contar o número de funções implementadas que são adequadas para desempenhar as tarefas especificadas, para medir a sua razão para as funções implementadas.</p> <p>O que deve ser mensurado:</p> <p>1- toda ou parte das especificações</p> <p>2- módulos completos/partes do produto de software</p>	$X=1-A/B$ <p>A = Número de funções em avaliação nas quais os problemas são detectados</p> <p>B = Número de funções verificadas</p>	$0 \leq X \leq 1$ - Quanto mais próximo de 1, mais adequado	Absoluta	X=contador/contador A=contador B=contador	Especificação de requisitos Código fonte Relatório de revisão	Desenvolvedor Analista de requisito

8.1.3 Fault Removal

Nome	Propósito	Método de Aplicação	Fórmula	Interpretação do valor	Tipo de Escala	Tipos	Entradas	Audiência Alvo
Fault Removal (Confiabilidade)	Quantas falhas foram corrigidas?	Contar o número das falhas removidas durante a codificação e compará-la com o número de falhas detectadas a revisão da codificação.	$X=A$ A=Número de falhas corrigidas durante a codificação.	$0 \leq X$ Um alto valor de X implica que menos falhas permanecem	Razão	X=contador A=contador	Valor A é proveniente e do relatório de remoção de falhas.	Desenvolvedor Analista de requisito
	Qual é a proporção de falhas removidas?		$Y=A/B$ A=Número de falhas corrigidas durante a codificação B= Número de falhas	$0 \leq Y \leq 1$ Quanto mais próximo de 1, melhor. (mais falhas removidas)	Absoluta	Y=contador/ contador B=contador	Valor B é proveniente e do relatório	

8.1.4 Input Validity Checking

Nome	Propósito	Método de Aplicação	Fórmula	Interpretação do valor	Tipo de Escala	Tipos	Entradas	Audiência Alvo
Input Validity Checking (Usabilidade)	Qual proporção de campos de entrada fornece checagem para dados válidos?	Contar o número de campos de entrada, o qual checa dado válido e compara com número de campos de entrada que poderia checar dado válido	$X=A/B$ A=Número de campos de entrada com checagem de dado válido B=Número de campos de entrada os quais poderia ter checagem de dado válido	$0 \leq X \leq 1$ Quanto mais próximo de 1, melhor.	Absoluta	X=contador/ contador A=contador B=contador	Especificação de requisitos Relatório de revisão	Desenvolvedor Analista de requisito

8.1.5 User Operation Cancellability

Nome	Propósito	Método de Aplicação	Fórmula	Interpretação do valor	Tipo de Escala	Tipos	Entradas	Audiência Alvo
User Operation Cancellability (Usabilidade)	Qual proporção de funções pode ser cancelada antes de completar?	Contador o número de funções implementadas, que podem ser canceladas pelo usuário antes de completar e compará-lo com o número de funções que requerem a capacidade de serem canceladas.	$X=A/B$ A=Número de funções implementadas que podem ser canceladas pelo usuário. B=Número de funções que requerem a capacidade de serem canceladas	$0 \leq X \leq 1$ Quanto mais próximo de 1, melhor.	Absoluta	X=contador/ contador A=contador B=contador	Especificação de requisitos Relatório de revisão	Desenvolvedor Analista de requisito

8.2 Anexo 2 - Padrão IEEE-982 (1988) - Medidas internas do padrão

8.2.1 Cumulative Failure Profile

Nome	Propósito	Método de Aplicação	Fórmula	Interpretação do valor	Tipo de Escala	Tipos	Entradas	Audiência Alvo
Cumulative Failure Profile	<p>Prever a confiabilidade através do uso de perfis de falhas.</p> <p>Identificar módulos ou subsistemas que requerem testes adicionais</p>	Estabelecer os níveis de severidade a designação de falhas.	<p>Gerar gráfico de falhas acumuladas versus a base de tempo adequada.</p> <p>A curva pode ser gerada para o sistema como um todo, subsistemas ou módulos.</p>	O valor de $F(i)$ deve ser interpretado conforme base histórica da organização e nível de confiabilidade desejado.	Absoluta	$F(i) =$ Contador	$F(i) =$ número de falhas para certa severidade para um dado intervalo de tempo, $i=1, \dots$	Gerente Desenvolvedor Testador

8.2.2 Requirements Traceability

Nome	Propósito	Método de Aplicação	Fórmula	Interpretação do valor	Tipo de Escala	Tipos	Entradas	Audiência Alvo
Requirements Traceability	Auxilia a identificação de requisitos que estão faltando, ou além, dos requisitos originais.	Um conjunto de mapeamento dos requisitos na arquitetura do software para os requisitos originais é criado. Contar cada requisito mapeado pela arquitetura (R1) e contar cada um dos requisitos originais (R2)	$TM = \frac{R1}{R2} \times 100\%$	$0\% \leq TM \leq 100\%$ Quanto mais próximo de 100% melhor.	Absoluta	R1 = Contador R2 = Contador TM = Contador/Contador (%)	R1 = Número de requisitos mapeados Pela arquitetura R2 = Número de requisitos originais	Gerente Desenvolvedor

8.3 Anexo 3 – Metodologia Praxis - Medidas internas do padrão

8.3.1 Defeitos de Testes de Aceitação

Nome	Propósito	Método de Aplicação	Fórmula	Interpretação do valor	Tipo de Escala	Tipos	Entradas	Audiência Alvo
Defeitos de Testes de Aceitação	Validar os requisitos do produto	Essa medida é coletada na atividade de testes de aceitação	$F = \text{número de defeitos encontrados}$	Quanto menor melhor	Absoluta	$F = \text{Contador}$	Defeitos registrados no RTSw	Auditor de qualidade Gerente do projeto

8.3.2 Defeitos de Avaliação pelo Usuário

Nome	Propósito	Método de Aplicação	Fórmula	Interpretação do valor	Tipo de Escala	Tipos	Entradas	Audiência Alvo
Defeitos de Avaliação pelo Usuário	Avaliar o uso do produto	Essa medida é coletada na Atividade de avaliação de uso.	$F = \text{número de defeitos encontrados}$	Quanto menor melhor	Absoluta	$F = \text{Contador}$	Defeitos registrados no RAU..	Auditor de qualidade Gerente do projeto

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)