

UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC
CURSO DE CIÊNCIA DA COMPUTAÇÃO

JULIANA BROGNOLI DAMIANI

AVALIAÇÃO DE PRODUTO DE SOFTWARE PARA GESTÃO EMPRESARIAL
COM BASE NAS NORMAS DE QUALIDADE

CRICIÚMA, DEZEMBRO DE 2010

JULIANA BROGNOLI DAMIANI

**AVALIAÇÃO DE PRODUTO DE SOFTWARE PARA GESTÃO EMPRESARIAL
COM BASE NAS NORMAS DE QUALIDADE**

Trabalho de Conclusão de Curso apresentado
para obtenção do Grau de Bacharel em Ciência da
Computação da Universidade do Extremo Sul
Catarinense.

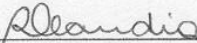
Orientadora: Profa. MSc. Ana Cláudia Garcia
Barbosa

CRICIÚMA, DEZEMBRO DE 2010

JULIANA BROGNOLI DAMIANI


Avaliação de Produto de Software para Gestão Empresarial com Base nas Normas de Qualidade

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

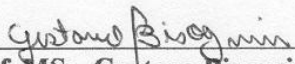


Profa. MSc. Ana Cláudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação


Banca Examinadora:



Prof. MSc. Ana Cláudia Garcia Barbosa (UNESC)
Orientadora



Prof. MSc. Gustavo Bisognin (UNESC)



Prof. MSc. Paracelso de Oliveira Caldas (UNESC)

Dedico esta pesquisa aos meus pais, Neri e Elizete, que sempre estiveram ao meu lado, apoiando e incentivando a não desistir dos meus objetivos.

AGRADECIMENTOS

Agradeço a Deus por ter me iluminado e me dado forças para percorrer este caminho e conseguir concluir este trabalho.

Aos meus pais que sempre acreditaram em mim e me ajudaram a superar todos os obstáculos que apareceram ao longo dessa caminhada.

A minha orientadora Ana Cláudia por todo apoio e dedicação, por estar sempre disposta a me ajudar.

As minhas amigas Giane, Giulia e Tamires que mesmo longe, estavam torcendo por mim e sempre me estenderam a mão quando precisei de um ombro amigo.

A minha amiga Tamara pelo companheirismo e pela paciência por me agüentar diariamente falando do TCC.

Ao casal de amigos, Sandra e Marco Aurélio pelo apoio e por me proporcionarem momentos de distração e de muita alegria nos finais de semana, depois de horas estudando.

A minha madrinha Nadir e minha prima Cristina por conseguirem um material muito importante para a realização deste trabalho e se empenharem para que chegasse até mim o mais rápido possível.

Aos meus colegas de curso e amigos pelo companheirismo e pelos momentos de descontração, pois sem eles esta conquista não seria possível.

E a todos aqueles que acreditaram e torceram por mim.

"Você não sabe o quanto eu caminhei pra chegar até aqui. Percorri milhas e milhas antes de dormir. Eu nem cochilei. Os mais belos montes escalei. Nas noites escuras de frio chorei. A vida ensina e o tempo traz o tom (...)"

A estrada - Cidade Negra

RESUMO

O uso de produtos de software está cada vez mais comum, como por exemplo, sistemas para gestão empresarial que são utilizados pelas empresas comerciais para controlar suas informações. O número de empresas que oferecem esses software crescem consideravelmente e para permanecer no mercado, o diferencial é a qualidade do produto de software desenvolvido. Porém, muitos dos produtos de software são construídos em não conformidade com as práticas de Engenharia de Software, o que afeta consideravelmente na sua qualidade. Contudo, existem muitos obstáculos que precisam ser superados para que se possa oferecer aos clientes produtos confiáveis, eficientes, funcionais e etc. Para avaliar as características de qualidade do software, existem as normas de qualidade para produto de software. Neste trabalho foi realizado uma avaliação de qualidade com um produto de software desenvolvido sem o uso de padrões e técnicas de Engenharia de Software. Na realização desta avaliação verificou-se que a ausência das práticas de Engenharia de Software durante o ciclo de vida do software afeta consideravelmente a qualidade do produto final e que mesmo com poucos recursos é possível melhorar esta situação.

Palavras-chaves: Engenharia de Software; Qualidade de Produto de Software; Avaliação de Produto de Software; Normas de Qualidade de Software.

ABSTRACT

The use of software products is increasingly getting common, such as business management systems that are used by commercial companies to control their information. The number of companies that offer such software is growing considerably and, in order to remain on the market, the differential is the quality of the software product developed. However, many software products are not built in accordance with the practices of Software Engineering, which greatly affects its quality. Nevertheless, there are many hurdles to overcome in order to offer customers products that are reliable, efficient, functional and so on. As means to assess the software quality, there are quality standards for software product. In this work, it was made a quality evaluation with a software product developed without the use of patterns and techniques of Software Engineering. In conducting this assessment it was found that the absence of the practices of Software Engineering during the life cycle of the software considerably affects the quality of the final product as well as even with few resources it is possible to improve this situation.

Keywords: Software Engineering; Software Product Quality; Software Product Evaluation; Software Quality Standards.

LISTA DE ILUSTRAÇÕES

Figura 1. Engenharia de Software em Camadas	19
Figura 2. Composição da Qualidade em Software	30
Figura 3. Representação da Norma ISO/IEC 9126 e suas Representações	47
Figura 4. Formulário para Avaliação Externa da Manutenibilidade	56
Figura 5. Processo de Avaliação da Manutenibilidade	57
Figura 6. Características de Qualidade em Uso	60
Figura 7. Qualidade no Ciclo de Vida	61
Figura 8. Representação da Norma ISO/IEC 14598	63
Figura 9. Processo de Avaliação	72

LISTA DE TABELAS

Tabela 1. Modelo ISO/IEC 9126	49
Tabela 2. Métricas da Manutenibilidade	54
Tabela 3. Partes das Normas ISO/IEC 9126 e 14598	79
Tabela 4. Avaliação do questionário	99

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
ERP	Planejamento de Recursos Empresariais
ES	Engenharia de Software
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
MA	Módulo de Avaliação
MA-MPS	Método de Avaliação
MPS.BR	Melhoria de Processo do Software Brasileiro
MR-MPS	Modelo de Referência
NBR	Norma Brasileira
RH	Recursos Humanos
SPICE	Process Improvement and Capability dEtermination
SQuaRE	Requisitos de Qualidade e Avaliação de Produtos de Software

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVO GERAL	15
1.2 OBJETIVOS ESPECÍFICOS	16
1.3 JUSTIFICATIVA	16
1.4 ESTRUTURA DO TRABALHO	18
2 ENGENHARIA DE SOFTWARE.....	20
2.1 CONTROLE DE VERSÃO.....	22
2.2 DOCUMENTAÇÃO DE SOFTWARE.....	24
2.3 ENGENHARIA REVERSA.....	25
2.4 PROCESSOS DE SOFTWARE.....	27
2.5 PRODUTO DE SOFTWARE.....	28
3 QUALIDADE DE PROCESSO DE SOFTWARE.....	32
3.1 NORMAS DE QUALIDADE PARA PROCESSO DE SOFTWARE.....	33
3.1.1 ISO/IEC 12207	34
3.1.2 CMM.....	34
3.1.3 CMMI.....	36
3.1.4 SPICE	36
3.1.5 Família ISO/IEC 9000	36
3.1.6 MPS.BR.....	37
4 QUALIDADE DE PRODUTO DE SOFTWARE.....	39
4.1 CARACTERÍSTICAS QUE DETERMINAM A QUALIDADE DE UM PRODUTO DE SOFTWARE.....	41
5 NORMAS DE QUALIDADE PARA PRODUTO DE SOFTWARE	44
5.1 ISO 9126.....	46

5.2	MODELO DE QUALIDADE INTERNA E EXTERNA	50
5.3	QUALIDADE EM USO.....	61
5.4	ISO/IEC 14598.....	64
5.4.1	ABNT NBR ISO/IEC 14598-1.....	65
5.4.2	ABNT NBR ISO/IEC 14598-2.....	65
5.4.3	ABNT NBR ISO/IEC 14598-3.....	68
5.4.4	ABNT NBR ISO/IEC 14598-4.....	71
5.4.5	ABNT NBR ISO/IEC 14598-5	72
5.4.6	ABNT NBR ISO/IEC 14598-6.....	78
5.5	ISO/IEC 25000.....	80
5.6	ISO/IEC 12119	83
6	SISTEMA PARA GESTÃO EMPRESARIAL	86
6.1	FUNCIONALIDADES DE UM SISTEMA DE GESTÃO EMPRESARIAL.....	87
7	TRABALHOS CORRELATOS	89
7.1	ELEMENTOS INTRÍNSECOS DO SOFTWARE E SUA INFLUÊNCIA NA QUALIDADE DO PROCESSO DE DESENVOLVIMENTO.....	89
7.2	NORMA ISO/IEC 9126: AVALIAÇÃO DE QUALIDADE DE PRODUTOS DE SOFTWARE.....	90
7.3	UMA PROPOSTA PARA QUALIDADE DE SOFTWARE POR MEIO DA APLICAÇÃO INTEGRADA DO MODELO CMM E DA NORMA NBR 13596	90
7.4	PROPOSTA DE METODOLOGIA PARA CONTROLE DE QUALIDADE EM UMA FÁBRICA DE SOFTWARE.....	91
8	AVALIAÇÃO DE PRODUTO DE SOFTWARE PARA GESTÃO EMPRESARIAL	92
8.1	PRÁTICAS DE ENGENHARIA DE SOFTWARE.....	93
8.2	OS CLIENTES E A QUALIDADE DO PRODUTO DE SOFTWARE	94

8.3 ESTUDO DE CASO.....	95
8.4 APLICAÇÃO DAS NORMAS DE QUALIDADE.....	96
8.5 ANÁLISE DO SOFTWARE CONFORME AS NORMAS DE QUALIDADE	97
8.6 MÓDULO DE AVALIAÇÃO EM CONFORMIDADE COM A NBR ISO/IEC 14598-6	99
8.7 RELATÓRIO DE AVALIAÇÃO.....	101
8.7.1 Recomendações de melhorias	103
BIBLIOGRAFIA	108
BIBLIOGRAFIA COMPLEMENTAR.....	112
APÊNDICE A – QUESTIONÁRIO PARA AVALIAÇÃO DO SOFTWARE	115
APÊNDICE B – RELAÇÃO DAS FALHAS ENCONTRADAS DURANTE O TESTE CAIXA-PRETA	122
APÊNDICE C – ARTIGO	124

1 INTRODUÇÃO

A Engenharia de software surgiu com o intuito de melhorar a qualidade dos produtos de software e aumentar a produtividade no seu processo de desenvolvimento.

Para que um produto final apresente bons resultados é necessário que a qualidade seja incorporada ao longo de seu desenvolvimento. A Engenharia de Software estabelece processos, métodos, técnicas, ferramentas, ambientes de suporte ao desenvolvimento e normas de qualidade para processo e produto de software (FALBO, 2005).

A melhoria da qualidade de software é dividida em melhoria da qualidade de processo e de produto.

As normas para obtenção da qualidade de processo de software, as quais estudam os requisitos necessários ao cliente, criam um ciclo de vida para os processos e realizam instalação e manutenção do software. Já as normas para qualidade de produto de software, definem as características que classificam um produto de qualidade, a maneira de medi-las e as descrições que permitem avaliar o produto.

A preocupação com a qualidade do produto apresentado ao cliente deve começar no processo de software, porém, empresas que não desenvolvem seus produtos com esses critérios, podem sofrer as conseqüências quando o produto estiver em operação. O produto de software pode ser bom, mas não atender as necessidades dos clientes, neste caso há demora em atender os requisitos pós desenvolvimento.

Um software de qualidade na visão do usuário é aquele fácil de usar, eficiente e confiável. Para um desenvolvedor o software precisa ser fácil de manter, enquanto que para o cliente o software deve agregar valor ao seu negócio, ou seja, qualidade em uso (FALBO, 2005).

Com o propósito de avaliar a qualidade dos softwares foram criadas normas de qualidade para produto de software, como a ISO/IEC 14598, ISO/IEC 12119 e ISO/IEC 9126.

Empresas utilizam sistemas de gestão para facilitar na administração e organização de suas atividades, lucros e investimentos.

Os sistemas permitem a integração entre os processos e dão suporte à maioria das operações em uma empresa, auxiliam as atividades empresariais que vão da logística de entrada até logística de saída e parte das atividades de marketing e vendas.

Neste projeto propõem-se avaliar a qualidade de um sistema para gestão empresarial. O objetivo é minimizar os erros do software entregue aos clientes, por meio da utilização das normas de qualidade para produto de software.

Diante das reclamações efetuadas constantemente pelos clientes, relativas a erros ou alterações que não satisfazem suas necessidades, resolveu-se buscar meios para minimizar os problemas e conquistar a satisfação dos clientes.

Os erros relatados serão analisados e um estudo de caso será realizado. Pretende-se realizar uma avaliação de qualidade profissional do software, de acordo com os critérios apresentados nas normas de qualidade.

1.1 OBJETIVO GERAL

Avaliar um sistema de Gestão Empresarial, com base na utilização de normas de qualidade para produto de software.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos são:

- a) compreender softwares de gestão empresarial;
- b) realizar um estudo de caso do software a ser avaliado;
- c) avaliar o software a partir das normas de qualidade para produto de software;
- d) propor as melhorias necessárias para atingir a qualidade do produto.

1.3 JUSTIFICATIVA

Os sistemas estão presentes nas empresas, facilitando a gestão empresarial. Seja na parte financeira, comercial, produção, contábil ou RH, um sistema proporciona aos usuários facilidades no desempenho de suas funções e a qualidade é fator fundamental na hora de escolher qual software utilizar.

Para atingir as expectativas dos usuários as empresas desenvolvedoras de software devem oferecer produtos de qualidade, e para tornar isso uma realidade, existem as normas que permitem fazer uma avaliação de qualidade dos produtos de software.

Qualidade é o conjunto de atributos que tornam um bem ou serviço, adequado ao uso para o qual foi desenvolvido, atendendo as necessidades dos usuários. Tais necessidades abrangem critérios como operabilidade, segurança, tolerância a falhas, conforto, durabilidade, facilidade de manutenção entre outros (ALBUQUERQUE, 2005).

Normas como a ISO/IEC 9126, ISO/IEC 12119, ISO/IEC 14598, ISO/IEC 12207 e MPS.BR possibilitam a avaliação e melhoria da qualidade de softwares.

A NBR ISO/IEC 9126 traz por exemplo, a funcionalidade como atributo de qualidade e a subdivide em adequação, acurácia, interoperabilidade, segurança de acesso e

conformidade. As características podem ser medidas por meio de métricas externas e internas e representam os instrumentos necessários para realizar uma avaliação, descrevendo amplamente como medir qualitativamente e quantitativamente a presença de qualidade, formando assim um modelo de qualidade para produto de software.

Produtos de software que estão disponíveis no mercado, são avaliados pela ISO/IEC 12119, estabelecendo os requisitos de qualidade e teste dos Pacotes de Software. O pacote a ser testado deve possuir: descrição do produto, manual do usuário e programa e dados.

A NBR ISO/IEC 14598 possui um conjunto de guias para orientar e planejar o processo de avaliação de um produto de software. Tal conjunto é formado por visão geral, planejamento e gerenciamento, processo para desenvolvedores, processo para adquirentes, processo para avaliadores e módulo de avaliação.

Na ISO/IEC 12207 tem-se como objetivo ajudar na definição dos papéis dos desenvolvedores de software, de forma a obter uma visão aperfeiçoada para o usuário das operações realizadas. Essa norma propõe processos de ciclo de vida do software organizados em três classes: processos fundamentais, processos de apoio e processos organizacionais.

A MPS.BR no nível de maturidade D tem o propósito de estabelecer os requisitos dos componentes do produto e do cliente, compor os componentes do produto, produzindo um produto integrado consistente com o projeto (design), e demonstrar que os requisitos funcionais e não-funcionais são satisfeitos para o ambiente alvo ou equivalente.

Para proporcionar às empresas maior operabilidade são desenvolvidos sistemas que auxiliam a gestão empresarial. E para possibilitar uma análise detalhada de um sistema faz-se um estudo de caso que proporciona conhecimento específico sobre o objeto de estudo.

Na empresa de desenvolvimento de software estudada, reclamações dos clientes são freqüentes, seja quando o resultado não foi o esperado ou quando o sistema apresenta erros.

Esta pesquisa propõe avaliar o produto de software, buscando por melhorias de qualidade e conseqüentemente conquistar a satisfação dos clientes. Será realizado um estudo de caso, para descobrir a causa da insatisfação dos clientes e encontrar as possíveis soluções para que determinados problemas não ocorram novamente.

1.4 ESTRUTURA DO TRABALHO

O capítulo 1 apresenta uma introdução do problema proposto, tendo como foco a Qualidade do Produto de Software, mostrando-se a necessidade de avaliar um produto de software e a importância de se utilizar normas na avaliação.

Os conceitos da Engenharia de Software, Processo e Produto de Software são abordados no capítulo 2, mostrando a relação que há entre eles, como e onde esses conceitos podem ser aplicados e como eles podem auxiliar na solução do problema.

No capítulo 3 faz-se uma abordagem sobre qualidade de processo de software, com breve apresentação de algumas das normas que auxiliam no desenvolvimento e na obtenção da qualidade destes processos.

O capítulo 4 descreve Qualidade de Produto de Software e a necessidade de se avaliar o produto final.

As normas de qualidade que auxiliam e permitem a avaliação de produto de software são apresentadas no capítulo 5.

Faz-se um breve relato sobre Sistema para Gestão Empresarial e a importância destes no mercado no capítulo 6.

No capítulo 7 são apresentados os trabalhos correlatos que serviram de base para a realização deste trabalho.

O capítulo 8 descreve a realização da avaliação de um produto de software para gestão empresarial.

2 ENGENHARIA DE SOFTWARE

Nas mais diversas áreas do conhecimento humano, os computadores são utilizados para facilitar o manuseio e armazenamento de informações, logo, aumenta a necessidade de ter-se um software capaz de auxiliar na tomada de decisão.

Os softwares para gestão empresarial, precisam apresentar soluções para problemas com certa complexidade, pois devem conter controle comercial, financeiro, estoque, etc., todos interligados em um único sistema, compartilhando informações e oferecendo resultados confiáveis. Para solucionar esses problemas e amenizar a complexidade utiliza-se as técnicas da Engenharia de Software (ES).

A ES surgiu com a necessidade de organização e padronização no desenvolvimento de software. Juntamente com ela surgiram os padrões de qualidade e o conceito de qualidade de software. Essa tecnologia busca a otimização na construção do software, a facilidade no desenvolvimento e permite oferecer um produto final acessível e de baixo custo (SODRÉ, 2006).

Área que engloba todos os aspectos da produção de software, desde descrição à manutenção. Com a finalidade de obter softwares econômicos, confiáveis e que trabalhem eficientemente em máquinas reais, a ES foca na qualidade dos produtos de software por meio de processos gerenciais, ferramentas e atividades que formam o conjunto de práticas que possibilitam o desenvolvimento de software.

A ES é uma tecnologia em camadas onde, a base de apoio é o foco na qualidade. A camada de processo faz a união entre as camadas de tecnologia e permite o desenvolvimento de software, conforme Figura 1 (PRESSMAN, 2006).



Figura 1. Engenharia de Software em Camadas
Fonte: Adaptado de PRESSMAN, R (2006 p. 17)

Vários problemas são encontrados no desenvolvimento de software e existem algumas possíveis causas para tal: estimativas de prazo e custos imprecisas, produtividade baixa das pessoas da área, qualidade do produto final inadequada.

Tais problemas são ocasionados pela falta de planejamento, coleta de dados insuficiente, falta de levantamento objetivo das necessidades e exigências dos clientes, entre outros motivos, que resultam em produtos de baixa qualidade e clientes insatisfeitos.

A ES auxilia na melhoria da qualidade, possibilitando como resultado um produto confiável e eficiente. Permite a escolha do método mais apropriado para um conjunto de circunstâncias, e da abordagem mais eficaz que os engenheiros de software devem adotar para produzir software de qualidade (SOMMERVILLE, 2003).

Os softwares servem de base para uma variedade de indústrias de diferentes ramos de atividades. Esses produtos são procurados para auxiliar na administração dos negócios, atuando na tomada de decisão.

Os sistemas de informação são usados no processo evolutivo das organizações, fazendo do conhecimento um recurso estratégico.

A tecnologia da informação é aplicada nas diversas indústrias, e portanto, está vinculada ao processo desencadeado de melhoria na qualidade dos produtos e serviços, mas primeiramente é preciso saber aonde se quer chegar, para que a qualidade possa ser alcançada.

Na tentativa de reduzir a necessidade de inventar soluções, há décadas metodologias, tecnologias e ferramentas estão sendo desenvolvidas, testadas e adaptadas para permitir que as tarefas possam ser automatizadas, diminuindo o trabalho das pessoas e garantindo uniformidade (KOSCIANSKI; SOARES, 2006).

As visões de qualidade mudam constantemente durante o ciclo de vida do software. Os requisitos de qualidade definidos no início do ciclo representam a visão do usuário e definem qualidade do produto, enquanto que a qualidade do projeto refere-se à visão dos desenvolvedores. Portanto, as tecnologias que determinam o nível de qualidade necessário, devem apoiar esses diversos pontos de vista (FERRARI, 2007).

É de extrema importância que a qualidade seja buscada ao longo de todo o ciclo de vida do software. Porém muitas vezes as empresas desenvolvedoras só percebem essa necessidade, quando o software já está em uso e apresentando falhas frequentemente.

Algumas tecnologias podem ser usadas para reverter o quadro e alcançar a qualidade do software, como por exemplo, ferramentas para controle de versão.

2.1 CONTROLE DE VERSÃO

Dias (2009) afirma que um controle de versão pode resolver problemas de desenvolvimento de software como:

- a) alguém sobrescrever o código de outra pessoa por acidente e acabar perdendo as alterações;
- b) dificuldades em saber quais as alterações efetuadas em um programa, quando foram feitas e quem fez;
- c) dificuldade em recuperar o código de uma versão anterior que está em produção;

d) problemas em manter variações do sistema ao mesmo tempo.

Um controle de versão registra a evolução do projeto, cada alteração sobre cada arquivo, com isso sabe-se quem fez o que, quando e onde e permite reconstruir uma revisão específica do arquivo sempre que desejado (DIAS, 2009).

Vários desenvolvedores podem trabalhar em paralelo sobre os mesmos arquivos sem que um sobrescreva o código do outro, evitando defeitos e perda de funcionalidade (DIAS, 2009).

O controle de versão mantém linhas de evolução do mesmo projeto, por exemplo, mantendo uma versão 1.0 enquanto a equipe prepara uma versão 2.0. (DIAS, 2009).

Existem vários softwares para controle de versão, até mesmo *open-source*, como o *Subversion* que gerencia arquivos e diretórios e as modificações feitas neles ao longo do tempo. Permite recuperar versões antigas dos dados, ou examinar o histórico de alterações (COLLINS-SUSSMAN; FITZPATRICK; PILATO, 2007).

Cada programador pode usar o Subversion em seu computador, pois ele funciona em rede, permitindo que várias pessoas modifiquem e gerenciem o mesmo conjunto de dados (COLLINS-SUSSMAN; FITZPATRICK; PILATO, 2007).

No núcleo do Subversion encontra-se um repositório, que é uma central de armazenamento de dados, onde as informações são armazenadas em forma de uma árvore de arquivos (COLLINS-SUSSMAN; FITZPATRICK; PILATO, 2007).

Quando um programador lê dados de um repositório, ele normalmente vê apenas a última versão da árvore de arquivos, mas ele pode ver estados anteriores do sistema de arquivos, podendo saber o que este diretório continha há dias atrás ou, quem foi a última pessoa que alterou o arquivo, e que alterações ela fez (COLLINS-SUSSMAN; FITZPATRICK; PILATO, 2007).

O Subversion usa um modelo de *copy-modify-merge* (copiar-modificar-combinar) onde, cada usuário se conecta ao repositório do projeto e cria uma cópia de trabalho pessoal, um espelho local dos arquivos e diretórios do repositório (COLLINS-SUSSMAN; FITZPATRICK; PILATO, 2007).

Os usuários trabalham simultaneamente e independentemente, modificando suas cópias privadas, depois essas cópias são combinadas (merged) numa nova versão final.

O sistema de controle de versão ajuda com a combinação das alterações, mas cabe aos programadores verificar se as mudanças foram realizadas de forma correta e resolver as situações de conflito definindo qual das alterações vão permanecer. Porém, conflitos não são muito freqüentes quando existe uma boa comunicação entre os usuários (COLLINS-SUSSMAN; FITZPATRICK; PILATO, 2007).

Para existir uma boa comunicação entre a equipe de desenvolvimento e para que todos conheçam bem o produto que estão desenvolvendo, é de fundamental importância que o software tenha documentação.

2.2 DOCUMENTAÇÃO DE SOFTWARE

Um software que não tem documentação torna-se dependente do programador. Se um dia aquele programador que conhece o software sair da empresa é muito difícil para os demais darem continuidade no projeto. A documentação incorreta, ou a falta dela afeta a compreensão do sistema, principalmente na atividade de análise de código (SOUZA et al, 2004).

O ciclo de vida do software envolve a manipulação de uma grande quantidade de documentação que garante que o conhecimento a respeito do projeto não é propriedade individual e não será perdido (KOSCIANSKI; SOARES, 2006).

A documentação pode influenciar na qualidade e na produtividade do software. A falta ou excesso de documentação pode colocar em risco o desenvolvimento e a manutenção do software (SOUZA et al, 2004).

A documentação é um meio de comunicação entre a equipe de desenvolvimento, contendo informações importantes para aqueles que fazem ou venham a fazer manutenção no software, auxiliando no entendimento das atividades.

Quando se trata de documentação de software, fala-se de manuais de usuário, documentação do processo e documentação do produto.

Um manual de usuário deve ser escrito/digitado numa linguagem acessível, de fácil compreensão e aprendizado, deve servir como um guia para utilização do sistema (CORRÊA, 2007).

Existem algumas técnicas, como é o caso da Engenharia Reversa, que auxiliam na documentação do software, sem que as empresas desenvolvedoras precisem dispor de muitos recursos financeiros.

2.3 ENGENHARIA REVERSA

A engenharia reversa é uma atividade que trabalha com um produto de software existente, tentando entender como este funciona, o que faz exatamente e como se comporta em todas as circunstâncias (CANHOTA JUNIOR et al, 2005).

A engenharia reversa pode ser usada quando se deseja trocar um software por outro semelhante, modificar ou entender como ele funciona, sem ter acesso a sua documentação (CANHOTA JUNIOR et al, 2005).

Enquanto que a Engenharia Reversa consiste em apenas analisar o software para criar uma representação dele, na Reengenharia analisa-se o projeto, cria-se uma representação

dele e, por meio desta, monta-se uma nova estrutura que funcione exatamente como a primeira, mas que não seja simplesmente uma cópia dela (CANHOTA JUNIOR et al, 2005).

O objetivo da engenharia reversa é criar versões de softwares já existentes com mais qualidade e melhor manutenibilidade (PRESSMAN, 2006).

Um software construído sem o uso de boas práticas de engenharia de software pode servir às necessidades dos clientes durante anos, porém, ao longo desse tempo ele sofre correções, adaptações e aperfeiçoamentos. A aplicação permanece funcionando, mas a cada modificação, efeitos colaterais inesperados e sérios ocorrem, por isso que a engenharia reversa tem sido motivada por problemas de manutenção de software (PRESSMAN, 2006).

Segundo Pressman (2006) a manutenção de software pode ser responsável por 60% de todo o esforço gasto por uma organização de desenvolvimento e a porcentagem continua a crescer à medida que mais softwares são produzidos.

Pressman (2006) afirma que existem quatro atividades que definem manutenção de software:

- a) manutenção corretiva;
- b) manutenção adaptativa;
- c) manutenção perfectiva ou de melhoria;
- d) manutenção preventiva ou reengenharia.

Quando a documentação do software precisa ser atualizada, mas têm-se poucos recursos, pode-se utilizar a abordagem *documentar quando tocar*, ou seja, em vez de documentar todo o software, são documentadas as partes que atualmente estão sofrendo modificações. Com a aplicação desta técnica ao longo do tempo, pode-se obter uma quantidade considerável de documentação útil e relevante (PRESSMAN, 2006).

A engenharia reversa é um processo de recuperação de projeto, onde as ferramentas utilizadas extraem informação do projeto de dados, arquitetural e procedimental, para um programa existente (PRESSMAN, 2006).

Processos de software determinam os passos, as atividades e os métodos que devem ser aplicados na construção de um produto de software. Esses processos devem constantemente sofrer melhorias, para que o produto final tenha qualidade.

2.4 PROCESSOS DE SOFTWARE

Processo de software é um conjunto de atividades executadas por engenheiros, cujos resultados associados geram um produto (SOMMERVILLE, 2003).

Processo é o ciclo de vida que une os passos a serem executados, atividades, métodos, práticas e transformações que conduzem os engenheiros na elaboração do software. O objetivo é favorecer a produção de sistemas de qualidade, atingindo as necessidades dos usuários finais, dentro de um cronograma e orçamento previsíveis (FALBO, 2005).

Um ciclo de vida envolve planejamento, análise e especificação de requisitos, projeto, implementação, testes, entrega e implantação, operação, manutenção (FALBO, 2005).

Quando eficiente, o processo considera relações entre atividades, elementos gerados no desenvolvimento, ferramentas, procedimentos necessários, habilidade, treinamento e motivação das pessoas envolvidas (FALBO, 2005).

A Melhoria de Processo de Software permite adaptar os processos de software conforme a necessidade da organização, propiciando gerar produtos que atendam as necessidades dos clientes. Reduz resultados indesejados, como entrega de produtos sem qualidade, trabalho desnecessário em consequência de operações e processos inadequados e,

as dificuldades encontradas nas mudanças e variações dos produtos de software, geradas pelos membros da equipe de desenvolvimento (SANT'ANA, 2002).

Para vencer a complexidade no desenvolvimento de software, é necessário o uso de linguagens de programação e ferramentas, assim como, integração entre clientes e fornecedores. A qualidade de um produto de software depende da união das pessoas, organizações e procedimentos que fazem parte do processo de criação e implantação (SANT'ANA, 2002).

Modelos de processos contêm passos do processo de desenvolvimento e uma representação abstrata da arquitetura, desenho ou definição do processo, onde, em diversos níveis de detalhe estão os elementos do processo (XAVIER, 2007).

Apesar das características em comum entre as organizações, existem fatores locais, organizacionais, procedimentos e padrões que influenciam o processo. As melhorias de processo devem ser sempre vistas como uma atividade específica de cada organização (SOMMERVILLE, 2003).

Melhoria de Processo consiste em compreender os processos existentes e modificá-los, com o objetivo de melhorar a qualidade do produto, reduzir custos, tempo de desenvolvimento e número de defeitos, nos softwares fornecidos. Uma vez que esse objetivo é alcançado, a redução dos custos e do tempo pode se tornar a principal meta da melhoria. (SOMMERVILLE, 2003).

2.5 PRODUTO DE SOFTWARE

Produto de software é o sistema que está pronto para ser vendido ou enviado ao cliente (SOMMERVILLE, 2003).

Todo produto de software é desenvolvido a partir de três fases: definição, desenvolvimento e manutenção (CAPOVILLA, 1999).

Na definição são especificadas as exigências fundamentais do sistema, atividades de análise de requisitos, análise do sistema e planejamento do projeto (CAPOVILLA, 1999).

O desenvolvimento consiste em detalhar a estrutura do sistema e seus componentes (dados, linguagens, procedimentos), tendo como atividades principais o projeto do sistema, implementação e testes (CAPOVILLA, 1999).

A manutenção é a etapa seguinte à entrega do sistema onde, são feitas as correções de erros descobertos após a entrega, adaptações exigidas pela evolução do ambiente e alterações determinadas pelo cliente, tendo como atividades a correção, adaptação e melhoria da funcionalidade (CAPOVILLA, 1999).

Na concepção de um produto o primeiro passo é definir o escopo do projeto, sendo necessário um levantamento de requisitos inicial, decomposição do problema em uma abordagem dividir para conquistar. O sistema deve ser decomposto em subsistemas que são também decompostos em módulos, que podem ainda ser decompostos em sub-módulos ou funções, até que se tenha uma visão geral das funcionalidades a serem tratadas no projeto (FALBO, 2005).

Um produto de software possui características que o diferencia de um produto manufaturado (SANT'ANA, 2002; CAPOVILLA, 1999):

- a) complexidade: um produto de software é formado por muitas regras, muitas linhas de código e idéias distintas vindas de diversos desenvolvedores. Grande número de elementos diferentes no mesmo sistema, interagindo com diversos outros dentro e fora do sistema. Dificuldades na visualização e entendimento podem levar a falhas de segurança, resultados indesejáveis. Quanto maior a complexidade, mais difícil garantir a confiabilidade;

- b) invisibilidade e intangibilidade: o usuário vê apenas o resultado da execução do software. Desenvolvedores precisam usar modelos para representação do produto, causando dificuldades de comunicação entre elementos da equipe de desenvolvimento e entre equipe e cliente;
- c) conformidade e modificabilidade: adaptações e adequações são feitas constantemente para atender as necessidades do usuário ou, quando o sistema ultrapassa o tempo de vida da tecnologia agregada e precisam atender as evoluções tecnológicas. Um produto de software está relacionado a aplicações, usuários, normas, hábitos profissionais e pessoais e tecnologias que estão em constante evolução;
- d) produção sob medida: cada cliente usa o software a sua maneira, e cada necessidade implica em um novo produto e conseqüentemente um novo projeto;
- e) não se desgasta com o uso e não tem prazo de validade;
- f) o custo final é basicamente o custo do projeto e do desenvolvimento;
- g) é o único produto que quando apresenta erro, o cliente paga para corrigir.

As falhas resultantes de projetos ou implementações e os defeitos continuam no software até que sejam percebidos devido à ocorrência de algum erro. Tais erros muitas vezes são identificados pelo usuário gerando insegurança na utilização do sistema e a correção resulta em alterações no projeto, o que pode gerar novas falhas.

Melhorias na qualidade do produto de software são necessárias para superar as falhas que muitas vezes apresentam conseqüências desastrosas, alto custo e elevado tempo de desenvolvimento e manutenção.

Para entregar ao cliente um software de qualidade não basta que ele tenha passado por vários testes e não apresente falhas. É necessário que apresente características de qualidade de acordo com as necessidades específicas dos usuários.

3 QUALIDADE DE PROCESSO DE SOFTWARE

Qualidade de software envolve tanto qualidade de processo quanto de produto, como mostra a Figura 2, portanto existem padrões organizacionais que qualificam essas duas modalidades. Para processo são os processos padrão e para produto são elementos gerados no processo de software, como modelos de documentos, roteiros e normas (FALBO, 2005).

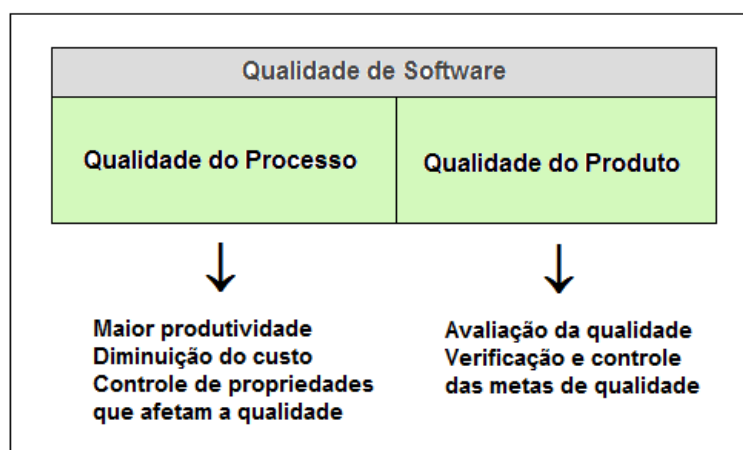


Figura 2. Composição da Qualidade em Software
Fonte: Adaptado de CAPOVILLA, I (1999 p. 26)

Na obtenção da qualidade do processo aplicam-se normas que estudam os requisitos necessários ao cliente, cria-se um ciclo de vida para os processos e realiza instalação e manutenção (SODRÉ, 2006).

Procedimentos e padrões, devem ser definidos pelos gerentes de projeto e utilizados por todos os engenheiros envolvidos no processo de desenvolvimento, para poder atingir níveis de qualidade (SOMMERVILLE, 2003).

A gestão da qualidade necessita de estrutura organizacional, responsabilidades, procedimentos, processos e recursos, por isso são desenvolvidos sistemas para ajudar as

empresas a oferecer produtos e serviços que satisfaçam as necessidades dos clientes (PRESSMAN, 2006).

A qualidade deve estar presente nos diversos estágios do desenvolvimento, fortalecendo a idéia de que as tarefas que visam qualificar um software devem ser planejadas e estruturadas antes mesmo de começar a ser desenvolvido (FERRARI, 2007).

Qualidade de software é a área de conhecimento da Engenharia de Software que busca a melhoria de software por meio da definição e normatização de processos de desenvolvimento definidos adequadamente (XAVIER, 2007).

No documento é apresentada a estruturação do projeto, envolvendo o Plano de Projeto, a Especificação de Requisitos e a Especificação de Projeto. O modelo do documento deve ser padrão para facilitar a leitura e compreensão do mesmo (FALBO, 2005).

São os padrões e critérios estabelecidos que tornam possível obter qualidade durante todo o processo de desenvolvimento, desde a concepção até a fase de manutenção.

Os padrões possibilitam capturar as melhores práticas da organização e facilitam a realização da avaliação da qualidade que podem ser dirigidas em semelhança ao padrão. Padrões gerais são propostos por instituições nacionais ou internacionais voltadas à qualidade de software, como por exemplo, as normas para avalia processo de software (FALBO, 2005).

3.1 NORMAS DE QUALIDADE PARA PROCESSO DE SOFTWARE

Existem muitas normas que avaliam processo de software e que, são mundialmente conhecidas e utilizadas pelas melhores e maiores empresas desenvolvedoras de software. Algumas das principais normas que auxiliam na obtenção da qualidade do processo de software são: ISO/IEC 12207, CMM, CMMI, SPICE, Família ISO/IEC 9000 e, a mais nova e brasileira MPS.BR.

3.1.1 ISO/IEC 12207

A ISO/IEC 12207 teve início em 1989, foi aprovada em 1995, com o objetivo de ajudar na definição dos papéis dos desenvolvedores, obtendo uma visão aperfeiçoada para o usuário. O ciclo de vida do processo é dividido em três partes (SODRÉ, 2006):

- a) processos fundamentais: atendem da contratação do sistema à manutenção do produto. O ciclo de vida é composto por processo de aquisição, processo de fornecimento, processo de desenvolvimento, processo de operação e processo de manutenção;
- b) processo de apoio: auxiliam na obtenção de qualidade de outros projetos: processo de documentação, de gerência de configuração, de garantia de qualidade, de verificação, de avaliação, de revisão conjunta, de auditoria e processo de resolução de problemas;
- c) processos organizacionais: utilizados fora do domínio do projeto normalmente para a melhoria da organização. São os processos de gerência, de infraestrutura, de melhoria e processo de treinamento.

3.1.2 CMM

O Capability Maturity Model (CMM) foi criado em 1991 para padronizar a qualidade dos softwares desenvolvidos pela força aérea americana, com base nos princípios da qualidade total e no amadurecimento gradativo do processo de desenvolvimento do software. A avaliação se dá por meio de cinco níveis de maturidade, divididos em áreas-chaves dos temas abordados (SODRÉ, 2006):

- a) nível 1 ou inicial: a obtenção de softwares de qualidade depende do desempenho dos desenvolvedores e os problemas maiores são os gerenciais pois, entram requisitos e sai o sistema pronto;
- b) nível 2 ou repetitivo: acompanhamento e documentação dos métodos de gerenciamento para assegurar o cumprimento de prazos e custos. São áreas-chaves o gerenciamento de requisitos; planejamento de projeto de software; acompanhamento de projeto de software; garantia de qualidade de software e gerenciamento da configuração de software;
- c) nível 3 ou definido: o processo de software é documentado, integrado e padronizado em processos padrão. Áreas-chaves: enfoque no processo da organização; definição do processo da organização; programa de treinamento; gerenciamento integrado de software; engenharia de produto de software; coordenação intergrupos e revisão;
- d) nível 4 ou gerenciado: processo e produto de software são avaliados quantitativamente e por medições detalhadas, o gerente tem base para tomar decisões. As áreas-chaves são: gerenciamento quantitativo do processo; gerenciamento da qualidade de software;
- e) nível 5 ou otimizado: melhoria contínua do processo por meio da realimentação quantitativa do processo, idéias e tecnologias novas. Áreas-chave são: prevenção de defeito; gerenciamento de mudanças tecnológicas; gerenciamento de mudanças no processo.

3.1.3 CMMI

Capability Maturity Model Integration (CMMI) é uma evolução do CMM, dividido em representação em estágio e contínua. Na representação em estágio é definido um conjunto de áreas de processos para melhoria e estes são distribuídos em níveis de maturidade. Já na representação contínua, uma área de processo é selecionada e a empresa melhora em relação a ela, usando níveis de capacidade para caracterizar a melhoria relacionada a essa área (SODRÉ, 2006).

3.1.4 SPICE

O Software Process Improvement and Capability dEtermination (SPICE) ou ISO/IEC 15504 avalia processos, melhorando-os continuamente e determinando sua capacidade. Formado por duas dimensões, dimensão de processo e dimensão de capacidade e definido em seis níveis de maturidade. A primeira dimensão dispõe de processos essenciais para a prática de engenharia de software e a segunda, é responsável por definir um modelo de medição para obter a capacidade de um processo de atingir seu objetivo (SODRÉ, 2006).

3.1.5 Família ISO/IEC 9000

A família ISO/IEC 9000 tem a finalidade de obter garantia de qualidade de processos, na década de 90 foram lançadas em 1987 as normas ISO 9001, 9002 e 9003. A ISO 9001 abrange o ciclo de vida de um produto e os requisitos que garantem a qualidade do produto. A ISO 9000-3 contém as orientações necessárias para aplicar a ISO 9001 em projeto,

desenvolvimento, fornecimento, instalação e manutenção de softwares. Divide-se em: estrutura, atividades do ciclo de vida e atividade de suporte (SODRÉ, 2006).

A maioria das empresas que possuem certificação CMMI ou SPICE, por exemplo, são empresas de grande porte, mas para pequenas e médias empresas a adequação a modelos de qualidade como esses é praticamente inviável, devido aos custos de implantação e certificação.

3.1.6 MPS.BR

Melhoria de Processo de Software Brasileiro (MPS.BR) foi criado com base nos modelos CMM, ISO 12207 e ISO 15504 para que as empresas nacionais pudessem amadurecer seu processo produtivamente (COLENCI NETO, 2008).

Conforme Colenci Neto (2008) este programa apresenta o modelo de referência para melhoria (MR-MPS) e para avaliação de processo (MA-MPS), uma combinação entre processos e capacidade de processos e, subdivide-se em sete níveis de maturidade:

- a) nível A (em otimização): inovação e implementação na organização, análise e resolução de causas;
- b) nível B (gerenciado quantitativamente): desempenho do processo organizacional, gerência quantitativa do projeto;
- c) nível C (definido): análise de decisão e resolução, gerência de risco;
- d) nível D (largamente definido): desenvolvimento de requisitos, solução técnica, integração, instalação, liberação, verificação, validação;
- e) nível E (parcialmente definido): treinamento, avaliação e melhoria do processo organizacional, definição do processo organizacional, adaptação do processo para gerência de projeto;

- f) nível F (gerenciado): medição, gerência de configuração, aquisição, garantia da qualidade;
- g) Nível G (parcialmente gerenciado): gerência de requisitos, gerência de projetos.

A análise começa no nível G e segue até o nível A, patamares de evolução vão sendo estabelecidos caracterizando estágios de melhoria (SOFTEX, 2009).

Melhorando a qualidade do processo de software é possível aperfeiçoar a qualidade do produto. Mas, mesmo quando os processos são bem elaborados e obedecem aos padrões de qualidade, o produto final pode apresentar algumas falhas, portanto avaliar a qualidade do produto de software é de extrema importância.

4 QUALIDADE DE PRODUTO DE SOFTWARE

A maioria das empresas que desenvolvem algum produto ou serviço, desejam atingir um alto nível de qualidade, porém quando se fala em produto de software é mais complexo apresentar definições de qualidade (SOMMERVILLE, 2003).

Um mesmo software quando analisado por pessoas diferentes pode apresentar diferentes características de qualidade e ambientes operacionais. O nível de qualidade pode ser diferente de um ambiente para outro, devido as diferentes necessidades e capacidades dos usuários e os distintos ambientes de hardware e de apoio (NBR ISO/IEC 9126-1, 2003).

Os usuários nem sempre conseguem expressar suas reais necessidades, logo, é praticamente impossível atingir a qualidade perfeita e definir completamente os requisitos de qualidade antes do início do projeto (NBR ISO/IEC 9126-1, 2003).

Atributos de software especificados por um usuário final, durante a fase de análise de requisitos, podem não atendê-lo na fase de operação, devido a mudança de requisitos e à dificuldade de especificar as necessidades implícitas (NBR ISO/IEC 9126-1, 2003).

As necessidades implícitas são aquelas que podem não ter sido especificadas, mas estão presentes em condições particulares de uso.

Para atingir a qualidade necessária e suficiente para cada contexto de uso especificado quando o produto for entregue e utilizado pelos usuários, é necessário entender detalhadamente as reais necessidades dos usuários (NBR ISO/IEC 9126-1, 2003).

Um software difícil de aprender e utilizar, mas de alta funcionalidade, também pode ser considerado um produto de qualidade (PLEEGER, 2004).

A garantia da qualidade de software assegura que os processos são estabelecidos e melhorados constantemente, satisfazendo os requisitos adequados ao uso. Conseqüentemente

reduz o número de defeitos, cria mecanismos que controlem o desenvolvimento, cumprindo prazos e custos, reduz o re-trabalho e proporciona o uso (ALBUQUERQUE, 2005).

No desenvolvimento de software o processo de qualidade é especialmente importante, devido à dificuldade de medir atributos de software sem usá-lo por um longo período. A melhoria da qualidade permite identificar os produtos de qualidade e examinar os processos utilizados no projeto (SOMMERVILLE, 2003).

A competitividade está cada vez maior, cada vez mais os clientes exigem melhor qualidade nos produtos. O diferencial tecnológico e a qualidade são pré-requisitos para vencer a competição e ganhar a preferência do cliente.

A qualidade está relacionada à definição de conformidade, as especificações e a satisfação do cliente, portanto, prazos, pontualidade de entrega e flexibilidade são fatores relevantes para a avaliação (SODRÉ, 2006).

É diferente avaliar um produto de software e avaliar o sistema no qual o software é executado. A confiabilidade de um produto de software é avaliada pela extração das falhas observadas, somente daqueles defeitos que ocorreram por causa do software, originários dos requisitos, projeto ou implementação. Já a confiabilidade de um sistema é avaliada pela observação de todas as falhas ocorridas independentemente da causa, seja hardware, software, erro humano etc. (NBR ISO/IEC 9126-1, 2003).

O número de falhas é contado durante a operação de um sistema computacional executando o programa, por isso o número de falhas encontradas durante o teste é uma medida externa do número de defeitos no programa (NBR ISO/IEC 9126-1, 2003).

Medidas externas podem ser usadas para avaliar atributos de qualidade de maneira mais próxima aos objetivos do projeto (NBR ISO/IEC 9126-1, 2003).

Uma falha é caracterizada pelo término da capacidade de um produto de executar uma função, ou a sua incapacidade de executá-la dentro de limites previamente especificados,

enquanto que o defeito é o passo, processo ou definição de dados incorretos no programa (NBR ISO/IEC 9126-1, 2003).

4.1 CARACTERÍSTICAS QUE DETERMINAM A QUALIDADE DE UM PRODUTO DE SOFTWARE

Julga-se que um produto de qualidade deve cumprir com a sua especificação, mas quando se trata de software, isso possui certa complexidade. Muitas vezes um produto que atenda à sua especificação pode não ser considerado pelo usuário como um produto de qualidade (SOMMERVILLE, 2003).

Segundo Sommerville (2003) o gerenciamento de qualidade abrange três atividades principais:

- a) garantia da qualidade: estabelecimento dos procedimentos e padrões que conduzem ao software de alta qualidade;
- b) planejamento de qualidade: seleção dos procedimentos e padrões adequados e adaptação a um projeto de software específico;
- c) controle de qualidade: definição e aprovação dos processos que assegurem que os procedimentos e padrões sejam seguidos.

A satisfação do cliente e a gestão empresarial são bases da qualidade, mas bons resultados são atingidos por meio do uso correto de metodologias (SODRÉ, 2006).

No desenvolvimento de software fatores como a influência das habilidades e experiências individuais, a novidade de uma aplicação ou a pressão comercial para liberação rápida de um produto, afetam na qualidade final (SOMMERVILLE, 2003).

Segundo Albuquerque (2005) dois conceitos importantes auxiliam na concepção qualificada e na obtenção da qualidade dos produtos de software:

- a) validação: pode ser aplicada no início ou no final do desenvolvimento e determina se o software satisfaz os requisitos especificados. Assegura que o produto atende ao uso para o qual foi proposto, garantido que foi desenvolvido corretamente;
- b) verificação: permite determinar se o produto de uma dada fase de desenvolvimento satisfaz as condições impostas no início da mesma e confirmar se o produto está de acordo com os requisitos especificados para ele, ou seja, que foi feito de maneira correta.

Esses conceitos permitem as organizações fornecer aos seus clientes, maior confiabilidade na realização das atividades desempenhadas no sistema.

Os requisitos que compõem um produto de software devem ser bem definidos e seu projeto deve ser gerenciado, para que sejam reduzidas ao mínimo as necessidades implícitas, e alcançadas as necessidades explícitas. Os defeitos devem ser reduzidos ao mínimo para conquistar a satisfação do cliente (SANT'ANA, 2002).

As necessidades implícitas, não são descritas nos documentos do desenvolvedor, mas necessárias ao usuário, como implicações estéticas, itens de segurança e outras.

As necessidades explícitas estão relacionadas com as condições em que o produto deve ser utilizado, incluindo definição dos requisitos, condições, objetivos propostos, funções e desempenho esperado.

Modelos e normas de qualidade são usados para avaliar a qualidade de um produto de software quando os objetivos de software são estabelecidos. O produto é decomposto em características e sub-características que podem ser usadas como uma lista de

verificação de questões relacionadas à qualidade. Quando o software é extenso não é possível medir todas as características e sub-características (FERRARI, 2007).

Características e sub-características devem ser avaliadas individualmente, por meio de normas de qualidade, podendo ter mais ou menos importância dependendo do propósito do software.

5 NORMAS DE QUALIDADE PARA PRODUTO DE SOFTWARE

Durante o desenvolvimento de um produto de software as melhorias de processo são usadas para avaliar a qualidade dos processos. Depois de finalizado o software pode passar por avaliações por meio de normas e modelos que determinam sua qualidade.

A ISO 9000 é um padrão internacional que pode ser usado para gerenciar a qualidade em qualquer indústria. A ISO 9001 foca nas organizações que projetam, desenvolvem e prestam manutenção a produtos (SOMMERVILLE, 2003).

A ISO 9000 expõe elementos que garantem a qualidade de forma genérica, quer podem ser aplicados a distintos produtos ou serviços. Para possuir a certificação a empresa interessada passa por verificações do atendimento das normas e da veracidade da operação (PRESSMAN, 2006).

A Série ISO 9000 apresenta várias normas que visam à qualidade, orientando organizações e estabelecendo sistemas de qualidade eficazes e eficientes, resultando na melhoria contínua dos produtos e no aumento da satisfação de clientes e colaboradores (CAPOVILLA, 1999).

A certificação ISO 9000 é um dos meios mais utilizados nas empresas de software para indicar que a organização é capaz de oferecer produtos ou serviços de qualidade (CAPOVILLA, 1999).

Segundo Pressman (2006) os requisitos da ISO 9000:2000 tratam de tópicos como:

- a) responsabilidade de gestão;
- b) sistema de qualidade;
- c) revisão de contrato;
- d) controle de projeto;

- e) controle de documento e dados;
- f) identificação de produtos e rastreabilidade;
- g) controle de processo;
- h) inspeção e teste;
- i) ação corretiva e preventiva;
- j) registros de controle de qualidade;
- k) auditorias de qualidade interna;
- l) treinamento;
- m) serviço;
- n) técnicas estatísticas.

Empresas que trabalham com desenvolvimento ou manutenção de software são avaliadas com a ISO 9001, usando as diretrizes específicas pela norma ISO 9000-3. Quando se trata de avaliação de serviços referentes a suporte, manutenção e aperfeiçoamento do produto são usadas as diretrizes da norma ISO 9004-2 (CAPOVILLA, 1999).

Empresas que prestam consultoria, treinamento e serviços de informática, não envolvendo projeto de software são avaliadas com a norma ISO 9002 em conjunto com a ISO 9004-2 (CAPOVILLA, 1999).

Como as normas ISO não se aplicam especificamente a um tipo de organização, cada uma delas deve definir um conjunto de processos de qualidade e documentar um manual de qualidade organizacional (SOMMERVILLE, 2003).

Priorizando o ganho de vantagem competitiva e o estabelecimento de metas objetivas para implantação de programas de qualidade, as empresas de software buscam a certificação ISO, que favorece a especificação de metas claras e objetiva, facilitando a avaliação e medição do progresso das atividades do sistema. (CAPOVILLA, 1999).

As empresas que não oferecerem produtos de qualidade, estarão em desvantagem. Na busca pela qualidade as empresas desenvolvedoras de software procuram bases em conceitos e diretrizes reconhecidos internacionalmente, por exemplo, as normas elaboradas e revisadas por órgãos responsáveis por normatizações técnicas, como a ISO.

A ISO é um órgão das Nações Unidas e tem o objetivo de fixar normas técnicas essenciais, de âmbito internacional, foi estabelecida em 1947 e é atualmente composta por mais de 90 países, inclusive o Brasil. O representante brasileiro na ISO é a ABNT - Associação Brasileira de Normas Técnicas, por meio do CB-25 - Comitê Brasileiro da Qualidade (SANT'ANA, 2002).

5.1 ISO 9126

A ISO/IEC 9126 foi publicada em 1991 e teve em 1996 sua tradução para o Brasil, chamada então de NBR 13596 – Tecnologia de Informação – Avaliação de produto de software e definiu seis características de qualidade e descreveu um modelo de processo para avaliação de produto de software (NBR ISO/IEC 9126-1, 2003).

As características de qualidade e as métricas associadas podem ser úteis na avaliação de produto de software, e também na definição de requisitos de qualidade e outros. Portanto, a NBR 13596 foi substituída por duas séries de normas relacionadas, a NBR ISO/IEC 9126 - Qualidade do produto de software e NBR ISO/IEC 14598 - Avaliação de produto de software (NBR ISO/IEC 9126-1, 2003).

Desenvolvida para identificar atributos de qualidade para software, a NBR ISO/IEC 9126 apresenta atributos-chave de qualidade e sub-características que determinam a capacidade do software (PRESSMAN, 2006; FERRARI 2007).

Métodos são usados para medir as sub-características, por exemplo, o nível de qualidade da característica funcionalidade é representado pela média de suas cinco sub-características e, se uma das medidas não for correspondente à funcionalidade, o valor agregado pode falhar e não representar completamente a qualidade desta característica (FERRARI, 2007).

A norma pode ser aplicada a qualquer produto de software desde que seja adaptada a um propósito específico. Algumas das sub-características apresentadas no modelo ISO 9126 são vistas em outros modelos como características, fatores ou critérios de qualidade (FERRARI, 2007).

Medir a complexidade de um sistema é de fundamental importância no controle e gestão da qualidade de software, pois a medição na maioria das vezes afeta os atributos de qualidade, portanto devem ser levadas em consideração, várias medições para esses atributos (FERRARI, 2007).

O padrão ISO define um modelo de qualidade como um conjunto de características e o relacionamento entre elas, servindo de embasamento para especificar requisitos e avaliar a qualidade.

A ISO/IEC 9126 permite apontar e avaliar produto de software a partir de diferentes aspectos como aquisição, requisitos, desenvolvimento, uso, avaliação, suporte, manutenção, garantia da qualidade e auditoria de software. Pode ser usada por desenvolvedores, adquirentes, equipe de garantia da qualidade e avaliadores independentes, particularmente aqueles responsáveis por especificar e avaliar produtos de software (NBR ISO/IEC 9126-1, 2003).

Segundo Ferrari (2007) o modelo de qualidade definido pela ISO/IEC 9126 pode ser usado para:

- a) validar a integridade de uma definição de requisitos;

- b) identificar requisitos de software;
- c) identificar objetivos de projeto de software;
- d) identificar objetivos de teste de software;
- e) identificar critérios para garantia de qualidade;
- f) identificar critérios de aceitação para um produto de software completo.

Ferrari (2007) afirma que a ISO/IEC 9126 que trata da qualidade de produto de software, em conjunto com a ISO/IEC 15504 relacionada com a avaliação do processo de software pode fornecer:

- a) *framework* para definição de qualidade de produto de software, no processo consumidor-fornecedor;
- b) suporte para revisão, verificação e validação e um *framework* para avaliação quantitativa da qualidade no processo de suporte;
- c) suporte para determinar objetivos de qualidade organizacional no processo de gerenciamento.

Em união com a ISO/IEC 12207 relacionada ao ciclo de vida do software pode fornecer: um *framework* para definição de requisitos de qualidade de produto de software; suporte para a revisão, verificação e validação no suporte aos processos do ciclo de vida (FERRARI, 2007).

Pode ser usada em conjunção com a ISO 9001 relacionada com processos de garantia da qualidade para prover: suporte para determinar objetivos de qualidade; suporte para revisão, verificação e validação de projeto (FERRARI, 2007).

De acordo com Ferrari (2007) a avaliação da qualidade de um produto de software é um processo do ciclo de desenvolvimento onde, o produto pode ser avaliado:

- a) medindo-se atributos internos, medidas estáticas de produtos intermediários;

- b) medindo-se atributos externos, medindo o comportamento do código quando executado;
- c) medindo os atributos da qualidade em uso para que o produto tenha o resultado necessário em um contexto particular de uso.

Revisões e sucessivas melhorias levaram a divisões dessa norma em ISO/IEC 9126-1, ISO/IEC 9126-2, ISO/IEC 9126-3 e ISO/IEC 9126-4, como mostra a Figura 3 (SODRÉ, 2006).

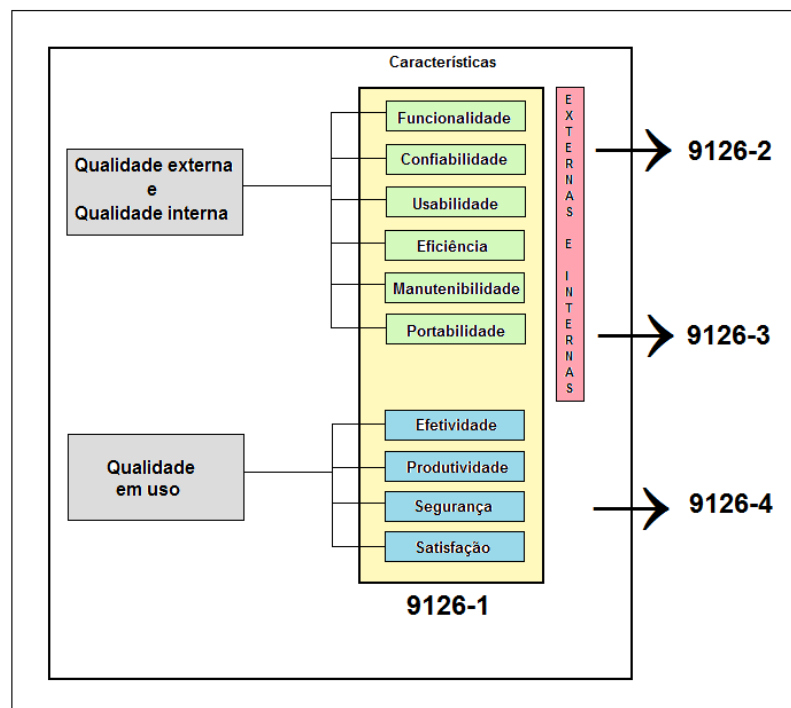


Figura 3. Representação da Norma ISO/IEC 9126 e suas Representações
 Fonte: Adaptado de SODRÉ, C, (2006, p. 27)

A ISO/IEC 9126-1 define um modelo de qualidade por meio de um conjunto de características que podem ser aplicadas a qualquer produto de software. Esse modelo de qualidade é dividido em: Modelo de Qualidade Interna e Externa e Modelo de Qualidade em Uso (SODRÉ, 2006).

A ISO/IEC 9126-2 utiliza métricas externas, medições baseadas nas necessidades do usuário para fazer a avaliação das características de qualidade externa. Para avaliar as características de qualidade interna são utilizadas as métricas internas, descritas na ISO/IEC 9126-3. A ISO/IEC 9126-4 define métricas para avaliação das características de qualidade em uso (SODRÉ, 2006).

5.2 MODELO DE QUALIDADE INTERNA E EXTERNA

A qualidade interna é composta pelas características do produto de software do ponto de vista interno e, é medida e avaliada com relação aos requisitos de qualidade interna, visando a melhoria do funcionamento e dos processos da empresa desenvolvedora.

Detalhes da qualidade do produto de software podem ser melhorados durante a implementação do código, revisão e teste, mas a natureza fundamental da qualidade do produto de software representada pela qualidade interna mantém-se inalterada, a menos que seja reprojeta (NBR ISO/IEC 9126-1, 2003).

A qualidade externa preocupa-se com a satisfação do cliente, é a totalidade das características do produto de software do ponto de vista externo, quando o software é executado. O produto é medido e avaliado enquanto está sendo testado num ambiente simulado, com dados simulados e usando métricas externas (NBR ISO/IEC 9126-1, 2003).

Durante os testes, convém que a maioria dos defeitos seja descoberta e eliminada, entretanto, alguns defeitos podem permanecer após o teste. Como é difícil corrigir a arquitetura do software ou outro aspecto básico do projeto do software, a base do projeto freqüentemente permanece inalterada ao longo do teste (NBR ISO/IEC 9126-1, 2003).

O Modelo de Qualidade Interna e Externa define os atributos de qualidade nas seis características e suas respectivas sub-características que definem a capacidade de um

produto de software: funcionalidade, confiabilidade, usabilidade, eficiência, portabilidade e manutenibilidade, como mostra a Tabela 1 (GOMES, 2000).

Tabela 1. Modelo ISO/IEC 9126

CARACTERÍSTICAS	SUB-CARACTERÍSTICAS	SIGNIFICADO
Funcionalidade O conjunto de funções satisfaz as necessidades explícitas e implícitas para a finalidade a que se destina o produto?	Adequação	Propõe a fazer o que é apropriado?
	Acurácia	Gera resultados corretos ou conforme acordados?
	Interoperabilidade	É capaz de interagir com os sistemas especificados?
	Segurança de acesso	Evita acesso não autorizado, acidental ou deliberado a programas e dados?
Confiabilidade O desempenho se mantém ao longo do tempo e em condições estabelecidas?	Conformidade à funcionalidade	Está de acordo com as normas e convenções previstas em leis e descrições similares?
	Maturidade	Com que frequência apresenta falhas?
	Tolerância a falhas	Ocorrendo falhas como ele reage?
Usabilidade É fácil de utilizar o software?	Recuperabilidade	É capaz de recuperar dados após uma falha?
	Inteligibilidade	É fácil entender os conceitos utilizados?
	Apreensibilidade	É fácil aprender a usar?
Eficiência Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto?	Operacionalidade	É fácil de operar e controlar a operação?
	Comportamento em relação ao tempo	Qual é o tempo de resposta e de processamento?
Manutenibilidade Há facilidade para correções, atualizações e alterações?	Comportamento em relação aos recursos	Quanto recurso utiliza?
	Analisabilidade	É fácil encontrar uma falha quando ocorre?
	Modificabilidade	É fácil modificar e remover defeitos?
	Estabilidade	Há grandes riscos de <i>bugs</i> quando se faz alterações?
Portabilidade É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação?	Testabilidade	É fácil testar quando se faz alterações?
	Adaptabilidade	É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no software considerado?
	Capacidade para ser instalado	É fácil instalar em outros ambientes?
	Capacidade para substituir	É fácil substituir por outro software?
	Conformidade à portabilidade	Está de acordo com padrões ou convenções de portabilidade?

Fonte: Adaptado de GOMES, N (2000, p. 4)

Funcionalidade é a capacidade de atender as necessidades explícitas e implícitas.

Grau em que o software satisfaz as necessidades reveladas. Essa característica está

relacionada com o que o software faz para atender as necessidades e tem como sub-características:

- a) adequação: capacidade de apresentar um conjunto de funções apropriadas para tarefas e objetivos específicos;
- b) acurácia: capacidade de fornecer resultados corretos com a precisão necessária;
- c) interoperabilidade: capacidade de interagir com um ou mais sistemas especificados;
- d) segurança de acesso: capacidade de proteger informações e dados, para que somente pessoas autorizadas ou sistemas, tenham acesso.
- e) conformidade à funcionalidade: capacidade de estar de acordo com normas, convenções ou regulamentações previstas em lei e prescrições similares, relacionadas à funcionalidade.

Confiabilidade é a capacidade de manter um nível especificado de desempenho quando usado em condições específicas (em software não ocorre desgaste ou envelhecimento, limitações são decorrentes de defeitos na especificação de requisitos, projeto e implementação, falhas decorrentes desses defeitos dependem de como o produto é usado e das opções de programa selecionadas e não do tempo decorrido). Período de tempo em que o software está disponível para uso. Essa característica tem como sub-características:

- a) maturidade: capacidade de evitar falhas, causadas por erros no software;
- b) tolerância a falhas: capacidade de desempenho especificado, no caso de falhas, ou violação nas interfaces;
- c) recuperabilidade: capacidade de restabelecer um nível de desempenho especificado e recuperar o dado diretamente afetado, em caso de falha (após uma falha, o produto de software poderia ficar inativo por certo período de tempo, a sua recuperabilidade é influenciada por este período de tempo);

d) conformidade à confiabilidade: capacidade do software de estar de acordo com as normas, convenções ou regulamentações relacionadas à confiabilidade.

Usabilidade é a capacidade de ser entendido, aprendido, usado e atrativo ao usuário, quando usado em condições específicas (como usuários pode-se incluir operadores, usuários finais e usuários indiretos que sejam dependentes ou estejam sob influência do uso do software). Essa característica tem como sub-características:

- a) inteligibilidade: capacidade de permitir que o usuário entenda se o software é apropriado e como pode ser usado para determinadas tarefas e condições de uso (a inteligibilidade dependerá da documentação e das impressões iniciais oferecidas pelo software);
- b) apreensibilidade: capacidade de possibilitar ao usuário o aprendizado referente à sua aplicação;
- c) operacionalidade: capacidade de possibilitar ao usuário operação e controle (aspectos de adequação, modificabilidade, adaptabilidade e capacidade para ser instalado podem afetar a operacionalidade. Operacionalidade corresponde à controlabilidade, tolerância a erros e conformidade com as expectativas do usuário);
- d) atratividade: capacidade de atrair o usuário (isto refere-se a atributos de software que possuem a intenção de tornar o software mais atraente para o usuário, como o uso de cores e da natureza do projeto gráfico);
- e) conformidade à usabilidade: capacidade do software de estar de acordo com as normas, convenções, guias de estilo ou regulamentações relacionadas à usabilidade.

Eficiência é a capacidade de fornecer desempenho apropriado, relativo à quantidade de recursos usados, em condições de uso específicas. Essa característica tem como sub-características:

- a) comportamento em relação ao tempo: capacidade de fornecer tempos de resposta e de processamento, além de taxas de transferência apropriadas, quando o software executa suas funções sob condições estabelecidas;
- b) utilização de recursos: capacidade de usar quantidades e tipos adequados de recursos quando o software executa suas funções sob condições especificadas (recursos humanos estão incluídos como parte da produtividade);
- c) conformidade à eficiência: capacidade do software de estar de acordo com as normas e convenções relacionadas à eficiência.

Portabilidade é a capacidade de transportar o software (organizacional, de hardware ou de software) de um ambiente para outro. Essa característica tem como sub-características:

- a) adaptabilidade: capacidade de adaptação a diferentes ambientes sem requerer ações ou sugerir outras não providenciadas para esse propósito (inclui a possibilidade de ajustes da capacidade interna como campos de tela, tabelas, volume de transações, formato de relógios, etc.);
- b) capacidade de ser instalado: capacidade de o software ser instalado em um ambiente especificado;
- c) co-existência: capacidade de coexistir com outro software independente, em um ambiente, compartilhando recursos;
- d) capacidade para substituir: capacidade de ser usado no lugar de outro software, com o mesmo propósito e no mesmo ambiente (numa nova versão de um produto de software, a capacidade para substituir é importante para o usuário

na atualização da versão. Capacidade para substituir pode incluir atributos de capacidade para ser instalado e adaptabilidade);

- e) conformidade à portabilidade: capacidade do software de estar de acordo com as normas e convenções relacionadas à portabilidade.

Manutenibilidade é a capacidade de modificação de correções, melhorias ou adaptação do software devido a mudanças no ambiente e nos requisitos e particularidades funcionais. Essa característica tem como sub-características:

- a) analisabilidade: capacidade de analisar as causas de falhas no software, ou a identificação das partes a serem modificadas;
- b) modificabilidade: capacidade de permitir que uma modificação especificada seja implementada (implementação inclui modificações no código, projeto e documentação);
- c) estabilidade: capacidade de evitar resultados não esperados causados por modificação no software;
- d) testabilidade: capacidade de tornar válido o software modificado;
- e) conformidade à manutenibilidade: capacidade do software de estar de acordo com normas ou convenções relacionadas à manutenibilidade.

Para medir a manutenibilidade são usadas algumas métricas, como mostra a Tabela 2.

Tabela 2. Métricas da Manutenibilidade

Atributos	Questões	Métricas
Analisabilidade		
Rastreabilidade das falhas	O usuário consegue identificar a operação específica que causou a falha?	$X = \text{falhas identificadas} / \text{falhas registradas}$
Suporte ao diagnóstico	O usuário consegue achar facilmente a causa da falha?	$X = \text{falhas diagnosticadas com funções de diagnóstico} / \text{falhas registradas}$
Capacidade de análise das falhas	O mantenedor consegue achar facilmente a causa da falha?	$X = 1 - \text{falhas sem causas identificadas} / \text{falhas registradas}$
Eficiência na análise das falhas	O usuário consegue analisar a causa de uma falha eficientemente?	$X = \text{Sum} (T \text{ acha causa} - T \text{ relat falha}) / \text{falhas registradas}$
Capacidade da monitoração do status	O usuário consegue identificar a operação causadora da falha via monitoração do status na operação?	$X = 1 - \text{casos sem informações monitoração} / \text{casos tentados}$
Modificabilidade		
Eficiência do ciclo de modificação	O problema do usuário foi solucionado satisfatoriamente para ele em tempo aceitável?	$X = \text{Sum} (T_{vm} - T_{sm}) / \text{versões}$
Tempo médio de modificação	O mantenedor consegue facilmente mudar o software para resolver o problema de falha?	$X = \text{Sum} (T_{out} - T_{in}) / \text{falhas registradas e removidas}$
Complexidade da modificação	O mantenedor consegue facilmente modificar o software para resolver problema?	$X = \text{Sum} (\text{Tempo modificação}) / \text{modificações realizadas}$
Modificabilidade parametrizada	O usuário ou mantenedor consegue facilmente mudar o software e resolver problemas via mudança de parâmetros?	$X = \text{casos resolvidos via mudança de parâmetros} / \text{tentativas via mudança de parâmetros}$
Capac. Controle de modificação	O mantenedor consegue facilmente mudar o software para resolver problemas?	$X = \text{registros efetivados no log de modificação} / \text{registros planejados para modificações}$
Estabilidade		
Razão de modificações bem sucedidas	O mantenedor consegue corrigir ou contornar as falhas causadas pelos efeitos colaterais de manutenção?	$X = (\text{falhas após modificação} / \text{Tempo após modificação}) / (\text{falhas prévias a modificação} / \text{Tempo prévio})$
Localização do impacto	Após a manutenção, o usuário consegue operar o sistema de software sem falhas?	$X = \text{falhas novas após modificação} / \text{falhas resolvidas}$
Testabilidade		
Disponibilidade de função de teste built-in	O usuário e o mantenedor conseguem facilmente realizar teste caixa-preta sem preparação adicional das facilidades para teste?	$X = \text{casos em que utiliza adequadamente a função de teste built-in} / \text{casos em que testes podem ser realizados}$
Eficiência do re-teste	O usuário e o mantenedor conseguem facilmente realizar teste caixa-preta e determinar se o software está pronto para operação ou não?	$X = \text{Sum} (\text{Tempo de teste para assegurar falha resolvida}) / \text{falhas resolvidas}$
Restartabilidade do Teste	Após manutenção, o usuário e o mantenedor conseguem facilmente realizar teste caixa-preta utilizando pontos de checagem?	$X = \text{casos em que pausa e recomeça o teste em pontos de checagem} / \text{casos identificados de pausas nos testes}$
Conformidade à manutenibilidade		
Conformidade	O quanto conforme é a manutenibilidade do produto a regulamento, normas e convenções	$X = \text{itens implementados} / \text{itens especificados de conformidade}$

Fonte: Adaptado de AGUAYO, M; GUERRA, A; COLOMBO, R, (2005, p. 4)

Os valores destas métricas são definidos para cada produto de software ou parte dele, de acordo com sua natureza, conforme categoria do software, nível de integridade e necessidades dos usuários. Na avaliação da manutenibilidade é necessária uma avaliação da versão inicial e uma avaliação da versão modificada, entre as quais se estabelece um prazo, para as devidas modificações (AGUAYO; GUERRA; COLOMBO, 2005).

Após o conhecimento da versão inicial e as devidas modificações no produto de software será possível estimar a capacidade de Manutenibilidade do produto. O tempo limite para efetuar as alterações será estabelecido de acordo com o número de modificações recomendadas, e com as exigências vigentes do mercado a que se destina o software (AGUAYO; GUERRA; COLOMBO, 2005).

Na avaliação da versão inicial, são geradas recomendações para: eliminar uma falha, inserir uma nova funcionalidade, ou mudar o ambiente (AGUAYO; GUERRA; COLOMBO, 2005).

Os desenvolvedores e os responsáveis pela avaliação são convocados a participar de uma reunião para esclarecimentos, após a avaliação, para esclarecer possíveis desentendimentos da equipe de avaliação sobre o funcionamento do produto avaliado. É possível que o desenvolvedor, mostre como executar funções corretamente, e estabelecer as modificações e prazos, que dependerão do número de modificações a serem realizadas de acordo com as exigências do mercado a que se destina o software (AGUAYO; GUERRA; COLOMBO, 2005).

Na manutenção, são realizadas as devidas modificações. Na instalação e apresentação da versão modificada, se necessário, deve ser feito um treinamento da nova versão. Deve ser instalada a versão inicial num equipamento independente, será útil para os casos em que precise verificar diferenças e semelhanças entre as versões, como falhas que aparentemente não existiam (AGUAYO; GUERRA; COLOMBO, 2005).

Na avaliação da versão modificada e avaliação da manutenibilidade, deve ser aplicado o mesmo método de avaliação que foi aplicado na versão inicial, com maior atenção e esforço para avaliação das partes modificadas. Em paralelo, deve ocorrer a avaliação da manutenibilidade, para a qual propõe-se utilizar um formulário, como mostra o Figura 4, contendo as questões básicas a serem respondidas e deve ser aplicado para cada modificação solicitada: correção de falha, melhoria ou adaptação (AGUAYO; GUERRA; COLOMBO, 2005).

<p>Descrição da falha (melhoria ou adaptação):</p> <p>1 Analisabilidade Durante a apresentação: <input type="checkbox"/> 1. Foi possível identificar a operação em que a falha (melhoria ou adaptação) acontece; <input type="checkbox"/> 2. O avaliador conseguiu identificar a causa da falha (melhoria ou adaptação); <input type="checkbox"/> 3. Foi possível identificar a causa da falha, via monitoração do status; <input type="checkbox"/> 4. Foi possível identificar a causa da falha, via utilização de funções de diagnóstico;</p> <p>2 Comunicação e Feed-back <input type="checkbox"/> 1. A descrição da falha foi enviada pelo avaliador ao mantenedor. Caso positivo, anote o momento do envio: data ___/___/___, hora: _____; <input type="checkbox"/> 2. Uma explicação satisfatória da causa da falha foi enviada. Caso positivo, anote o momento do envio: data ___/___/___, hora: _____; Descreva o feed-back: _____; <input type="checkbox"/> 3. Uma versão corrigida foi enviada. Caso positivo, anote o momento do envio: data ___/___/___, hora: _____;</p> <p>3 Modificabilidade <input type="checkbox"/> 1. A falha foi satisfatoriamente resolvida; <input type="checkbox"/> 2. Foi possível eliminar a falha via mudança de parâmetros; <input type="checkbox"/> 3. Os itens de especificação foram mudados visando a remoção da falha. Caso positivo, anote o número de itens de especificação de requisitos mudados visando a remoção da falha: _____; <input type="checkbox"/> 4. A documentação foi mudada em decorrência da modificação. Caso positivo, anote o número de seções que foram mudadas: _____; <input type="checkbox"/> 5. modificação foi efetivamente registrada no log de modificações.</p> <p>4 Estabilidade <input type="checkbox"/> 1. Caso a falha foi satisfatoriamente resolvida, foram propagadas novas falhas decorrentes da modificação? Caso positivo, anote o número de falhas propagadas identificadas: _____; Descreva-as: _____;</p> <p>5 Testabilidade <input type="checkbox"/> 1. Caso a falha em questão foi satisfatoriamente resolvida e coube a utilização de teste built-in (ex.: simulação), foi possível utilizar adequadamente tal função; <input type="checkbox"/> 2. Caso a falha em questão foi satisfatoriamente resolvida, foi possível realizar teste caixa-preta aplicando pausa em pontos de checagem; <input type="checkbox"/> 3. Caso foi realizar teste caixa-preta aplicando pausa em pontos de checagem, foi possível recomençar a execução do teste.</p> <p>6 Conformidade Descrição dos itens de conformidade adequados à modificação para eliminação da falha: _____ Número de itens de conformidade considerados na eliminação da falha _____; Número de itens de conformidade observados na modificação _____.</p>
--

Figura 4. Formulário para Avaliação Externa da Manutenibilidade
Fonte: Adaptado de AGUAYO, M; GUERRA, A; COLOMBO, R, (2005, p. 8)

Para avaliar a estabilidade devem ser consideradas as seguintes situações (AGUAYO; GUERRA; COLOMBO, 2005):

- a) quando aparece uma falha na versão inicial, deverá tentar-se a replicação na versão modificada;
- b) quando aparece uma falha na versão modificada que não tinha sido registrada na versão inicial, ela deverá ser identificada, bem como a seqüência de passos ou a situação que leva à sua replicação na versão modificada e, deve-se tentar-se replicá-la na versão inicial.

A decisão da realização de outro ciclo será feita segundo os objetivos da avaliação. O relatório final deverá conter o histórico do processo de avaliação, e os resultados (AGUAYO; GUERRA; COLOMBO, 2005).

O processo de avaliação da manutenibilidade pode ser visto na Figura 5.

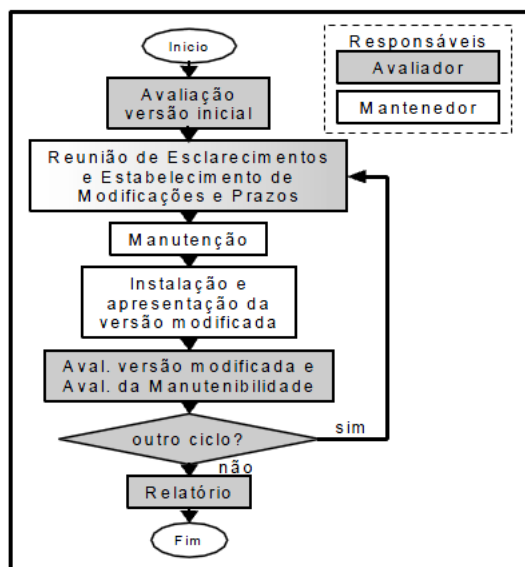


Figura 5. Processo de Avaliação da Manutenibilidade
 Fonte: AGUAYO, M; GUERRA, A; COLOMBO, R, (2005, p. 5)

Para projetar a avaliação, se estabelece um plano, contendo a especificação de: instrumentos, pessoas e responsabilidade e um cronograma (AGUAYO; GUERRA; COLOMBO, 2005).

Os instrumentos que o avaliador deve dispor são (AGUAYO; GUERRA; COLOMBO, 2005):

- a) método de avaliação da qualidade de produto de software a ser aplicado nas versões inicial e modificada;
- b) guia de avaliação da manutenibilidade, contendo formulário de identificação da avaliação e questionário a ser aplicado para cada modificação (falha, melhoria ou adaptação) planejada;
- c) manual da avaliação da manutenibilidade, contendo diretrizes, a seqüência das atividades, e explicação das questões presentes no guia citado acima;
- d) modelo de roteiro de esclarecimentos e pontos acertados para modificação;
- e) modelo de relatório da avaliação da manutenibilidade;
- f) ferramenta de coleta e armazenamento dos dados da avaliação;
- g) recursos necessários: o avaliador deverá ter um computador exclusivo, para executar a avaliação do produto com os requisitos de hardware e software necessários para tal execução;

As pessoas responsáveis pela avaliação formam a equipe de avaliação. A interação com o desenvolvedor é de grande importância, sem isso é praticamente impossível medir a manutenibilidade (AGUAYO; GUERRA; COLOMBO, 2005).

O cronograma deverá ser estabelecido do início ao fim, para o avaliador não perder o foco da avaliação (AGUAYO; GUERRA; COLOMBO, 2005).

As entradas da avaliação são (AGUAYO; GUERRA; COLOMBO, 2005):

- a) versão inicial do produto, com descrição do produto, manual para o usuário e documentos de especificação de conformidade a normas, convenções e regulamentos relacionados;
- b) versão modificada, com descrição do produto, manual para o usuário e, caso mude, os documentos de especificação de conformidade;
- c) relatórios de avaliação da versão inicial, resolução de problemas durante a manutenção, testes durante a manutenção, e operação após a manutenção, avaliação da versão modificada;
- d) procedimento da avaliação;
- e) formulário de questionário.

5.3 QUALIDADE EM USO

Qualidade em uso é a qualidade estimada para o produto final de software, em cada estágio de desenvolvimento e para cada característica de qualidade em uso, baseada no conhecimento da qualidade interna e externa.

Qualidade em uso é a qualidade do produto de software na visão do usuário, quando usado em um ambiente e um contexto de uso especificados. Ela mede o quanto usuários podem atingir seus objetivos num determinado ambiente e não as propriedades do software em si (NBR ISO/IEC 9126-1, 2003).

A obtenção da qualidade em uso é dependente da obtenção da necessária qualidade externa, a qual, por sua vez, é dependente da obtenção da necessária qualidade interna (NBR ISO/IEC 9126-1, 2003).

Normalmente, são necessárias medidas em todos os três níveis. Atender aos critérios para medidas internas, em geral não é suficiente para garantir o atendimento aos critérios para medidas externas. Atender aos critérios para medidas externas de sub-características, em geral não é suficiente para garantir o atendimento aos critérios para qualidade em uso (NBR ISO/IEC 9126-1, 2003).

Os atributos de qualidade são categorizados em quatro características que determinam as capacidades dos produtos de software: efetividade, produtividade, segurança e satisfação, como mostra a figura 6.

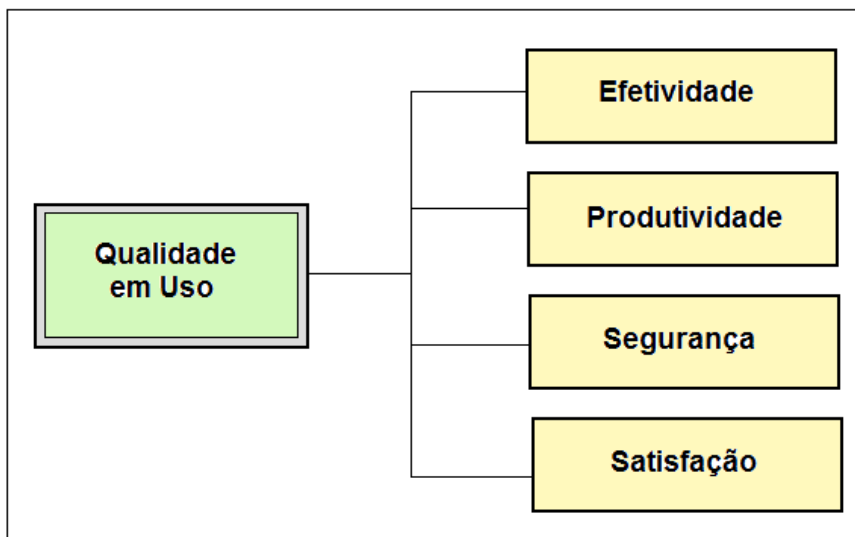


Figura 6. Características de Qualidade em Uso
Fonte: Adaptado de SODRÉ, C (2006, p. 33)

Efetividade é a capacidade de permitir que o usuário alcance os objetivos especificados com exatidão e completude, em um contexto específico de uso (SODRÉ, 2006).

Produtividade é a capacidade de possibilitar ao usuário gastar quantidades apropriadas de recursos em relação à efetividade obtida em um contexto de uso específico (SODRÉ, 2006).

Segurança é a capacidade de atingir apropriadamente, níveis aceitáveis de riscos de danos a pessoas, negócios, software ou ambiente em um contexto de uso específico (SODRÉ, 2006).

Satisfação é a capacidade de satisfazer os usuários em um contexto específico de uso (SODRÉ, 2006).

A qualidade deve ser acompanhada durante todo o ciclo de vida do software, qualidade interna, externa e em uso devem existir simultaneamente conforme a Figura 7.

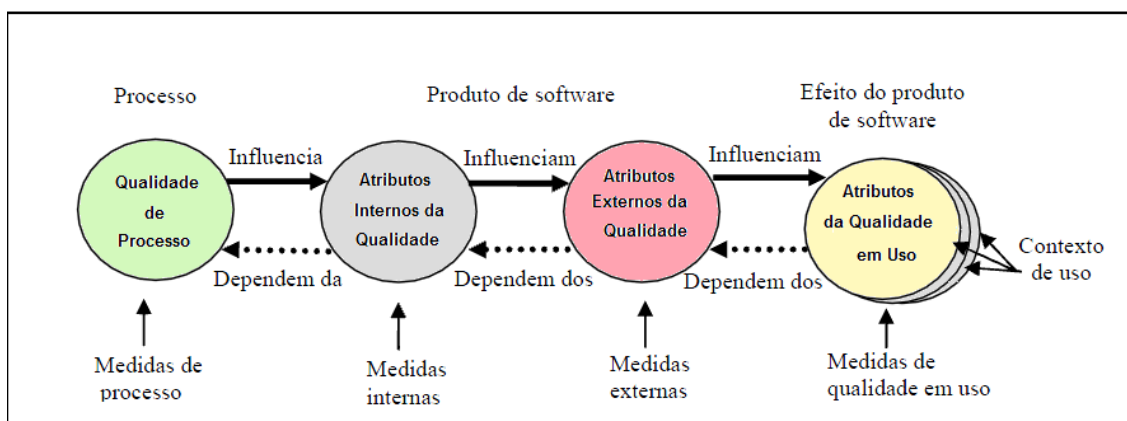


Figura 7. Qualidade no Ciclo de Vida
Fonte: Adaptado de Ferrari, S (2007, p. 106)

Não se pode medir todas as sub-características internas e externas para todas as partes de um produto de software de grande porte. Também não é prático medir a qualidade em uso para todos os cenários de uso. É necessário alocar recursos para avaliação entre os diferentes tipos de medições, dependendo dos objetivos de negócios e da natureza do produto e dos processos utilizados no projeto (NBR ISO/IEC 9126-1, 2003).

5.4 ISO/IEC 14598

Avaliar a qualidade é um processo que exige tempo, pessoal e recursos. A norma ISO/IEC 14598 define as principais características de um processo de avaliação (NBR ISO/IEC 14598-5, 2001):

- a) repetibilidade: avaliações repetidas do mesmo produto, com a mesma especificação de avaliação e pelo mesmo avaliador devem apresentar resultados que possam ser aceitos como idênticos;
- b) reprodutibilidade: as avaliações do mesmo produto, com a mesma especificação de avaliação, por avaliadores diferentes devem produzir resultados idênticos;
- c) imparcialidade: convém que a avaliação não seja tendenciosa com relação a algum resultado particular;
- d) objetividade: resultados das avaliações devem ser baseados em fatos, desprovidos de sentimentos ou opiniões do avaliador.

Um único produto quando avaliado com diferentes especificações de avaliação pode apresentar resultados distintos (NBR ISO/IEC 14598-5, 2001).

Segundo Sodré (2006) a ISO/IEC 14598 possui um conjunto de guias para orientar e planejar o processo de avaliação de um produto de software: ISO/IEC 14598-1, ISO/IEC 14598-2, ISO/IEC 14598-3, ISO/IEC 14598-4, ISO/IEC 14598-5, e ISO/IEC 14598-6, conforme ilustrado na Figura 8.

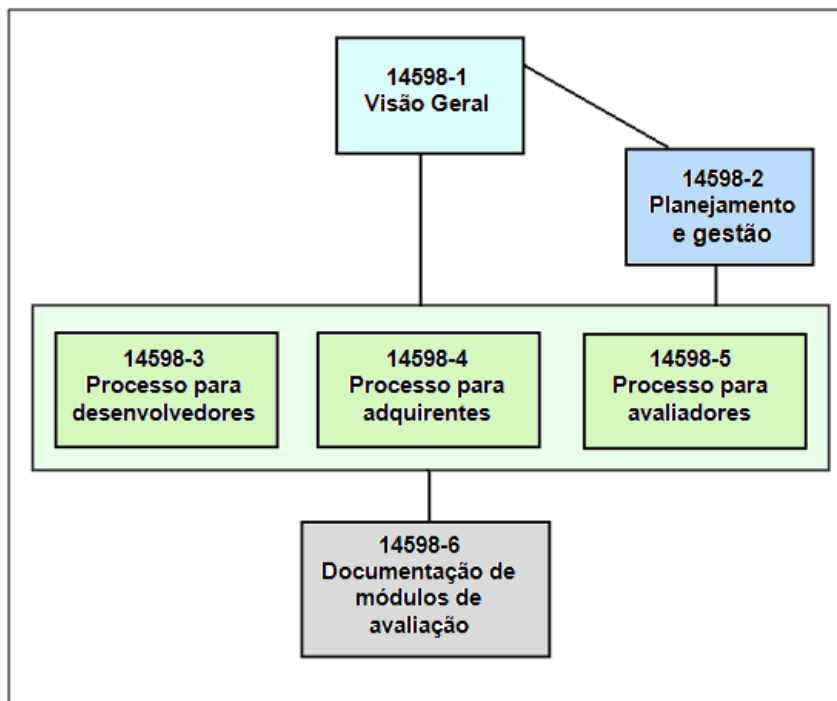


Figura 8. Representação da Norma ISO/IEC 14598
 Fonte: Adaptado de SODRÉ, C (2006, P.24)

5.4.1 ABNT NBR ISO/IEC 14598-1

A NBR ISO/IEC 14598-1:2001 – Tecnologia de informação – Avaliação de produto de software – Parte 1: Visão geral é uma estrutura de funcionamento das normas para a avaliação da qualidade de produto de software e definição dos termos técnicos desse modelo. Possui conceitos e funcionamento de processo de avaliação de qualquer tipo de software, para utilização de pessoas envolvidas em desenvolvimento, uso de tecnologia de avaliação e padronização (NBR ISO/IEC 14598-1, 2001).

5.4.2 ABNT NBR ISO/IEC 14598-2

A NBR ISO/IEC 14598-2:2003 - Engenharia de software – Avaliação de produto – Parte 2: Planejamento e gestão fornece requisitos e guias para suportar funções de avaliação

dos produtos de software, relacionados ao desenvolvimento, aquisição, padronização, controle, transferência e realimentação do uso de tecnologias de avaliação (NBR ISO/IEC 14598-2, 2003).

Essa norma é aplicável à função de apoio, a qual fornece, para todos os projetos em organizações de desenvolvimento de software um apoio no âmbito organizacional (NBR ISO/IEC 14598-2, 2003).

A NBR ISO/IEC 14598-2 (2003) afirma que é de responsabilidade da função de apoio obter:

- a) normas nacionais e internacionais relevantes;
- b) informações técnicas;
- c) apoio especializado;
- d) desenvolvimento de normas e ferramentas internas, baseadas em requisitos de projeto e organizacionais;
- e) desenvolvimento de critérios para estabelecer benchmarks para a avaliação (ato de executar um software a fim de avaliar a performance relativa executando uma série de testes padrões e ensaios nele);
- f) revisão da eficácia e da qualidade de qualquer aquisição ou desenvolvimento de software;
- g) coleta e análise de resultados de avaliação e sua disseminação dentro da organização com o uso de uma base de dados;
- h) facilitação da transferência de tecnologia, baseada nas experiências dentro da organização e em seus respectivos projetos de avaliação;
- i) apoio a todos os projetos de avaliação e a seus respectivos gerentes de projeto.

A função de apoio oferece novas tecnologias, normas nacionais e internacionais, expertise (consultoria), treinamento, base de dados organizacional, apoio a projetos de avaliação (NBR ISO/IEC 14598-2, 2003).

Segundo a NBR ISO/IEC 14598-2 (2003) os seguintes passos devem ser seguidos no planejamento e execução de avaliação de software:

- a) definir os objetivos da avaliação de software;
- b) assegurar que para todo projeto de avaliação seja elaborado um plano de avaliação quantitativa. Este pode ser subdividido em planos mais detalhados, de acordo com a complexidade de cada avaliação;
- c) incluir na base de dados da organização as experiências de projeto e de resultados da avaliação do produto, para melhorar a abordagem de avaliação de software da organização;

A NBR ISO/IEC 14598-2 (2003) assegura que as avaliações de software devem ser executadas conforme os itens a seguir:

- a) avaliar se o software está de acordo com normas internacionais, nacionais ou internas (quando aplicáveis);
- b) garantir que os resultados da avaliação possam ser quantificados, claramente apresentados e rastreados;
- c) garantir que estejam sendo utilizadas tecnologias adequadas e eficazes, e as melhores práticas;
- d) garantir que a avaliação seja realizada de modo eficaz;
- e) garantir a disponibilidade de planos e recomendações para dar apoio a todas as futuras atividades de avaliação.

5.4.3 ABNT NBR ISO/IEC 14598-3

A NBR ISO/IEC 14598-3:2003 - Engenharia de software – Avaliação de produto – Parte 3: Processo para desenvolvedores é uma norma para ser usada durante o desenvolvimento e manutenção do software. Fornece critérios para seleção de indicadores de qualidade, guia para avaliar dados de medição e guia para melhoria do processo de medição (NBR ISO/IEC 14598-3, 2003).

Essa norma destina-se ao uso durante o desenvolvimento de software. Ela é aplicável a todas as atividades de desenvolvimento de software que requeiram um processo disciplinado e à todas as fases do ciclo de vida. Fornece requisitos e recomendações para a implementação de avaliação de produto de software quando a avaliação é conduzida em paralelo com o desenvolvimento e executada pelo desenvolvedor (NBR ISO/IEC 14598-3, 2003).

A NBR ISO/IEC 14598-3 fornece orientação para detalhar requisitos de qualidade e para implementar e analisar medidas de qualidade de software. Focaliza a seleção e relato de indicadores úteis para prever a qualidade do produto final pela medição da qualidade de produtos intermediários, e também à medição da qualidade de produto final (NBR ISO/IEC 14598-3, 2003).

Descreve as atividades necessárias para analisar os requisitos de avaliação para especificar, projetar e executar as ações de avaliação, e concluir a avaliação de qualquer tipo de produto de software (NBR ISO/IEC 14598-3, 2003).

Atributos dos produtos intermediários são indicados e medidos nas atividades de desenvolvimento, permitindo monitorar e controlar quantitativamente a qualidade do produto de software em desenvolvimento. O objetivo é identificar problemas para obter a qualidade desejada, tão cedo quanto possível (NBR ISO/IEC 14598-3, 2003).

A seleção de atributos do produto de software, dos produtos intermediários e das atividades de desenvolvimento é baseada na experiência da organização desenvolvedora (NBR ISO/IEC 14598-3, 2003).

Identificar as necessidades dos usuários é um aspecto importante no estabelecimento de requisitos gerais de qualidade. Estes requisitos gerais são informais por natureza e necessitam ser formalizados, podem ser quantificados e avaliados por meio de métricas de qualidade em uso (NBR ISO/IEC 14598-3, 2003).

Atributos de qualidade externa e interna devem ser medidos durante o processo de desenvolvimento. São atribuídos valores de medição esperados para cada atributo, os valores reais são coletados durante o desenvolvimento. A avaliação se dá por meio da comparação dos valores reais e dos valores esperados (NBR ISO/IEC 14598-3, 2003).

Os requisitos de qualidade externa são traduzidos em requisitos de produtos intermediários e em atividades de desenvolvimento, por meio da tradução dos valores esperados de atributos externos em valores esperados de atributos internos, de produtos intermediários e em atividades de desenvolvimento (NBR ISO/IEC 14598-3, 2003).

Atributos internos podem ser usados para identificar irregularidades, mostrando que tais entidades precisam ser examinadas mais detalhadamente. Podem ser usados também para identificar tendências no desenvolvimento, que podem identificar problemas relacionados com produto e processo (NBR ISO/IEC 14598-3, 2003).

A NBR ISO/IEC 14598-3 (2003) afirma que a avaliação consiste em um conjunto de atividades conduzidas pelo desenvolvedor e executadas baseadas nos valores medidos durante o processo de desenvolvimento:

- a) estabelecimento dos requisitos de avaliação: identificar os requisitos gerais de qualidade, de acordo com um modelo de qualidade previamente aceito;

- b) especificação da avaliação: determinar métricas externas e internas e os valores esperados de medição para cada uma delas (critérios para avaliação);
- c) projeto da avaliação: planejar as ações para a coleta dos dados;
- d) execução da avaliação: obter os valores das medidas internas durante o desenvolvimento do produto e compará-los com os valores esperados (avaliação durante o desenvolvimento). Os valores dos atributos internos são utilizados para estimar a qualidade do produto final. Os valores das medidas externas também devem ser coletados e comparados com os valores esperados (avaliação da qualidade de produto);
- e) *feedback* para a organização, com base numa revisão dos resultados da avaliação.

Algumas ferramentas podem auxiliar a avaliação, como por exemplo, analisadores de código-fonte para calcular métricas de código, ferramentas CASE para produzir modelos formalizados, ambientes de teste para processar os programas executáveis ou planilhas eletrônicas para produzir sínteses de medidas (NBR ISO/IEC 14598-3, 2003).

Para elaborar uma especificação de avaliação a fim de satisfazer alguns requisitos, é necessário especificar métricas (NBR ISO/IEC 14598-3, 2003).

Segundo a NBR ISO/IEC 14598-3 (2003) as métricas são baseadas nas técnicas de avaliação que podem ser escolhidas de acordo com as características de qualidade e níveis de avaliação. A seguir é proposta, para cada característica de qualidade na NBR ISO/IEC 9126-1, uma lista de técnicas de avaliação, ordenadas em níveis crescentes de exigência:

- a) funcionalidade: teste funcional ou caixa-preta; inspeção de documentos de desenvolvimento, orientada por listas de verificação; teste de unidade com critério de cobertura;

- b) confiabilidade: verificação do uso de facilidades de uma determinada linguagem de programação; análise de construções tolerantes a falhas no projeto e implementação do software; modelo de crescimento de confiabilidade;
- c) usabilidade: inspeção de documentação e de interface com o usuário; verificação da conformidade aos padrões de interface; execução de experimentos de uso com usuários reais;
- d) eficiência: medição de tempo de execução; testes comparativos (benchmark); análise do projeto para determinar a complexidade algorítmica;
- e) manutenibilidade: inspeção de documentos de desenvolvimento orientada por listas de verificação; verificação de métricas de código e regras de programação; análise de rastreabilidade entre elementos da documentação de desenvolvimento;
- f) portabilidade: análise dos procedimentos de instalação de software; verificação de regras de programação; análise de projeto de software.

5.4.4 ABNT NBR ISO/IEC 14598-4

A NBR ISO/IEC 14598-4:2003 – Engenharia de software – Avaliação de produto – Parte 4: Processo para adquirentes é destinada (mas não limitada) a gerentes de projetos, engenheiros de sistemas, equipes de desenvolvimento e manutenção de software e usuários finais que planejam adquirir produtos de software e, também, para fornecedores de tais produtos (NBR ISO/IEC 14598-4, 2003).

Essa norma contém requisitos, recomendações e orientações para a medição, julgamento e avaliação sistemática da qualidade durante a aquisição de produtos de software

de prateleira, produtos de software sob encomenda ou, modificações em produtos de software existentes (NBR ISO/IEC 14598-4, 2003).

O processo de avaliação da NBR ISO/IEC 14598-4 auxilia a decidir sobre a aceitação de um único produto ou selecionar um entre vários produtos. A norma foi projetada para ser usada em conjunto com outras normas (NBR ISO/IEC 14598-4, 2003).

Para a NBR ISO/IEC 14598-4 (2003) as atividades do processo de aquisição são:

- a) iniciação: identificação dos requisitos de software para o produto a ser adquirido, preparo do plano de aquisição e definição da estratégia e dos critérios de aceitação;
- b) preparação de pedido de proposta: especificação e documentação dos requisitos de aquisição;
- c) preparação e atualização do contrato: seleção de fornecedores, preparação e negociação do contrato e controle de modificações no contrato;
- d) monitoração do fornecedor: atividades de avaliação aplicadas durante a execução do contrato, conduzindo a aceitação e a entrega do produto de software;
- e) aceitação e conclusão: atividades executadas durante a aceitação do produto e entrega do produto final de software.

5.4.5 ABNT NBR ISO/IEC 14598-5

Produtos de software têm se tornado cada vez mais importantes em todas as áreas da indústria e de serviços, diante disso é necessário que se possa avaliar a qualidade desses produtos (NBR ISO/IEC 14598-5, 2001).

A NBR ISO/IEC 14598-5:2001 - Tecnologia de informação – Avaliação de produto de software – Parte 5: Processo para avaliadores, fornece requisitos e recomendações para a implementação prática de avaliação de um produto de software, quando várias partes envolvidas necessitam entender, aceitar e confiar nos resultados da avaliação. Em especial ela pode ser usada na aplicação dos conceitos descritos na NBR ISO/IEC 9126 (NBR ISO/IEC 14598-5, 2001).

Existe uma variedade de produtos de software, produzidos para atender requisitos muito diversificados, portanto, o contexto para o uso também pode variar: software aplicativo em um sistema de informações gerenciais, software embutido em outros produtos, ou software de jogos, etc. (NBR ISO/IEC 14598-5, 2001).

A NBR ISO/IEC 14598-5 (2001) afirma que os potenciais benefícios da avaliação de um produto de software são:

- a) o desenvolvedor pode usar os resultados da avaliação de seu produto para identificar ações corretivas, com o objetivo de melhorá-lo ou de tomar decisões sobre a estratégia de evolução do produto;
- b) para os fornecedores, o benefício de uma avaliação pode ser obter confiança no valor do produto, além disso o relatório de avaliação pode ser usado com finalidades comerciais;
- c) para adquirentes, resultados de avaliação podem ser usados como dados objetivos para decisões de aquisição;
- d) para negócios em geral, a disseminação da avaliação de produto de software pode ajudar no uso da qualidade como argumento de marketing.

Na avaliação, documentos de projeto, relatórios de teste e validação, código fonte e documentação de usuário podem ser considerados parte do produto de software (NBR ISO/IEC 14598-5, 2001).

Avaliar um produto de software é uma operação técnica que permite elaborar um julgamento de uma ou mais características de acordo com um procedimento definido, seqüência de passos que determinam os requisitos de avaliação por meio de características de qualidade (NBR ISO/IEC 14598-5, 2001).

O processo de avaliação pode ser usado para avaliar produtos já existentes, caso os componentes do produto necessários estejam disponíveis, e também para avaliar produtos em desenvolvimento (NBR ISO/IEC 14598-5, 2001).

As atividades do processo de avaliação usam como base os dados fornecidos pelo requisitante, pelo avaliador, ou produzidos por outras atividades do conjunto. Produzem dados como resultados do processo de avaliação, que podem ser utilizados por outras atividades do processo conforme Figura 9 (NBR ISO/IEC 14598-5, 2001).

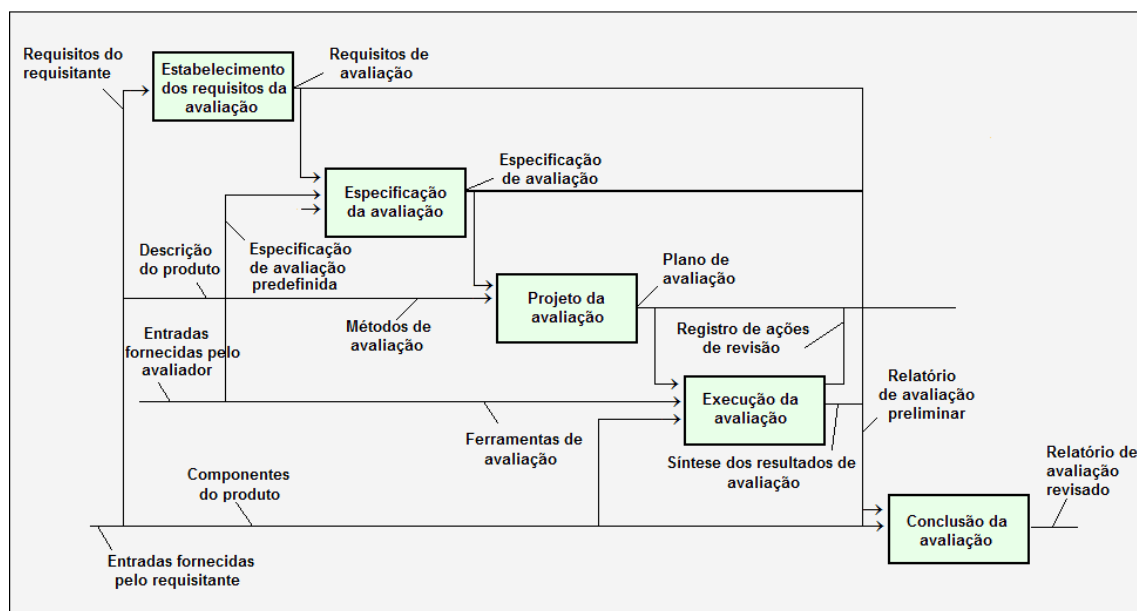


Figura 9. Processo de Avaliação
Fonte: Adaptado de NBR ISO/IEC 14598-5 (2001)

As técnicas de avaliação possíveis são numerosas e a sua seleção precisa levar em consideração os objetivos da avaliação e a constituição do produto (NBR ISO/IEC 14598-5, 2001).

Segundo a NBR ISO/IEC 14598-5 (2001) o processo de avaliação é composto de cinco atividades:

- a) estabelecer requisitos de avaliação: descrição geral do domínio da aplicação do produto; descrição geral do objetivo do produto; lista dos requisitos de qualidade e as informações do produto avaliado, incluindo sempre que possível, referência às características de qualidade e aos níveis de avaliação.
- b) especificar a avaliação: abrangência da avaliação, fazendo referência a descrição do produto; referência cruzada entre as informações solicitadas nos requisitos de avaliação e os componentes do produto; a especificação de medições e verificações; mapeamento entre a especificação de medições e verificações dos requisitos de avaliação;
- c) projeto da avaliação: um plano baseado na especificação da avaliação, atividade que leva em consideração os componentes do produto de software a serem avaliados e os métodos de avaliação propostos. O relatório de avaliação deve conter a documentação dos métodos de avaliação utilizados para executar a avaliação. O projeto de avaliação deve documentar os procedimentos a serem usados pelo avaliador para realizar as medições contidas na especificação de avaliação (os recursos considerados aqui podem ser recursos humanos para executar as ações de avaliação, recursos computacionais ou espaço físico);
- d) execução do plano de avaliação: inspeção, modelagem, medição e teste dos produtos e de seus componentes de acordo com o plano de avaliação.

- e) conclusão da avaliação: liberação do relatório de avaliação e devolução do produto avaliado e de seus componentes, resultados da avaliação propriamente ditos; resultados intermediários ou decisões de interpretação, sempre que necessário; referência as ferramentas usadas durante a avaliação.

O plano de avaliação é composto de três sub-atividades: documentar os métodos de avaliação e produzir um plano preliminar; otimizar o plano de avaliação; programar as ações de avaliação, tendo em consideração os recursos disponíveis. Deve ser composto por duas partes, a documentação dos métodos de avaliação e o cronograma das ações do avaliador (NBR ISO/IEC 14598-5, 2001).

Quando o plano de avaliação requer que o programa executável do produto seja testado, a configuração em teste e o ambiente de teste devem ser registrados com precisão (NBR ISO/IEC 14598-5, 2001).

A NBR ISO/IEC 14598-5 (2001) descreve as seguintes entradas referente ao produto necessárias ao processo de avaliação:

- a) descrição que identifica o produto de software;
- b) componentes do produto.

O produto pode incluir os documentos relacionados ao planejamento, processo ou métodos de desenvolvimento utilizados na sua produção. O documento de planejamento pode incluir cronograma, estrutura organizacional e estimativa de custos (NBR ISO/IEC 14598-5, 2001).

Conforme a NBR ISO/IEC 14598-5 (2001) as entradas fornecidas pelo avaliador são:

- a) especificações de avaliações predefinidas;
- b) métodos de avaliação;
- c) ferramentas de avaliação.

A NBR ISO/IEC 14598-5 (2001) descreve que durante o processo de avaliação serão fornecidas pelo avaliador as seguintes saídas:

- a) registros de avaliação, incluindo plano de avaliação e registro das ações de avaliação;
- b) relatório preliminar de avaliação, incluindo requisitos de avaliação, especificação de avaliação e síntese dos resultados da avaliação;
- c) relatório de avaliação revisado.

Os requisitos, a especificação e o plano de avaliação são produtos intermediários do processo de avaliação. Os requisitos de avaliação que descrevem os objetivos da avaliação e o relatório de avaliação são os produtos finais do processo de avaliação (NBR ISO/IEC 14598-5, 2001).

A especificação de avaliação define análises e medições a serem realizadas. Os componentes do produto que serão analisados e medidos devem ser identificados. Os métodos e ferramentas a serem utilizados na avaliação devem ser descritos (NBR ISO/IEC 14598-5, 2001).

Os registros de avaliação consistem no plano de avaliação e no detalhamento das ações realizadas pelo avaliador durante a execução do plano de avaliação e são mantidos pelo avaliador (NBR ISO/IEC 14598-5, 2001).

O relatório de avaliação contém os requisitos de avaliação, a especificação de avaliação, resultado das medições e das análises realizadas e outras informações necessárias para permitir a repetição ou reprodução da avaliação. Inicialmente é emitido como uma versão preliminar para revisão, quando na sua forma final, é entregue ao requisitante (NBR ISO/IEC 14598-5, 2001).

Quando realizada desde o início do processo de desenvolvimento, a avaliação assegura uma maior probabilidade de o produto satisfazer aqueles requisitos considerados no

processo de avaliação, e minimiza o risco de ocorrência de custos extras ou inesperados (NBR ISO/IEC 14598-5, 2001).

Algumas ações de avaliação devem ser feitas no ambiente de operação. Para projetos muito grandes e complexos, o desenvolvedor deve cooperar continuamente durante o desenvolvimento do produto para minimizar a duração e o custo do processo de avaliação (NBR ISO/IEC 14598-5, 2001).

A avaliação e melhoria de um produto de software permite melhorar a qualidade em uso e assim satisfazer o cliente. A avaliação da qualidade em uso pode fornecer *feedback* para melhorar o produto e avaliar o produto pode fornecer *feedback* para melhorar um processo (NBR ISO/IEC 14598-5, 2001).

As visões de qualidade interna, externa e em uso podem variar conforme o estágio do ciclo de vida. Os requisitos de qualidade no início do ciclo de vida é uma visão de qualidade externa e do usuário, a qualidade na fase de projeto é uma visão de qualidade interna e do desenvolvedor (NBR ISO/IEC 14598-5, 2001).

As tecnologias utilizadas para atingir o nível de qualidade necessário, como especificação e avaliação de qualidade devem apoiar os diversos pontos de vista e os diversos estágios do ciclo de vida (NBR ISO/IEC 14598-5, 2001).

5.4.6 ABNT NBR ISO/IEC 14598-6

A NBR ISO/IEC 14598-6:2004 - Engenharia de software – Avaliação de produto – Parte 6: Documentação de módulos de avaliação, define a estrutura e o conteúdo da documentação a ser usada para descrever um Módulo de Avaliação (MA). Esta norma deve ser usada por peritos em tecnologia de avaliação, tais como laboratórios de teste, institutos de

pesquisa e outros, quando produzindo novos módulos de avaliação (NBR ISO/IEC 14598-6, 2004).

Um MA é um pacote de tecnologia de avaliação para medir características, sub-características ou atributos de qualidade de software. O pacote inclui: técnicas e métodos de avaliação; entradas para a avaliação; dados a serem coletados e medidos; ferramentas e procedimentos de apoio. Tecnologia de avaliação são as técnicas, ferramentas, métricas, medidas e outras informações técnicas usadas para avaliação (NBR ISO/IEC 14598-6, 2004).

A avaliação de produtos de software pode ser uma tarefa muito abrangente. Aspectos diferentes de características e sub-características de qualidade podem requerer que diferentes técnicas de avaliação sejam aplicadas e diferentes dados sejam coletados (NBR ISO/IEC 14598-6, 2004).

A avaliação deve ser estruturada em unidades gerenciais e cada uma delas pode cobrir um ou mais aspectos de qualidade. Cada unidade deve focar na avaliação de um aspecto de qualidade particular, aplicando uma técnica de avaliação específica. A informação necessária para conduzir uma destas avaliações deve ser coletada e empacotada para uso futuro. Este pacote é chamado de MA (NBR ISO/IEC 14598-6, 2004).

Segundo a NBR ISO/IEC 14598-6 (2004) a documentação de um MA possui seis partes:

- a) MA0: apresenta informações oficiais sobre o MA e uma introdução sobre a técnica de avaliação descrita no MA;
- b) MA1: define o objetivo do MA;
- c) MA2: fornece referências relevantes;
- d) MA3: inclui as definições necessárias ao MA;
- e) MA4: especifica os produtos de entrada requeridos para a avaliação e define os dados a serem coletados e as medidas a serem obtidas;

- f) MA5: contém informações sobre como interpretar os resultados da medição;
- g) MAA: anexo opcional que inclui o procedimento detalhado para a aplicação do MA. Embora seja opcional, sugere-se que o MAA seja incluído.

5.5 ISO/IEC 25000

A norma ISO/IEC 25000 ou, Requisitos de Qualidade e Avaliação de Produtos de Software (SQuaRE) é uma evolução da ISO/IEC 9126 e da ISO/IEC 14598 (KOSCIANSKI; SOARES, 2006).

Um total de 10 documentos compõem a ISO/IEC 9126 e a ISO/IEC 14598, assim para compreender um determinado tópico o leitor poderia ter de consultar vários desses documentos. O conteúdo abordado pelas normas era relevante, porém a maneira como era apresentado merecia ser repensada (KOSCIANSKI; SOARES, 2006).

A idéia era criar um guia ou manual de utilização para auxiliar a compreender idéias e conceitos básicos das normas, assim como as informações contidas ao longo dos 10 documentos, por isso criaram a ISO/IEC 25000 (KOSCIANSKI; SOARES, 2006).

O material já existente nas duas normas foi reorganizado, mas não sofreu mudanças radicais. O modelo hierárquico de qualidade proposto na 9126 e os diversos aspectos organizacionais abordados na 14598, continuaram válidos, conforme Tabela 3 (KOSCIANSKI; SOARES, 2006).

Tabela 3. Partes das Normas ISO/IEC 9126 e 14598

Norma	Conteúdo
9126-1	Modelo de qualidade de software
9126-2	Métricas externas
9126-3	Métricas internas
9126-4	Métricas para qualidade em uso
14598-1	Guia de avaliação – visão geral
14598-2	Planejamento e gerenciamento de avaliações
14598-3	Processo de avaliação para desenvolvedores
14598-4	Processo de avaliação para adquirentes
14598-5	Processo de avaliação para avaliadores
14598-6	Documentação de módulos de avaliação

Fonte: Adaptado de Koscianski, A; Soares, M (2006, p. 205)

Os motivos do surgimento da norma SQuaRE são (KOSCIANSKI; SOARES, 2006):

- a) abrange amplamente o assunto e estabelece uma base precisa para definir o modelo e para realizar a avaliação;
- b) os documentos contemplam certo caráter didáticos, houve a preocupação em fornecer uma extensa lista de exemplos de métricas, o que não existe em nenhum documento anterior;
- c) os documentos são resultados de um esforço e consenso de centenas de pesquisadores e representam assim, uma soma de experiências únicas no assunto;
- d) outros modelos de qualidade elaborados por pesquisadores de maneira pontual podem ser mapeados para o modelo SQuaRE.

O projeto SQuaRE estabelece uma linguagem, ou vocabulário comum e válido internacionalmente, evitando ambigüidades ao analisar um modelo de qualidade. O modelo é formado por cinco divisões: gerenciamento, modelo de qualidade, medição, requisitos de qualidade e avaliação (KOSCIANSKI; SOARES, 2006).

O gerenciamento consiste nos documentos voltados para os possíveis usuários, como gerentes, programadores, avaliadores ou compradores. Termos utilizados nos demais documentos são definidos, são feitas recomendações e sugestões de caráter geral, sobre como utilizar o SQuaRE, o documento apresenta uma introdução geral a todo o conjunto de normas (KOSCIANSKI; SOARES, 2006).

O modelo de qualidade corresponde à antiga ISO/IEC 9126-1, define os conceito de qualidade externa, interna e em uso, que permitem orientar diferentes aspectos de avaliação. Visões e preocupações de desenvolvedores e clientes são diferentes, por isso é definido um modelo hierárquico de características de qualidade, permitindo que se faça uma descrição extensa e precisa do que cada um espera do produto (KOSCIANSKI; SOARES, 2006).

Depois de definida a medição, deve-se descrever os aspectos relacionados a essa tarefa, como, garantia da precisão dos resultados obtidos. Posteriormente deve-se propor uma série de métricas que podem ser utilizadas ou adaptadas pelos usuários das normas conforme suas necessidades (KOSCIANSKI; SOARES, 2006).

Os requisitos de qualidade são os objetivos para garantir a qualidade de um produto, é preciso que valores-alvo tenham sido previamente especificados. Esses valores fazem parte da especificação de requisitos do software (KOSCIANSKI; SOARES, 2006).

A norma é concretizada na avaliação de qualidade a partir de medições cujos resultados devem ser confrontados contra um modelo definido pelo usuário. A divisão de avaliação é direcionada aos diferentes públicos da norma, como desenvolvedores e compradores. São sugeridos procedimentos a serem adotados em cada caso para realizar uma avaliação (KOSCIANSKI; SOARES, 2006).

O modelo SQuaRE é voltado para a medição da qualidade de produtos de software, em todos os estágios do ciclo de vida e é dividido em: qualidade em uso, qualidade externa e qualidade interna (KOSCIANSKI; SOARES, 2006).

A qualidade relacionada ao ciclo de vida permite organizar o gerenciamento de um projeto, onde cada nível guarda uma dependência com o anterior. A qualidade em uso tem maior probabilidade de ser obtida se a qualidade externa for assegurada e esta, por sua vez, depende da qualidade interna (KOSCIANSKI; SOARES, 2006).

Esse modelo propõe definir requisitos em cada nível em função dos objetivos desejados, requisitos de qualidade em uso estabelecidos em meio aos usuários levam a determinação de requisitos de qualidade externa e, daí, a requisitos arquiteturais do software (KOSCIANSKI; SOARES, 2006).

Estabelecer os requisitos em cada nível é uma dificuldade prática resolvida com o acúmulo de experiências na empresa. A documentação do sistema de qualidade, uma biblioteca de características e métricas de qualidade, torna essa tarefa mais fácil. Essas características são as mesmas da ISO/IEC 9126 (KOSCIANSKI; SOARES, 2006).

5.6 ISO/IEC 12119

A norma ISO/IEC 12119 avalia produtos de software que estão disponíveis no mercado, conhecidos como pacotes de software ou software de prateleira.

Essa norma estabelece requisitos de qualidade e teste dos pacotes de software e define que o pacote que será testado deve possuir (SODRÉ, 2006):

- a) descrição do produto: documento com a definição das propriedades do produto, com o objetivo de orientar os compradores na avaliação antes da aquisição;

- b) manual do usuário: documentos fornecidos como parte integrante do produto que auxiliam na sua aplicação;
- c) programa e dados: conjunto de programas e dados necessários para a aplicação do produto de software e também como parte integrante deste.

Segundo Sodré (2007) essa norma foi publicada em 1994 e define que cada pacote de software deve possuir uma descrição do produto e uma documentação do usuário, e estabelece alguns requisitos de qualidade:

- a) descrição do produto compreensível e completa para ajudar o usuário ou comprador na avaliação da adequação do produto a sua realidade e fornecer informações comerciais;
- b) documentação do usuário de fácil compreensão, permitindo uma visão geral do produto e de todas as suas funções, identificando conhecimento necessário para uso da aplicação;
- c) identificação do tipo de interface com o usuário: interface gráfica, linhas de comandos, menu de comandos, janelas, etc.;
- d) instruções detalhadas sobre como instalar o produto, caso a instalação possa ser conduzida pelo usuário;
- e) possibilidade de verificar se a instalação foi bem sucedida;
- f) especificação de valores-limite para quantidade de registros e dados de entrada, como, por exemplo, precisão de casa decimal;
- g) operação normal, mesmo quando os dados informados estão fora dos limites especificados;
- h) consistência de vocabulário entre as mensagens e a documentação;
- i) função de auxílio (help) sensível ao contexto;
- j) mensagens de erro com informações necessárias para solucionar o problema;

- k) diferenciação de tipos de mensagem: confirmação, consulta, advertência e erro;
- l) clareza e padronização nos formatos de telas de entrada, relatórios e outras entradas e saídas;
- m) capacidade de reverter funções de efeito drástico;
- n) capacidade de recuperar dados após uma falha de hardware ou software, queda de energia ou erro fatal;
- o) alertas claros para o usuário das conseqüências de uma determinada confirmação;
- p) identificação dos arquivos utilizados pelo programa;
- q) identificação da função do programa que está sendo executada no momento;
- r) capacidade de interromper um processamento demorado.

Alguns dos problemas mais encontrados nos softwares são:

- a) descrição do produto incompleta;
- b) manual desatualizado ou incompleto;
- c) difícil de navegar entre as telas, aprender e compreender;
- d) *help* não sensível ao contexto;
- e) funcionalidade prometida não implementada;
- f) erros fatais levam a perda de dados.

As vezes pelo fato de ser mais conveniente e lucrativo para a empresa deixar pendências e melhorias para serem implementadas nas próximas versões, os problemas acabam não sendo solucionados (GOMES, 2000).

Problemas não solucionados causam resultados incertos e inseguros. Softwares que guardam e fornecem informações gerenciais são de grande importância no dia-a-dia de organizações comerciais, através deles são controlados lucros e prejuízos gerados e tomadas as decisões administrativas, portanto esses sistemas precisam sofrer avaliações de qualidade.

6 SISTEMA PARA GESTÃO EMPRESARIAL

As empresas precisam ter um bom relacionamento com o ambiente externo para conquistar seu espaço entre a concorrência. A capacidade dos administradores está sempre em julgamento, portanto eles precisam procurar as melhores formas de alcançar o crescimento sustentável e duradouro. Essa não é uma tarefa fácil, devido à quantidade de informações envolvidas no processo administrativo (SALLES, 2007).

Informação é um conjunto de dados organizados para gerar significado e valor que possa ser interpretado pelo receptor para que conclusões sejam encontradas.

A organização tem o desafio de acessar as informações, decifrar o seu valor estratégico e saber o momento certo para utilizá-las. A falta disso pode levar uma organização ao fracasso.

Para facilitar o uso e aplicação dessas informações é necessário um sistema que possibilite transformar essas informações em conhecimento, auxiliando nas tarefas administrativas e no cumprimento das estratégias estabelecidas.

O conhecimento é formado por dados e/ou informações organizados e processados com o objetivo de fornecer entendimento para ser aplicado aos problemas ou atividades empresariais (TURBAN; RAINER; POTTER, 2007).

Os sistemas e as informações influenciam nas decisões tomadas em uma empresa, a economia é baseada em informações. Os computadores, os softwares e as redes tornam as organizações mais flexíveis, eliminam deslizes na administração, separam os diversos locais de trabalho e reestruturam os fluxos de trabalho, auxiliando trabalhadores e administradores (SALLES, 2007).

Diferentes necessidades exigem diversos sistemas de informação que podem ter as mais variadas funções, mas com uma característica em comum, atender as necessidades dos

usuários. As empresas são mantidas pelo conhecimento, um amplo sistema formado pelos serviços de informação, sistemas de informação e um ambiente informacional (SALLES, 2007).

O conhecimento é fundamental para que as empresas se mantenham ativas, logo elas exigem cada vez mais de seus colaboradores. Os sistemas apresentam recursos organizacionais relacionados a recursos humanos, tecnológicos e financeiros por meio de recursos computacionais.

Os sistemas de informação dão suporte à tomada de decisão e auxiliam na gestão empresarial, armazenando e processando grandes quantidades de informações e transformando a estrutura organizacional. São vistos como um recurso estratégico que, devem atrair a atenção de gerentes e administradores e devem comportar-se de acordo com as situações impostas pelo mercado (KAPPEL, 2006).

Estes sistemas podem apoiar parte de uma organização, uma organização por inteiro ou, grupos de organizações.

6.1 FUNCIONALIDADES DE UM SISTEMA DE GESTÃO EMPRESARIAL

Os sistemas de gestão empresarial fazem uma integração dos Sistemas de Informação de área funcional, relacionando todos os departamentos de uma organização por meio de um banco de dados em comum, melhorando a comunicação entre as áreas funcionais, aumentando a produtividade organizacional, logo, uma mudança em uma das áreas afeta todas as outras (TURBAN; RAINER; POTTER, 2007).

Sistemas empresariais auxiliam nas atividades diárias de uma empresa, como vendas, receitas, depósito em dinheiro, decisões de crédito e o fluxo de materiais em uma fábrica, executando e registrando transações rotineiras necessárias na condução do negócio,

como por exemplo, os dados que compõem um pedido, ajudando a solucionar problemas relacionando empresa e ambiente externo (LAUDON; LAUDON, 1999).

Os Sistemas de Planejamento de Recursos Empresariais (ERP) oferecem as informações que controlam os processos empresariais, ou seja, controlam etapas e procedimentos relacionados para gerar resultados específicos. Tais processos podem estar localizados em uma única área funcional ou, se espalhar em diversas delas (TURBAN; RAINER; POTTER, 2007).

Um ERP possui módulos interdependentes ligados a um único banco de dados, que apóiam processos empresariais internos nas áreas de finanças e contabilidade, fabricação e produção, vendas e marketing e recursos humanos (TURBAN; RAINER; POTTER, 2007).

Estes sistemas são responsáveis pelo processamento dos dados de operações executadas em uma empresa, dando suporte às funções de gravação, monitoramento e avaliação das atividades básicas, permitindo que as organizações executem suas atividades mais importantes com maior facilidade (LAUDON; LAUDON, 1999).

7 TRABALHOS CORRELATOS

A qualidade de software é o objetivo de qualquer empresa de desenvolvimento de software e tem sido um instrumento de estudo em trabalhos de conclusão de cursos, teses, dissertações e etc.

Alguns trabalhos relacionados à qualidade de software e as práticas de Engenharia de Software foram utilizados como fonte de estudo para impulsionar o desenvolvimento desta pesquisa e estão descritos abaixo.

7.1 ELEMENTOS INTRÍNSECOS DO SOFTWARE E SUA INFLUÊNCIA NA QUALIDADE DO PROCESSO DE DESENVOLVIMENTO

Trabalho final de Mestrado Profissional de Izilda Gomes Garcez Capovilla, apresentado ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito parcial para obtenção do título de Mestre em Qualidade, em 1999.

Considerando a crescente influência dos softwares, a dependência das organizações sobre a qualidade do software, e os problemas freqüentemente encontrados, este trabalho apresenta elementos classificados como intrinsecamente naturais do software e analisa os desvios manifestados nos processos de desenvolvimento com relação a esses elementos. São discutidas algumas iniciativas na área de qualidade de software como o modelo CMM, projeto SPICE e normas ISO, e apresentadas considerações sobre aplicabilidade (CAPOVILLA, 1999).

7.2 NORMA ISO/IEC 9126: AVALIAÇÃO DE QUALIDADE DE PRODUTOS DE SOFTWARE

Trabalho de Conclusão de Curso desenvolvido por Cibele Cristina Pelizer Sodré em 2006, durante o 4º ano do Curso de Graduação em Ciência da Computação como requisito parcial à obtenção do título de Bacharel.

Este trabalho aborda a Qualidade de Software, detalhando normas e padrões de Qualidade de Processo de Software e Qualidade de Produto de Software, tendo como foco a descrição da norma ISO/IEC 9126, que é uma norma de Qualidade de produto de software (SODRÉ, 2006)

7.3 UMA PROPOSTA PARA QUALIDADE DE SOFTWARE POR MEIO DA APLICAÇÃO INTEGRADA DO MODELO CMM E DA NORMA NBR 13596

Trabalho Final de Mestrado Profissional, apresentado por Mary Lucy Sant'Ana em 2002 à comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Mestre Profissional em Engenharia Mecânica / Gestão da Qualidade Total.

Esta dissertação apresenta uma proposta de integração das duas abordagens: aplicação do modelo de processo CMM e a Norma NBR 13596, sobre características de qualidade de produto de software. Procurou-se analisar atividades, metas e produtos de trabalho das áreas de processo, proposto pelo CMM, verificando instâncias onde as definições, diretrizes e características de qualidade de produto de software abordadas na Norma NBR 13596 podem ser utilizadas (SANT'ANA, 2002).

7.4 PROPOSTA DE METODOLOGIA PARA CONTROLE DE QUALIDADE EM UMA FÁBRICA DE SOFTWARE

Tese apresentada por Sandra Ferrari, em 2007 ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina como requisito parcial para obtenção do título de doutor em Engenharia de Produção.

O objetivo deste trabalho é a proposição de uma abordagem que compreende a utilização combinada de reuso sistemático, teste e inspeção de software, no contexto de fábrica de software, visando à melhoria da qualidade (FERRARI, 2007).

8 AVALIAÇÃO DE PRODUTO DE SOFTWARE PARA GESTÃO EMPRESARIAL

Atualmente, a grande maioria das organizações comerciais necessitam de um software para gestão empresarial que, auxilie nas atividades diárias, facilitando o armazenamento dos dados e oferecendo informações confiáveis, relatórios gerenciais que ajudam na gerência financeira e comercial. Portanto é de grande importância que essas informações sejam confiáveis e verídicas.

Oferecer um produto de software de qualidade é o objetivo de qualquer empresa de desenvolvimento de software, porém isso não é uma tarefa fácil.

O desenvolvimento de um produto de software é dividido em processos e, para atingir a qualidade do produto final, esses processos devem ser avaliados para que sejam desempenhados de maneira correta.

Sabe-se que as grandes empresas de desenvolvimento de software aplicam padrões e ferramentas que oferecem auxílio durante o ciclo de vida de seus processos.

As normas que dão suporte ao longo do desenvolvimento dos produtos de software e avaliam seus processos são mundialmente conhecidas, como por exemplo, CMM, CMMI, algumas ISO, SPICE, MPS.BR, e freqüentemente são realizados estudos com essas normas.

Porém mesmo quando os processos são desenvolvidos de acordo com normas e padrões, isso não é garantia que o produto final terá qualidade, pois um produto de software tem características que determinam sua qualidade e que, variam conforme as necessidades de quem está avaliando.

Quando se fala de pequenas e médias empresas de desenvolvimento de software, a situação é mais complexa, pois a maioria de seus produtos são desenvolvidos sem a utilização de padrões e ferramentas que auxiliam na obtenção da qualidade.

As normas de qualidade de software auxiliam no desenvolvimento de software e na avaliação dos processos. Porém, para empresas desenvolvedoras de software de pequeno porte é muito difícil obterem normas e certificação de qualidade, pelo fato de ter um custo elevado.

Viu-se então, a necessidade de avaliar produtos de software já existentes, principalmente em empresas de pequeno porte, e sugerir meios de manter a qualidade desses produtos.

Vale ressaltar que o objetivo desta avaliação não é possuir uma certificação e sim conseguir melhorar a qualidade do software para ganhar a confiança e satisfação dos clientes e conseqüentemente possibilitar um crescimento comercial para a empresa desenvolvedora.

Acredita-se que as pequenas empresas, mesmo sem capital suficiente para conseguir uma certificação de qualidade, também podem oferecer para seus clientes um bom produto, com o auxílio das práticas de Engenharia de Software.

8.1 PRÁTICAS DE ENGENHARIA DE SOFTWARE

Para desenvolver este trabalho foi necessário estudar Engenharia de Software para identificar as diferenças entre processo e produto de software e as normas que permitem avaliar a qualidade de cada um, buscando por técnicas e padrões que auxiliam no desenvolvimento de software.

Esse estudo permitiu entender os conceitos que possibilitam estruturar uma avaliação de produto de software e sugerir algumas mudanças que, colocadas em prática podem resultar em uma melhoria considerável na qualidade do software.

Pode-se perceber que a Engenharia de Software tem papel fundamental no desenvolvimento e manutenção do software e que, a ausência dessas práticas pode afetar consideravelmente a qualidade do produto final.

O resultado de um produto de qualidade é a satisfação dos clientes, por isso é muito importante ouvi-los e avaliar suas colocações a respeito do software.

8.2 OS CLIENTES E A QUALIDADE DO PRODUTO DE SOFTWARE

Os diferentes clientes, usuários do software podem ter necessidades distintas e é difícil atender a todas elas. Porém, quando o número de insatisfação e reclamações é constante, a empresa desenvolvedora deve preocupar-se em mudar essa situação, pois é sinal de que algo não está em conformidade com as práticas de engenharia de software.

Com a função de suporte técnico na empresa desenvolvedora do software, é perceptível a quantidade de reclamações efetuadas pelos clientes em consequência de erros que aparecem no software com certa frequência.

Devido aos problemas enfrentados diante das reclamações não se consegue atender a todos os clientes com eficiência e rapidez, aumentando ainda mais o seu descontentamento.

Essa situação necessita de melhorias na qualidade do produto para conquistar a satisfação dos clientes atuais, para assim, conseguir novos clientes e, não perder os já conquistados.

Os relatos de erros e as reclamações dos clientes foram armazenados e analisados no cotidiano da empresa ao decorrer dos três semestres que o trabalho foi desenvolvido, o que possibilitou fazer um estudo de caso.

8.3 ESTUDO DE CASO

O produto de software avaliado destina-se à gestão empresarial e tem como principais módulos: controle financeiro, comercial e estoque.

Foram avaliadas características de qualidade externa, ou seja, a qualidade observada durante a execução do software.

O estudo de caso efetuado, permitiu observar o comportamento do software após alterações rotineiras. Por meio dos relatos efetuados pelos clientes ao suporte técnico, observou-se que comumente eram encontrados falhas e defeitos, e que uma das reclamações mais frequentes é o fato de uma correção acabar causando novos erros.

Foi possível perceber a necessidade de uma função de apoio dentro da empresa, atuando na revisão da eficácia e da qualidade do software, na coleta e análise de resultados de avaliação e sua disseminação dentro da organização com o uso de uma base de dados. Alguém para buscar conformidade à normas.

A NBR ISO/IEC 14598-2 fornece requisitos, recomendações e diretrizes para uma função de apoio responsável pela gestão da avaliação de produto de software e pelas tecnologias necessárias para a avaliação de produto de software. Essa norma auxilia as pessoas envolvidas em funções de garantia da qualidade e responsáveis pelo apoio à avaliação de produto de software.

Atuando como uma função de apoio, estudou-se normas de qualidade para produto de software e a partir da seleção de algumas delas, buscou-se meios de avaliar o software e perceber os pontos que devem ser melhorados ou acrescentados para manter um produto permanentemente com qualidade.

8.4 APLICAÇÃO DAS NORMAS DE QUALIDADE

Foi realizado um estudo detalhado das normas de qualidade para produto de software, com o objetivo de encontrar meios para avaliar um produto de software e apresentar algumas ações que pudessem ajudar na melhoria deste software.

Pode-se perceber que normas como CMM, CMMI, SPICE e MPS.BR, as mais conhecidas e comentadas em pesquisas e estudos científicos aplicam-se na avaliação e desenvolvimento de processos. Porém tinha-se um produto que já passou pela fase de desenvolvimento e encontra-se atualmente em manutenção, e vem sendo utilizado por vários clientes há um tempo considerável.

Aprofundaram-se ainda mais os estudos e encontrou-se normas que oferecem um melhor embasamento para esse tipo de avaliação:

- a) NBR ISO/IEC 9126-1: que define características e sub-características que devem estar presentes em um produto de software de qualidade;
- b) NBR ISO/IEC 14598-2: fornece detalhes sobre requisitos de planejamento e gestão, associados com a avaliação de produto de software;
- c) NBR ISO/IEC 14598-5: auxilia na avaliação dessas características e pode ser aplicada em todas as fases do ciclo de vida do software;
- d) NBR ISO/IEC 14598-6: define um modelo de documentação para esta avaliação.

A NBR ISO/IEC 9126-2 e a NBR ISO/IEC 9126-3 apresentam métricas para medir a qualidade interna e externa, conforme as características da NBR ISO/IEC 9126-1. Nesta avaliação não foram utilizadas essas métricas, pois se aplicam melhor quando o código-fonte do software é avaliado.

Depois de definidas as normas para serem utilizadas nesta avaliação, tinha-se então a necessidade de buscar uma quantidade maior de informações sobre as mesmas, fazer um estudo mais aprofundado, e assim, fazer uma análise do software.

8.5 ANÁLISE DO SOFTWARE CONFORME AS NORMAS DE QUALIDADE

Seguindo o planejamento de avaliação da NBR ISO/IEC 14598-2, depois de identificar as normas para avaliar a qualidade do produto de software e definir os objetivos da avaliação é necessário definir qual tecnologia de avaliação usar.

A tecnologia de avaliação define as técnicas, ferramentas, métricas, medidas e outras informações técnicas usadas para avaliação.

Um produto de software é resultado do trabalho de uma equipe e, a sua qualidade depende do desempenho desta equipe. Mudanças em busca de melhoria, são aplicadas apenas com o consentimento de todos.

Notou-se que o julgamento da equipe é muito importante para alcançar a qualidade do software, então, um questionário seria útil para detectar a opinião dos desenvolvedores sobre a qualidade de seu produto.

Um questionário foi elaborado por meio do modelo de qualidade definido na NBR ISO/IEC 9126-1 e aplicado com a equipe de desenvolvimento, com perguntas sobre funcionalidade, confiabilidade, eficiência, manutenibilidade e portabilidade.

Precisava-se observar o comportamento do software em relação à essas características, por meio de um teste com o produto em execução. Esse teste é chamado de teste caixa-preta ou teste funcional e permite verificar características como funcionalidade e manutenibilidade.

O teste caixa-preta é identificado na NBR ISO/IEC 14598-6 como uma técnica de avaliação que não se preocupa com os detalhes da construção e estrutura do software, não há conhecimento sobre a operação interna. O avaliador observa as funções que o software deve desempenhar.

O teste caixa-preta permite avaliar problemas como:

- a) funções incorretas ou omitidas;
- b) erros de interface;
- c) erros de comportamento ou desempenho;
- d) erros de iniciação e término.

No teste caixa-preta, a manutenibilidade foi a principal característica avaliada. Foram encontradas algumas falhas durante a execução e foram indicadas algumas sugestões para melhorar alguns aspectos, como por exemplo o *feedback* com o usuário, por meio de mensagens de fácil entendimento e o ajuste de relatórios para que todos obedecessem um mesmo padrão, entre outros.

Depois do teste finalizado foram encaminhadas à equipe de desenvolvimento solicitações de alterações no software, incluindo as sugestões e posteriormente foram realizados mais alguns testes verificando as alterações, então alguns resultados foram obtidos.

Como requisitos para avaliar a qualidade do produto de software foram utilizadas algumas das características de qualidade definidas na NBR ISO/IEC 9126-1.

A NBR ISO/IEC 14598-5 foi utilizada na aplicação dos conceitos descritos na NBR ISO/IEC 9126-1 e na implementação prática da avaliação, por meio de requisitos, recomendações e definição das atividades necessárias para projetar e executar as ações de avaliação.

As informações utilizadas na avaliação foram agrupadas em um pacote, que pode ser usado futuramente para avaliar outros softwares. Esse pacote é definido pela NBR ISO/IEC 14598-6 como Módulo de Avaliação.

8.6 MÓDULO DE AVALIAÇÃO EM CONFORMIDADE COM A NBR ISO/IEC 14598-6

Esta avaliação teve o intuito de melhorar a qualidade do software e conseqüentemente minimizar as reclamações, os problemas mais freqüentes, conquistar a satisfação dos clientes e também, apontar os pontos críticos e sugestões de melhorias.

Este módulo de avaliação foi elaborado com o objetivo de avaliar a qualidade externa de um produto de software utilizando normas de qualidade.

A qualidade externa se preocupa com a satisfação do cliente. O software é medido e avaliado enquanto é executado e testado num ambiente simulado e com dados simulados. Os testes têm o objetivo de encontrar a maioria dos defeitos e eliminá-los, entretanto alguns deles podem permanecer no software.

Para que uma avaliação seja executada de forma correta é necessário que se tenha documentado os requisitos de software e as necessidades explícitas bem descritas.

A avaliação pode ser realizada em níveis que permitem o uso de técnicas diferentes em cada um deles. Eles determinam também a importância de cada característica, definem a profundidade ou a extensão da avaliação e dos resultados a serem alcançados. Dessa forma, se têm diferentes níveis de confiança na qualidade do produto.

Na ABNT NBR ISO/IEC 14598-5 são definidos quatro níveis chamados de A, B, C e D. O nível A representa o nível mais alto, onde as técnicas de avaliação são mais rigorosas (levando em conta uma considerável quantidade de esforço e tempo). O nível D é o mais baixo, são usados métodos menos rigorosos.

Essa avaliação é de nível D, pois não foram utilizadas técnicas rigorosas e por ser a primeira avaliação realizada com o produto em questão, portanto é importante que posteriormente os demais níveis sejam avaliados.

Foram utilizadas duas técnicas de avaliação para medir a qualidade do software: questionário e teste caixa-preta.

O questionário é uma técnica de levantamento de dados e foi elaborado a partir das características de qualidade definidas na NBR ISO/IEC 9126-1 e aplicado com os desenvolvedores do software com o intuito de saber a opinião deles sobre a qualidade do software desenvolvido. As questões podem ser vistas no Apêndice A.

O teste caixa-preta teve a finalidade de encontrar falhas no software, providenciar as devidas correções e apontar as possíveis causas. Foram solicitadas em torno de 130 modificações no software onde, foi possível classificar 24 tipos de falhas, que encontram-se no Apêndice B.

Este módulo de avaliação permite encontrar os problemas nos produtos de software que são desenvolvidos por pequenas empresas, onde técnicas e ferramentas de engenharia de software são pouco utilizadas, e identificar as possíveis causas para estes problemas. Este módulo de avaliação pode ser aplicado à qualquer software desenvolvido nestas condições.

Para realizar a avaliação foi necessário ter acesso ao código executável do software, um banco de dados para teste e o sistema gerenciador de banco de dados.

8.7 RELATÓRIO DE AVALIAÇÃO

A partir do questionário e do teste caixa-preta aplicados, o estudo de caso realizado na empresa desenvolvedora do software e o contato diário com os clientes foi possível a obtenção de alguns resultados.

A Tabela 4 mostra a média dos percentuais positivos de cada característica avaliada no questionário aplicado à equipe de desenvolvimento.

Tabela 4. Avaliação do questionário

Característica	Sub-característica	Respostas positivas
Funcionalidade 60,0%	Adequação	75%
	Acurácia	50%
	Interoperabilidade	75%
	Segurança de Acesso	50%
	Conformidade à funcionalidade	50%
Confiabilidade 75%	Maturidade	100%
	Recuperabilidade	50%
Eficiência 25%	Comportamento em relação ao tempo	25%
Manutenibilidade 42,42%	Analisabilidade	25%
	Modificabilidade	36,36%
	Testabilidade	50%
	Comunicação e <i>Feedback</i>	58,33%
Portabilidade 56,25%	Capacidade para ser instalado	62,5%
	Capacidade para substituir	50%

Com esses resultados é possível definir junto à empresa desenvolvedora do software, as principais características de qualidade que seu produto deve alcançar e identificar os percentuais que gostariam de atingir e assim fazer um comparativo. Avaliações devem ser realizadas até que, seja alcançado o percentual desejado.

Depois de concluído o teste caixa-preta, alguns clientes tiveram sua versão atualizada, com as correções e modificações recomendadas. O nível de reclamações diminuiu satisfatoriamente.

Com o estudo de caso realizado pode-se perceber que muitas das alterações e correções efetuadas no sistema acabavam ocasionando novos problemas, e como os clientes muitas vezes tinham urgência em receber a nova versão com a correção da falha, não se tinha tempo o suficiente para fazer todos os testes necessários.

Os problemas observados nesta avaliação, também são causados pelo fato de algum módulo no software estar passando por algumas alterações e, conseqüentemente não estar pronto para ser instalado no cliente.

O módulo em alteração é enviado ao cliente quando este identifica uma falha no software que impede a realização de alguma atividade, então, o software deve ser atualizado imediatamente. Em conseqüência das alterações inacabadas, aparecem novos problemas e assim, novas reclamações e a insatisfação.

Percebeu-se a necessidade de ter uma versão estável e sem erros, pois só é possível trabalhar em função de melhorias no software quando existe essa versão estável, caso contrário, perde-se muito tempo corrigindo erros. Enquanto isso, nada é acrescentado ou melhorado.

Se os clientes estiverem utilizando uma versão estável, que não apresente erros, eles poderão trabalhar corretamente e sem problemas. Enquanto isso, os desenvolvedores poderão fazer melhorias no software e realizar todos os testes necessários após cada modificação.

As melhorias devem ser realizadas em conformidade com as normas de qualidade. A NBR ISO/IEC 14598-3 auxilia os desenvolvedores, por meio de requisitos e recomendações que permitem realizar uma avaliação de qualidade em paralelo com o desenvolvimento do software.

Com a utilização da norma, é possível identificar as características específicas a serem implementadas ou alteradas no software para atender aos requisitos de qualidade. A

NBR ISO/IEC 14598-3 auxilia também na tomada de decisões referentes à reprojeto, reengenharia ou implementações de alterações necessárias.

Dessa forma quando uma nova versão for instalada no cliente, continuará funcional e cada vez com mais qualidade. Conseqüentemente, os clientes estarão satisfeitos e a empresa desenvolvedora conquistará novos clientes e crescerá entre os concorrentes no mercado de software.

Uma função de apoio também tem o papel de identificar melhorias futuras, por isso, a seguir são apresentadas algumas recomendações que podem representar a solução para esses problemas, se acatadas pelas empresas desenvolvedoras de software.

8.7.1 Recomendações de melhorias

Os problemas encontrados nesta avaliação podem ser solucionados com a utilização de uma ferramenta de controle de versão, documentação do software e práticas de engenharia reversa.

Um controle de versão permite:

- a) controlar as alterações, sabendo quem foi o responsável e o que alterou;
- b) recuperar códigos de versões anteriores;
- c) examinar o histórico de alterações.

O controle de versão faz um registro de todas as alterações efetuadas no software e da evolução do projeto. Os desenvolvedores podem trabalhar paralelamente com os mesmos arquivos, sem que um sobrescreva o código do outro, evitando defeitos e perda de funcionalidade. Dessa forma sempre existirá uma versão estável, sem falhas e que pode ser instalada nos clientes.

Paralelamente a isso, é importante que a documentação do software seja atualizada, sempre que alterações forem realizadas no projeto.

Um software que não tem documentação torna-se dependente do programador. Se um dia aquele programador que conhece o software sai da empresa é muito difícil para os demais darem continuidade no projeto.

A documentação é um meio de comunicação entre a equipe de desenvolvimento, contendo informações importantes para aqueles que fazem ou venham a fazer manutenção no software.

Quando se trata de documentação de software, fala-se de documentos de projeto, relatórios de teste e validação, código-fonte e documentação de usuário. A avaliação de software apresentada na NBR ISO/IEC 14598-5 leva em consideração todos esses documentos que podem ser considerados como parte do produto de software.

Caso a empresa desenvolvedora do software desejar atualizar a documentação do software, mas com poucos recursos, pode-se utilizar uma abordagem de Engenharia Reversa chamada de *documentar quando tocar*. As partes que atualmente estão sofrendo modificações são documentadas. Ao longo do tempo, pode-se obter uma quantidade considerável de documentação útil e relevante.

A Engenharia Reversa tem o objetivo de criar versões de programas já existentes com mais qualidade e melhor manutenibilidade. Essa técnica pode ser aplicada em empresas desenvolvedoras de software, auxiliando na estruturação e documentação do software.

Nesta pesquisa pode-se constatar que essas técnicas recomendadas podem melhorar significativamente a qualidade do produto de software quando levadas em consideração e aplicadas corretamente, possibilitando a existência de uma versão estável, a realização de alterações de melhorias, a satisfação dos clientes e o reconhecimento da empresa desenvolvedora no mercado de software.

CONCLUSÃO

As organizações comerciais necessitam de sistemas de gestão empresarial para auxiliar no armazenamento e organização da grande quantidade de dados gerados diariamente. Esses sistemas oferecem relatórios gerenciais que são utilizados pelos gerentes para tomar decisões comerciais e financeiras. Portanto, as informações oferecidas pelos softwares precisam ser confiáveis.

O mercado de software está cada vez mais competitivo e as empresas desenvolvedoras buscam meios de oferecer produtos de qualidade.

A Engenharia de Software oferece padrões, técnicas e ferramentas para serem aplicadas durante o ciclo de vida do software para que sejam desenvolvidos produtos de qualidade. Porém, em muitas empresas essas práticas não são aplicadas e um dos motivos é a falta de recursos.

Com o auxílio das normas de qualidade para produto de software, foi avaliado um sistema para gestão empresarial de uma dessas empresas. A avaliação realizada no software possibilitou perceber a necessidade de se utilizar as práticas de Engenharia de Software como auxílio no desenvolvimento de software. Ficou claro que a ausência dessas práticas afetam a qualidade final do produto de software.

As pequenas empresas de desenvolvimento de software não possuem certificação de qualidade devido ao alto custo.

Esta pesquisa pode auxiliar empresas que enfrentam problemas na manutenção do software e têm dificuldades para satisfazer os clientes, por meio da avaliação da qualidade de seus produtos. Se os resultados da avaliação não forem satisfatórios, as recomendações aqui mencionadas podem ser aplicadas para atingir uma melhoria considerável na qualidade do produto de software.

Nesta pesquisa concluiu-se que as empresas podem oferecer aos seus clientes softwares de qualidade mesmo sem muitos recursos financeiros, basta fazer uso das práticas de Engenharia de Software e estar sempre avaliando a qualidade de seus produtos e buscando melhorias.

Encontraram-se algumas dificuldades devido à falta de documentação do software avaliado, o que dificultou a avaliação e também, a carência de materiais didáticos para auxiliar no entendimento sobre avaliação de produto de software. A primeira dificuldade foi superada com o convívio diário dentro da empresa desenvolvedora do software e o contato direto com o produto e a segunda, quando se conseguiu as normas de qualidade para produto de software na Universidade Federal de Santa Catarina.

Contudo, os objetivos desta pesquisa foram atingidos, os conceitos estudados permitiram a realização de um estudo de caso e a partir daí, foi realizado uma avaliação de qualidade de um produto de software, foram estudadas e recomendadas as técnicas em conformidade com a Engenharia de Software que possibilitam a obtenção da qualidade do software.

Para finalizar, tem-se algumas sugestões para trabalhos futuros a fim de dar continuidade nesta pesquisa:

- a) avaliação da qualidade externa do produto de software utilizando as métricas apresentadas na NBR ISO/IEC 9126-2;
- b) avaliação da qualidade interna do produto de software, por meio da avaliação dos processos e dos produtos intermediários, utilizando as métricas apresentadas na NBR ISO/IEC 9126-3;
- c) avaliação da qualidade em uso do produto de software utilizando as métricas apresentadas na NBR ISO/IEC 9126-4.

- d) avaliação da usabilidade diretamente com usuários. Pode-se reunir um grupo de pessoas que não conhecem o software e instruí-los para explorar o software, sem fornecer informações de uso;
- e) continuidade nesta avaliação a partir dos próximos níveis, utilizando novas técnicas de avaliação;
- f) aplicação das recomendações aqui mencionadas e avaliação dos resultados obtidos;
- g) desenvolvimento de uma estrutura de documentação que possa ser usada como base por desenvolvedores de software.

BIBLIOGRAFIA

AGUAYO, Maria Teresa Villalobos; GUERRA, Ana Cervigni; THIENNE, Regina Maria. **Processo de avaliação da manutenibilidade de produtos de software**. Disponível em: <http://www.angelicatoffano.pro.br/upload_arquivos/pt/A08_3_artigo14730.pdf>. Acesso em: 15 set. 2009.

ALBUQUERQUE, Carlos Antônio Menezes de. **Qualidade ágil de software**. 2005. 157 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Pernambuco, Recife. Disponível em: < http://www.btd.ufpe.br/tedeSimplificado//tde_busca/arquivo.php?codArquivo=872 >. Acesso em: 10 set. 2009.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: informação e documentação: referências: elaboração. Rio de Janeiro, 2002.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14598 -2:2003**: engenharia de software: avaliação de produto: parte 2: planejamento e gestão. Rio de Janeiro, 2003.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14598 -3:2003**: engenharia de software: avaliação de produto: parte 3: processo para desenvolvedores. Rio de Janeiro, 2003.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14598 -4:2003**: engenharia de software: avaliação de produto: parte 4: processo para adquirentes. Rio de Janeiro, 2003.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14598 -5:2001**: tecnologia de informação: avaliação de produto de software: parte 5: processo para avaliadores. Rio de Janeiro, 2001.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14598 -6:2004**: engenharia de software: avaliação de produto: parte 6: documentação de módulos de avaliação. Rio de Janeiro, 2004.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9126 -1:2003**: engenharia de software: qualidade de produto de software: parte 1: modelo de qualidade. Rio de Janeiro, 2003.

CANHOTA JUNIOR, Antonio Jorge Sapage da et al. **Engenharia reversa**. 2005. Disponível em: <http://www.ic.uff.br/~otton/graduacao/informatica/apresentacoes/eng_reversa.pdf>. Acesso em: 07 ago. 2010.

CAPOVILLA, Izilda Gomes Garcez. **Elementos intrínsecos do software e sua influência na qualidade do processo de desenvolvimento**. 1999. 108 f. Dissertação (Mestrado em Qualidade) – Universidade Estadual de Campinas, Campinas. Disponível em: <<http://libdigi.unicamp.br/document/?code=000192483>>. Acesso em: 10 set. 2009.

COLENCI NETO, Alfredo. **Proposta de um modelo de referência para desenvolvimento de software com foco na certificação do MPS.BR**. 2008. 179 f. Tese (Doutorado em Engenharia de Produção) – Universidade de São Paulo, São Carlos. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18140/tde-29012009-093822/>>. Acesso em: 18 set. 2009.

COLLINS-SUSSMAN, Ben; FITZPATRICK, Brian W.; PILATO, C. Michael. **Controle de versão com Subversion: Para Subversion 1.4 (Compilado da revisão 320)**. 2007. Disponível em: <svnbook-pt-br.googlecode.com/files/svn-book-r320.pdf>. Acesso em: 28 out. 2010.

CORRÊA, Tiago Nonoai Flores. **Estudo e planejamento de manuais de usuário para Softwares**. 2007. 53 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Federal de Santa Maria, Santa Maria. Disponível em: <[www-inf.ufsm.br/bdtg/arquivo.php?id=65&download=1](http://www.inf.ufsm.br/bdtg/arquivo.php?id=65&download=1)>. Acesso em: 27/10/2010.

DIAS, André Felipe. **Conceitos básicos de controle de versão de software: centralizado e distribuído**. 2009. Disponível em: <http://www.pronus.eng.br/artigos_tutoriais/gerencia_configuracao/conceitos_basicos_controle_versao_centralizado_e_distribuido.php?pagNum=1>. Acesso em: 28 out. 2010.

FALBO, Ricardo de Almeida. **Engenharia de software: notas de aula**. Disponível em: <<http://www.pdf-search-engine.com/engenharia-de-software-html-www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/NotasDeAula.html>>. Acesso em: 25 out. 2009.

FERRARI, Sandra. **Proposta de metodologia para controle de qualidade em uma fábrica de software**. 2007. 302 f. Tese (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <http://www.tede.ufsc.br/tedesimplificado//tde_busca/arquivo.php?codArquivo=77>. Acesso em: 15 out. 2009.

GOMES, Nelma da Silva. **Qualidade de software: uma necessidade**. Petrópolis. 2000. Disponível em: <www.fazenda.gov.br/ucp/pnafe/cst/.../Qualidade_de_Soft.pdf>. Acesso em: 15 out. 2009.

KAPPEL, Alexandre Moreira. **Uma abordagem para o gerenciamento do projeto de sistemas de informação na web com características sazonais**. 2006. 88 f. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal do Rio Grande do Sul, Porto Alegre. Disponível em: < <http://www.lume.ufrgs.br/handle/10183/5712> >. Acesso em: 10 set. 2009.

KOSCIANSKI, André; SOARES, Michel dos Santos. **Qualidade de software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. São Paulo: Novatec, 2006.

LAUDON, Kenneth C.; LAUDON, Jane Price. **Sistemas de informação: com Internet**. 4. ed. Rio de Janeiro: LTC, 1999.

PLEEGER, Shari Lawrence. **Engenharia de software: teoria e prática**. São Paulo: Pearson, 2004.

PRESSMAN, Roger S. **Engenharia de software**. 6. ed São Paulo: McGraw-Hill, 2006.

SALLES, Antonio Augusto Martinelli. **Sistemas de informações e a construção do conhecimento empresarial: um estudo a partir da ótica da ciência da informação**. 2007. 98 f. Dissertação (Mestrado em Ciência da Informação) – Pontifícia Universidade Católica de Campinas, Campinas. Disponível em: <http://www.bibliotecadigital.puc-campinas.edu.br/tde_busca/arquivo.php?codArquivo=342>. Acesso em: 10 set. 2009.

SANT'ANA, Mary Lucy. **Uma proposta para qualidade de software por meio da aplicação integrada do modelo CMMI e da norma NBR 13596**. 2002. 124 f. Dissertação (Mestrado em Engenharia Mecânica/gestão da Qualidade Total) - Universidade Estadual de Campinas, Campinas. Disponível em: <<http://libdigi.unicamp.br/document/?code=vtls000252451>>. Acesso em: 10 set. 2009.

SOCIEDADE SOFTEX. **MPS.BR - Melhoria de Processo de Software Brasileiro: guia geral**. 2009. Disponível em: <http://www.softex.br/mpsbr/_guias/default.asp>. Acesso em: 15 set. 2009.

SOCIEDADE SOFTEX. **MPS.BR - Melhoria de Processo de Software Brasileiro: guia de aquisição**. 2009. Disponível em: <http://www.softex.br/mpsbr/_guias/default.asp>. Acesso em: 15 set. 2009.

SOCIEDADE SOFTEX. **MPS.BR - Melhoria de Processo de Software Brasileiro: guia de avaliação**. 2009. Disponível em: <http://www.softex.br/mpsbr/_guias/default.asp>. Acesso em: 15 set. 2009.

SODRÉ, Cibele Cristina Pelizer. **Norma ISO/IEC 9126**: avaliação de qualidade de produtos de software. 2006. 52 f. Trabalho de Conclusão de Curso (Graduação em Ciência de Computação) – Universidade Estadual de Londrina, Londrina. Disponível em: < <http://www2.dc.uel.br/nourau/document/?view=625> >. Acesso em: 10 set. 2009.

SOMMERVILLE, Ian. **Engenharia de software**. São Paulo: Addison-Wesley, 2003. 592 p.

SOUZA, Sérgio Cozzetti Bertoldi de et al. Documentação Essencial para Manutenção de Software II. In: JORNADAS IBEROAMERICANAS DE INGENIERÍA DEL SOFTWARE E INGENIERÍA DEL CONOCIMIENTO - JIISIC 2004, 1., 2004, Brasília. 2004. Madri.

TURBAN, Efraim; RAINER, R. Kelly; POTTER, Richard E.. **Introdução a sistemas de informação**: uma abordagem gerencial. Rio de Janeiro: Elsevier, 2007.

XAVIER, Juliana Moura Cavalcanti. **Proevaluator**: uma ferramenta para avaliação de processos de software. 2007. 123 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Pernambuco, Recife. Disponível em: < http://www.bdtf.ufpe.br/tede/Simplificado//tde_busca/arquivo.php?codArquivo=3754 >. Acesso em: 10 set. 2009.

BIBLIOGRAFIA COMPLEMENTAR

BARRUECO, Fernando Mauro; PERROTTI, Paulo Salvador Ribeiro; LERNER, Walter. **Empresas familiares: aspectos jurídicos e estratégias para uma boa gestão.** São Paulo: IOB Thomson, 2007.

BRASIL. Secretaria de Política de Informática e Automação. **Tecnologia da informação: Programa Brasileiro da Qualidade e Produtividade em Software.** 5. ed. rev. e ampl Brasília: Ministério da Ciência e Tecnologia, 2008.

CORREA, Juliana. **Adoção, seleção e implementação de um ERP livre.** 2008. 173 f. Dissertação (Mestrado em Engenharia) - Universidade de São Paulo, São Paulo, 2008. Disponível em: <<http://www.pro.poli.usp.br/publicacoes/dissertacoes/adocao-selecao-e-implementacao-de-um-erp-livre>>. Acesso em: 23 set. 2009.

DINIZ, Alfraino de Souza. **PROREC - Um guia facilitador para a implantação dos processos de gestão de requisitos.** 2007. 92 p. Dissertação (Mestrado em Ciências de Computação e Matemática) – Universidade de São Paulo, São Carlos. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-31032008-161043/>>. Acesso em: 18 set. 2009.

FABRIN, Alex Onacli Moreira. **Fundamentos para formulação de estratégias competitivas em organizações de desenvolvimento de software.** 2005. 117 f. Dissertação (Mestrado em Administração) – Universidade do Estado de Santa Catarina, Florianópolis. Disponível em: <http://www.tede.udesc.br/tde_busca/arquivo.php?codArquivo=359>. Acesso em: 10 set. 2009.

GUERRA, Ana Cervigni; COLOMBO, Regina Maria Thienne. **Tecnologia da informação: qualidade de produto de software.** Brasília: Ministério da Ciência e Tecnologia, 2009. 429 p. GUSTAFSON, David A. **Teoria e problemas de engenharia de software.** Porto Alegre: Bookman, 2003.

INTHURN, Cândida. **Qualidade & teste de software:** engenharia de software, qualidade de software, qualidade de produtos de software, teste de software, formalização do processo de teste, aplicação prática dos testes. Florianópolis: Visual Books, 2001.

ISO. International Standard. **ISO/IEC 14598-5: Information technology — Software product evaluation — Part 5: Process for evaluators.** 1998. Disponível em: <http://webstore.iec.ch/preview/info_isoiec14598-5%7Bed1.0%7Den.pdf>. Acesso em: 20 set. 2009.

ISO. International Standard. **ISO/IEC 9126-1: Software engineering - product quality - part 1: quality model.** 2001. Disponível em: <http://webstore.iec.ch/preview/info_isoiec9126-1%7Bed1.0%7Den.pdf>. Acesso em: 02 nov. 2009.

OLIVEIRA, Otávio J. **Gestão empresarial: sistemas e ferramentas.** São Paulo: Atlas, 2007.

RIOS, Emerson; MOREIRA FILHO, Trayahú R. . **Teste de software.** 2. ed. rev. e ampl Rio de Janeiro: Alta Books, 2006.

ROCHA, Ana Regina Cavalcanti da; MALDONADO, José Carlos; WEBER, Kival Chaves. **Qualidade de software: teoria e prática.** São Paulo: Prentice Hall, 2001.

RODRIGUEZ, M. V. R. . **Gestão empresarial em organizações aprendizes: a arte de gerir mudanças.** Rio de Janeiro: Qualitymark, 2007.

SANTANA, André Felipe Lemos. **Problemas em iniciativas de melhoria de processos de software sob a ótica de uma teoria de intervenção.** 2007. 176 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Pernambuco, Recife. Disponível em: <http://www.di.ufpe.br/~hermano/download/dissertacoes/AF_Santana-Dissertacao_Vfinal.pdf>. Acesso em: 10 set. 2009.

SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 22., 2008, Campinas, SP; MENDONÇA NETO, Manoel Gomes de. . **Anais ...** Campinas, SP: Sociedade Brasileira de Computação, 2008.

SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE, 7, 2008, Florianópolis, SC . UNIVERSIDADE DO VALE DO ITAJAÍ UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO. . **Anais ...** Florianópolis; Recife: UNIVALI; UFRPE, 2008.

SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE, 8., 2009, Minas Gerais, MG UNIVERSIDADE FEDERAL DE OURO PRETO PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS. . **Anais ...** Minas Gerais: Sociedade Brasileira de Computação, 2009.

SOUZA, Cesar Alexandre de; SZAFIR-GOLDSTEIN, Cláudia. **Tecnologia da informação aplicada à gestão empresarial: um modelo para a empresa digital.** Disponível em: <http://www.cyta.com.ar/elearn/syma/textos/empresa_digital_archivos/modelo_empresa_digital.pdf>. Acesso em: 20 set. 2009.

TAKEUCHI, Hirotaka; NONAKA, Ikujiro. . **Gestão do conhecimento.** Porto Alegre: Bookman, 2008.

WORKSHOP ON FORMAL METHODS 8: 2001 out. 1-2: Rio de Janeiro; OLIVEIRA, káthia Marçal de; WEBER, Kival Chaves (Editor). **Anais ...** Rio de Janeiro: CNPq, 2001. Não paginado.

APÊNDICE A – QUESTIONÁRIO PARA AVALIAÇÃO DO SOFTWARE

Este questionário tem o objetivo de coletar informações referentes ao sistema para avaliação de qualidade.

1. O sistema é adequado as necessidades dos clientes?

Adequação as necessidades dos clientes	Freq.	%
Não	0	0,00
Em partes	1	25,00
Sim	3	75,00
Total	4	100,00

2. Gera resultados corretos ou conforme solicitados pelos clientes?

Resultados corretos ou conforme solicitados	Freq.	%
Não	0	0,00
Sim	2	50,00
Em partes	2	50,00
Total	4	100,00

3. É capaz de interagir com outros sistemas utilizados pelo cliente?

Capacidade de interagir com outros sistemas	Freq.	%
Não	0	0,00
Em partes	1	25,00
Sim	3	75,00
Total	4	100,00

4. Evita acesso não autorizado a programas e dados?

Evita acesso não autorizado	Freq.	%
Não	0	0,00
Sim	2	50,00
Em partes	2	50,00
Total	4	100,00

5. Está de acordo com as normas, leis etc.?

Conformidade com as	Freq.	%
Não	0	0,00
Sim	2	50,00
Em partes	2	50,00
Total	4	100,00

6. Com que frequência apresenta falhas?

Frequência das falhas apresenta	Freq.	%
De 6 a 10 vezes por semana	0	0,00
Mais de 10 vezes por semana	0	0,00
De 0 a 5 vezes por semana	4	100,00
Total	4	100,00

7. É capaz de recuperar dados após uma falha?

Capacidade de recuperar dados	Freq.	%
Não	1	25,00
Em partes	1	25,00
Sim	2	50,00
Total	4	100,00

8. Como é o tempo de resposta e de processamento?

Tempo de resposta e de processamento	Freq.	%
Regular	0	0,00
Ruim	0	0,00
Ótimo	1	25,00
Bom	3	75,00
Total	4	100,00

9. Quando ocorre falha no sistema é fácil encontrar a causa no código fonte?

Fácil encontrar a causa da falha no código fonte	Freq.	%
Não	0	0,00
Em partes	0	0,00
Sim	4	100,00
Total	4	100,00

10. O usuário consegue identificar a operação específica que causou a falha?

Usuário consegue identificar a operação que causa da falha	Freq.	%
Sim	0	0,00
Não	1	25,00
Em partes	3	75,00
Total	4	100,00

11. O usuário consegue achar facilmente a causa da falha?

Usuário consegue achar a causa da falha	Freq.	%
Sim	1	25,00
Em partes	1	25,00
Não	2	50,00
Total	4	100,00

12. O suporte técnico consegue achar facilmente a causa da falha?

Suporte técnico consegue achar a causa da falha	Freq.	%
Sim	0	0,00
Não	1	25,00
Em partes	3	75,00
Total	4	100,00

13. O usuário consegue identificar a causa de uma falha com as mensagens de erro apresentadas na tela?

Usuário identificar causa da falha com mensagens de erro	Freq.	%
Sim	0	0,00
Não	1	25,00
Em partes	3	75,00
Total	4	100,00

14. É fácil modificar e remover defeitos?

Fácil modificar e remover defeitos	Freq.	%
Não	0	0,00
Sim	2	50,00
Em partes	2	50,00
Total	4	100,00

15. O problema do usuário é solucionado satisfatoriamente em tempo aceitável?

Problema solucionado satisfatoriamente em tempo aceitável	Freq.	%
Não	0	0,00
Sim	2	50,00
Em partes	2	50,00
Total	4	100,00

16. A equipe de desenvolvimento consegue facilmente mudar o software para resolver o problema de falha?

Desenvolvimento consegue mudar o software para resolver a falha	Freq.	%
Não	1	25,00
Em partes	1	25,00
Sim	2	50,00
Total	4	100,00

17. A equipe de desenvolvimento consegue facilmente mudar o software para resolver problemas?

Desenvolvimento mudar o software para resolver problemas	Freq.	%
Não	0	0,00
Sim	2	50,00
Em partes	2	50,00
Total	4	100,00

18. A falha é satisfatoriamente resolvida?

Falha satisfatoriamente resolvida	Freq.	%
Não	0	0,00
Sim	2	50,00
Em partes	2	50,00
Total	4	100,00

19. Na correção de uma falha, são necessárias grandes alterações no código fonte?

Correção de uma falha, necessita grandes alterações no código fonte	Freq.	%
Sim	0	0,00
Não	2	50,00
Em partes	2	50,00
Total	4	100,00

20. A documentação é alterada em decorrência da modificação?

Documentação alterada em decorrência da modificação	Freq.	%
Sim	0	0,00
Não	0	0,00
Em partes	0	0,00
Não possui documentação	4	100,00
Total	4	100,00

21. Modificação foi efetivamente registrada no log de modificações?

Modificação registrada no log de modificações	Freq.	%
Sim	0	0,00
Em partes	1	25,00
Não	3	75,00
Total	4	100,00

22. Há grandes riscos de *bugs* quando se faz alterações?

Riscos de <i>bugs</i> nas alterações	Freq.	%
Não	1	25,00
Em partes	1	25,00
Sim	2	50,00
Total	4	100,00

23. Caso a falha seja satisfatoriamente resolvida, são propagadas novas falhas decorrentes da modificação?

Novas falhas decorrentes da modificação	Freq.	%
Sim	0	0,00
Não	0	0,00
Em partes	4	100,00
Total	4	100,00

24. A empresa desenvolvedora consegue corrigir ou contornar as falhas causadas pelos efeitos colaterais de manutenção?

Contornar falhas causadas pelos efeitos colaterais	Freq.	%
Não	0	0,00
Em partes	0	0,00
Sim	4	100,00
Total	4	100,00

25. Após a manutenção, o usuário consegue operar o sistema de software sem falhas?

Após a manutenção, sistema de software pode operar sem falhas	Freq.	%
Não	0	0,00
Em partes	1	25,00
Sim	3	75,00
Total	4	100,00

26. É fácil testar quando se faz alterações?

Fácil testar quando se faz alterações	Freq.	%
Em partes	0	0,00
Não	1	25,00
Sim	3	75,00
Total	4	100,00

27. Consegue-se facilmente realizar teste caixa-preta sem preparação adicional (treinamento)?

Realizar teste caixa-preta sem preparação adicional	Freq.	%
Sim	0	0,00
Não	2	50,00
Em partes	2	50,00
Total	4	100,00

28. A empresa desenvolvedora consegue facilmente realizar teste caixa-preta e determinar se o software está pronto para uso ou não?

A empresa desenvolvedora realizar teste caixa-preta	Freq.	%
Não	0	0,00
Em partes	1	25,00
Sim	3	75,00
Total	4	100,00

29. A falha foi corrigida, mas foi conforme a solicitação do usuário e atendeu sua solicitação?

Falha foi corrigida conforme solicitação do usuário	Freq.	%
Não	0	0,00
Sim	2	50,00
Em partes	2	50,00
Total	4	100,00

30. A descrição da falha é enviada a equipe de desenvolvimento?

Descrição da falha enviada a equipe de desenvolvimento	Freq.	%
Não	0	0,00
Em partes	1	25,00
Sim	3	75,00
Total	4	100,00

31. Uma explicação satisfatória da causa da falha é enviada ao cliente?

Explicação satisfatória da causa da falha enviada ao cliente	Freq.	%
Sim	0	0,00
Em partes	1	25,00
Não	3	75,00
Total	4	100,00

32. Uma versão corrigida é enviada ao cliente?

Versão corrigida enviada ao cliente	Freq.	%
Não	0	0,00
Em partes	0	0,00
Sim	4	100,00
Total	4	100,00

33. É fácil instalar em outros ambientes?

Fácil instalar em outros ambientes	Freq.	%
Em partes	0	0,00
Não	1	25,00
Sim	3	75,00
Total	4	100,00

34. É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no software considerado?

Fácil adaptar a outros ambientes	Freq.	%
Não	1	25,00
Em partes	1	25,00
Sim	2	50,00
Total	4	100,00

35. É fácil substituir por outro software?

Fácil substituir por outro software	Freq.	%
Não	1	25,00
Em partes	1	25,00
Sim	2	50,00
Total	4	100,00

APÊNDICE B – RELAÇÃO DAS FALHAS ENCONTRADAS DURANTE O TESTE CAIXA-PRETA

O teste caixa-preta proporcionou identificar diferentes tipos de falhas:

- a) erros de ortografia e gramática;
- b) erros de *feedback* ao usuário: mensagens que o usuário não conseguia compreender;
- c) relatórios que não seguem o padrão;
- d) relatórios com colunas sobrepondo outras;
- e) relatórios ultrapassando o limite da página, principalmente quando existe a possibilidade de impressão em formulário razão;
- f) campos sem formatação correta;
- g) cadastros permitindo colocar uma data maior que a data atual;
- h) nos cadastros após ocorrer um erro o botão para gravar não era desabilitado, o que permitia gravar diversas vezes, duplicando o cadastro;
- i) informações errôneas (Cadastro de Cliente: na relação de aniversariantes do mês corrente o contato estava buscando errado);
- j) telas diferentes porém com funções semelhantes (por exemplo as duas telas de venda) não são corrigidas ou alteradas simultaneamente;
- k) campos sem limite de caracteres, podendo ocasionar erros;
- l) falta de filtros e verificações. Ex.: relatório de fornecedores: imprimia nos relatórios fornecedores com data de exclusão;
- m) erros de foco;

- n) alterações feitas no banco de dados sem fazer as devidas alterações no sistema e vice-versa (por exemplo a diferença de colunas nas tabelas, nome de variáveis modificadas no banco e no executável não);
- o) campos obrigatórios sem identificação de campo obrigatório;
- p) verificações incompletas no código;
- q) aceitação de datas fora do período aceitável, o que ocasiona erros. Ex.:
01/03/0010;
- r) erros de escrita no código fonte (por exemplo em um select que busca alguma informação);
- s) campos não nulos são criados e não são atribuídos valores a eles;
- t) mau posicionamento dos campos no relatórios e impressões, falta de alinhamento ;
- u) alteração de layout incompleta;
- v) painel sobrepondo outro nas telas;
- w) referências a alguma coisa que não existe ou não foi criado;
- x) correção de um erro ocasiona outro;
- y) falta de testes no desenvolvimento, alguns erros voltam exatamente iguais.

APÊNDICE C – ARTIGO

Avaliação de Produto de Software para Gestão Empresarial com Base nas Normas de QualidadeJuliana Brognoli Damiani¹, Ana Cláudia Garcia Barbosa²¹ Acadêmica do curso de Ciência da Computação² Professora do curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC – Brasil

jujudamiani@hotmail.com, agb@unesc.net

Resumo. Este trabalho propõe a melhoria da qualidade dos produtos de software desenvolvidos nas pequenas e médias empresas de desenvolvimento de software. Baseada nos conceitos da Engenharia de Software e nas normas ISO, foi realizada uma avaliação de qualidade de um produto de software desenvolvido por uma pequena empresa da cidade de Criciúma – SC. O objetivo desta pesquisa foi proposto no Trabalho de Conclusão de Curso apresentado para obtenção do Grau de Bacharel em Ciência da Computação da Universidade do Extremo Sul Catarinense.

Palavras-chave: Engenharia de Software; Qualidade de Produto de Software; Avaliação de Produto de Software; Normas de Qualidade de Software.

1. Introdução

A Engenharia de software surgiu com o intuito de melhorar a qualidade dos produtos de software e aumentar a produtividade no seu processo de desenvolvimento.

A preocupação com a qualidade do produto apresentado ao cliente deve começar no processo de software, porém, empresas que não desenvolvem seus produtos com esses critérios, podem sofrer as conseqüências quando o produto estiver em operação. O produto de software pode ser bom, mas não atender as necessidades dos clientes, neste caso há demora em atender os requisitos pós desenvolvimento.

Um software de qualidade na visão do usuário é aquele fácil de usar, eficiente e confiável. Para um desenvolvedor o software precisa ser fácil de manter, enquanto que para o cliente o software deve agregar valor ao seu negócio, ou seja, qualidade em uso (FALBO, 2005).

Com o propósito de avaliar a qualidade dos softwares foram criadas normas de qualidade para produto de software, como a ISO/IEC 14598 e ISO/IEC 9126.

Foi realizada uma avaliação de qualidade em um sistema para gestão empresarial, com o objetivo de minimizar os erros do software entregue aos clientes, por meio da utilização das normas de qualidade para produto de software.

Diante das reclamações efetuadas constantemente pelos clientes, relativas a erros ou alterações que não satisfaziam suas necessidades, buscou-se meios para minimizar os problemas e conquistar a satisfação dos clientes.

2. Engenharia de Software

A Engenharia de Software surgiu com a necessidade de organização e padronização no desenvolvimento de software. Juntamente com ela surgiram os padrões de qualidade e o conceito de qualidade de software. Essa tecnologia busca a otimização na construção do software, a facilidade no desenvolvimento e permite oferecer um produto final acessível e de baixo custo (SODRÉ, 2006).

Segundo Sommerville (2003) a Engenharia de Software auxilia na melhoria da qualidade, possibilitando como resultado um produto confiável e eficiente. Permite a escolha do método mais apropriado para um conjunto de circunstâncias, e da abordagem mais eficaz que os engenheiros de software devem adotar para produzir software de qualidade.

É de extrema importância que a qualidade do produto seja buscada ao longo de todo o ciclo de vida do software. Porém muitas vezes as empresas desenvolvedoras só percebem essa necessidade, quando o software já está em uso e apresentando falhas freqüentemente.

3. Qualidade de Produto de Software

A maioria das empresas que desenvolvem algum produto ou serviço, desejam atingir um alto nível de qualidade, porém quando se fala em produto de software é mais complexo apresentar definições de qualidade (SOMMERVILLE, 2003).

Um mesmo software quando analisado por pessoas diferentes pode apresentar diferentes características de qualidade e ambientes operacionais (NBR ISO/IEC 9126-1, 2003).

Os usuários nem sempre conseguem expressar suas reais necessidades, logo, é praticamente impossível atingir a qualidade perfeita e definir completamente os requisitos de qualidade antes do início do projeto. Atributos de software especificados por um usuário final, durante a fase de análise de requisitos, podem não atendê-lo na fase de operação (NBR ISO/IEC 9126-1, 2003).

Para atingir a qualidade necessária e suficiente para cada contexto de uso especificado quando o produto for entregue e utilizado pelos usuários, é necessário entender detalhadamente as reais necessidades dos usuários (NBR ISO/IEC 9126-1, 2003).

A garantia da qualidade de software assegura que os processos são estabelecidos e melhorados constantemente, satisfazendo os requisitos adequados ao uso. Conseqüentemente reduz o número de defeitos, cria mecanismos que controlem o desenvolvimento, cumprindo prazos e custos, reduz o re-trabalho e proporciona o uso (ALBUQUERQUE, 2005).

A competitividade está cada vez maior, cada vez mais os clientes exigem melhor qualidade nos produtos. O diferencial tecnológico e a qualidade são pré-requisitos para vencer a competição e ganhar a preferência do cliente.

4. Avaliação de Produto de Software

Foi realizado um estudo detalhado das normas de qualidade para produto de software, com o objetivo de encontrar meios para avaliar um produto de software e apresentar algumas ações que pudessem ajudar na melhoria deste software.

Pode-se perceber que normas como CMM, CMMI, SPICE e MPS.BR, as mais conhecidas e comentadas em pesquisas e estudos científicos aplicam-se na avaliação e desenvolvimento de processos. Porém tinha-se um produto que já passou pela fase de desenvolvimento e encontra-se atualmente, na em manutenção e vem sendo utilizado por vários clientes há um tempo considerável.

Aprofundaram-se ainda mais os estudos e encontrou-se normas que oferecem um melhor embasamento para esse tipo de avaliação:

- a) NBR ISO/IEC 9126-1: que define características e sub-características que devem estar presentes em um produto de software de qualidade;
- b) NBR ISO/IEC 14598-2: fornece detalhes sobre requisitos de planejamento e gestão, associados com a avaliação de produto de software;
- c) NBR ISO/IEC 14598-5: auxilia na avaliação dessas características e pode ser aplicada em todas as fases do ciclo de vida do software;
- d) NBR ISO/IEC 14598-6: define um modelo de documentação para esta avaliação.

A partir de um questionário aplicado com a equipe de desenvolvimento, do teste caixa-preta, um estudo de caso realizado na empresa desenvolvedora do software e o contato diário com os clientes foi possível a obtenção de alguns resultados.

O questionário é uma técnica de levantamento de dados e foi elaborado a partir das características de qualidade definidas na NBR ISO/IEC 9126-1 e aplicado com os desenvolvedores do software com o intuito de saber a opinião deles sobre a qualidade do software desenvolvido.

Com esses resultados do questionário é possível definir junto à empresa desenvolvedora do software, as principais características de qualidade que seu produto deve alcançar e identificar os percentuais que gostariam de atingir e assim fazer um comparativo. Avaliações devem ser realizadas até que, seja alcançado o percentual desejado.

O teste caixa-preta teve a finalidade de encontrar falhas no software, providenciar as devidas correções e apontar as possíveis causas. Foram solicitadas em torno de 130 modificações no software onde, foi possível classificar 24 tipos de falhas, que encontram-se no Apêndice B.

Depois de concluído o teste caixa-preta, alguns clientes tiveram sua versão atualizada, com as correções e modificações recomendadas. O nível de reclamações diminuiu satisfatoriamente.

Com o estudo de caso realizado pode-se perceber que muitas das alterações e correções efetuadas no sistema acabavam ocasionando novos problemas, e como os clientes muitas vezes tinham urgência em receber a nova versão com a correção da falha, não se tinha tempo o suficiente para fazer todos os testes necessários.

Os problemas observados nesta avaliação, também são causados pelo fato de algum módulo no software estar passando por algumas alterações e, conseqüentemente não estar pronto para ser instalado no cliente.

Percebeu-se a necessidade de ter uma versão estável e sem erros, pois só é possível trabalhar em função de melhorias no software quando existe essa versão estável, caso contrário, perde-se muito tempo corrigindo erros. Enquanto isso, nada é acrescentado ou melhorado.

Se os clientes estiverem utilizando uma versão estável, que não apresente erros, eles poderão trabalhar corretamente e sem problemas. Enquanto isso, os desenvolvedores poderão fazer melhorias no software e realizar todos os testes necessários após cada modificação.

Dessa forma quando uma nova versão for instalada no cliente, continuará funcional e cada vez com mais qualidade. Conseqüentemente, os clientes estarão satisfeitos e a empresa

desenvolvedora conquistará novos clientes e crescerá entre os concorrentes no mercado de software.

Os problemas encontrados nesta avaliação podem ser solucionados com a utilização de uma ferramenta de controle de versão, documentação do software e práticas de engenharia reversa.

5. Considerações Finais

As organizações comerciais necessitam de sistemas de gestão empresarial para auxiliar no armazenamento e organização da grande quantidade de dados gerados diariamente. Esses sistemas oferecem relatórios gerenciais que são utilizados pelos gerentes para tomar decisões comerciais e financeiras. Portanto, as informações oferecidas pelos softwares precisam ser confiáveis.

O mercado de software está cada vez mais competitivo e as empresas desenvolvedoras buscam meios de oferecer produtos de qualidade.

A Engenharia de Software oferece padrões, técnicas e ferramentas para serem aplicadas durante o ciclo de vida do software para que sejam desenvolvidos produtos de qualidade. Porém, em muitas empresas essas práticas não são aplicadas e um dos motivos é a falta de recursos.

Com o auxílio das normas de qualidade para produto de software, foi avaliado um sistema para gestão empresarial de uma dessas empresas. A avaliação realizada no software possibilitou perceber a necessidade de se utilizar as práticas de Engenharia de Software como auxílio no desenvolvimento de software. Ficou claro que a ausência dessas práticas afeta a qualidade final do produto de software.

As pequenas empresas de desenvolvimento de software não possuem certificação de qualidade devido ao alto custo.

Esta pesquisa pode auxiliar empresas que enfrentam problemas na manutenção do software e têm dificuldades para satisfazer os clientes, por meio da avaliação da qualidade de seus produtos. Se os resultados da avaliação não forem satisfatórios, as recomendações aqui mencionadas podem ser aplicadas para atingir uma melhoria considerável na qualidade do produto de software.

Nesta pesquisa concluiu-se que as empresas podem oferecer aos seus clientes softwares de qualidade mesmo sem muitos recursos financeiros, basta fazer uso das práticas de Engenharia de Software e estar sempre avaliando a qualidade de seus produtos e buscando melhorias.

Encontraram-se algumas dificuldades devido à falta de documentação do software avaliado, o que dificultou a avaliação e também, a carência de materiais didáticos para auxiliar no entendimento sobre avaliação de produto de software. A primeira dificuldade foi superada com o convívio diário dentro da empresa desenvolvedora do software e o contato direto com o produto e a segunda, quando se conseguiu as normas de qualidade para produto de software na Universidade Federal de Santa Catarina.

Contudo, os objetivos desta pesquisa foram atingidos, os conceitos estudados permitiram a realização de um estudo de caso e a partir daí, foi realizado uma avaliação de qualidade de um produto de software, foram estudadas e recomendadas as técnicas em conformidade com a Engenharia de Software que possibilitam a obtenção da qualidade do software.