

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

Sâmela Soraia Sartin Silva

**UMA LISTA DE CHECAGEM: TESTES E
REQUISITOS DE QUALIDADE
PARA PACOTES DE SOFTWARE**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

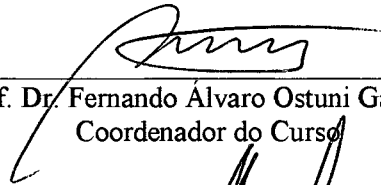
Dr. Vitório Bruno Mazzola

Florianópolis, Outubro de 2000

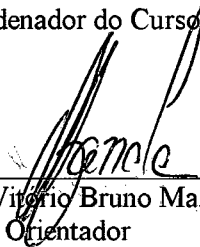
UMA LISTA DE CHECAGEM: TESTES E REQUISITOS DE QUALIDADE PARA PACOTES DE SOFTWARE

Sâmela Soraia Sartin Silva

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.



Prof. Dr. Fernando Alvaro Ostuni Gauthier
Coordenador do Curso



Prof. Dr. Vittorio Bruno Mazzola
Orientador

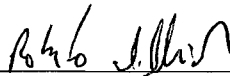
Banca Examinadora




Prof. Dr. Vittorio Bruno Mazzola
Presidente da Banca



Prof. Dr. Murilo Silva de Camargo



Prof. Dr. Roberto Willrich



Prof. M.Sc. Maria Marta Leite

Caminhar, o caminho das pedras
Tatear ao longo do desconhecido
Enxergar não a luz ao fim do túnel
Mas carregar a tocha para iluminar os passos.

No obscuro trajeto do ser inventivo
No inesperado resultado do manipular conhecimento
Vive o objeto sedutor que move o ser de pesquisas.

Seu caminho é o inexequível, seus pés os pioneiros
Nas marchas fatigantes por suas próprias trilhas.

Do atrito com os obstáculos tira o calor
Para aquecer o desejo, voluntário desarmado
Pronto para facear o erro, porém incansável
Até obter êxito.

Roberto Nascimento da Silva

Minha gratidão:

A Deus que sempre esteve presente em meus caminhos, minha vida.
A minha mãe, que firmou meus primeiros passos como mulher que persegue seus objetivos.

Ao meu pai, um descendente de italianos, cujo temperamento adquiri.
Ao meu marido, que apesar da distância que nos separou por dois anos, nunca deixou de me apoiar.

Aos mestres:

Vitório Bruno Mazzola – pelo apoio e orientação com tanta paciência e dedicação
Walter Cybis – que me introduziu nos ensinamentos de redação de artigos técnicos.
Enfim, todas as pessoas que passaram pela minha vida nestes anos de estudo e pesquisas me contemplando com algum ensinamento.

Ao Professor José Gonzaga da Silva Neto, por me proporcionar mais este objetivo alcançado, acreditando apenas no ser humano e no quanto ele pode ser produtivo.

Aos meus amigos, em ordem alfabética:

Adriana e Leopoldo (Guarapuava), Ricardo (Lages), Rosa Maria (Belém), Taís (Salvador), Verinha e Valdete (secretarias da pós-graduação), pela dedicação, carinho e respeito, muito obrigado.

Aos amigos do Centro de Ensino Superior de Primavera, que contribuíram de forma direta ou indireta para que este trabalho tivesse êxito, em ordem alfabética:
Andréa, Fabiana, Iracy (Preta), José Wanderley, Leopoldo Oberst, Regina, Celma, Roseli, Maria, Professores da Faculdade de Primavera e Centro de Formação Profissional “ROSAPRIMA”, pelo companheirismo, espírito de equipe e constante atenção, a minha eterna gratidão.

LISTA DE FIGURAS

Figura 1 - Fatores de qualidade de software de McCall.	26
Figura 2 – Principais fatores da qualidade de produtos.	48
Figura 3– Estrutura Geral do SOFTCHECK	81

LISTA DE QUADROS

Quadro 1 – Definições para qualidade de software.	25
Quadro 2 – Fatores de qualidade de software	27
Quadro 3 – Métricas de graduação definidas por McCall	27
Quadro 4 – Principais técnicas e estratégias de testes de softwares	39
Quadro 5 - Evolução no posicionamento da qualidade do produto x qualidade do processo	47
Quadro 7 - Descrições do Modelo GQM	56
Quadro 8 – Definição de maturidade do Modelo CMM	57
Quadro 9 – Níveis de Maturidade e áreas chave de processo do Modelo CMM	57
Quadro 11 – Níveis de Capacitação	61
Quadro 12 – Características e Subcaracterísticas descritas na norma NBR 13596	65
Quadro 14 – Níveis do Modelo PSP	71
Quadro 16 – Utilitários do SOFTCHECK	82
Quadro 17 – Notação da lista de checagem SOFTCHECK.	82
Quadro 18 – Glossário da lista de checagem	124
Quadro 19 - Comparação entre os modelos e o SoftCheck.	133

LISTA DE TABELAS

Tabela 1 – Implantação de Programa da Qualidade Total ou Similar no Brasil.	55
Tabela 2 – Perfil de Maturidade das Organizações no período de 1992 a 1997	59

SUMÁRIO

<i>LISTA DE FIGURAS</i>	5
<i>LISTA DE QUADROS</i>	5
<i>LISTA DE TABELAS</i>	5
RESUMO	8
ABSTRACT	9
1 INTRODUÇÃO	10
1.1 APRESENTAÇÃO	10
1.2 PROBLEMÁTICA	11
1.3 LIMITAÇÕES DO TEMA	13
1.4 OBJETIVOS	14
1.4.1 Objetivo Geral	14
1.4.2 Objetivos Específicos	14
1.5 JUSTIFICATIVA	14
1.6 ESTRUTURA DO DOCUMENTO	16
2 METODOLOGIA	18
2.1 LEVANTAMENTO E ELEIÇÃO	18
2.2 GERAÇÃO DA LISTA DE CHECAGEM DE SOFTWARE - SOFTCHECK	19
3 REVISÃO DE LITERATURA	20
3.1 QUALIDADE DE SOFTWARE	20
3.1.1 Introdução	20
3.1.2 Qualidade de Software e Garantia de Qualidade de Software	23
3.1.4 Atividades de Teste	38
3.1.5 Atividades de Padrões e Procedimentos	40
3.1.6 Anotação e Manutenção de Registros (documentação)	44
3.1.7 Controle de Mudanças	44
3.1.8 Métricas	45
3.1.9 Qualidade do Produto x Qualidade do Processo	46
3.2 NORMAS E MODELOS DE QUALIDADE	50
3.2.1 Introdução	51
3.2.2 Modelo TQM – Total Quality Management (Gerenciamento da Qualidade Total)	53
3.2.3 Modelo PDCA – Plan-Do-Check-Action (Planejar-Executar-Checar-Agir)	55
3.2.4 Modelo GQM - Goal-Question-Metric (Metas-Questões-Métricas)	56
3.2.5 CMM – Capability Maturity Model (Modelo de Maturidade da Capacidade)	56
3.2.6 SPICE – Software Process Improvement and Capability Determination (Melhoria do Processo de Software e Determinação da Capacidade)	59
3.2.7 Metodologia Bootstrap	62
3.2.8 ISO/IEC 9126 ou NBR 13596 (Tecnologia de informação – Métricas e características de qualidade de software)	65
3.2.9 ISO/IEC 14598 (Tecnologia de informação – Avaliação de produtos de software)	66
3.2.10 ISO/IEC 12119 ou NBR ISO/IEC 12119 (Tecnologia de informação – Pacotes de software – Teste e requisitos de qualidade)	68
3.2.11 Modelo PSP	70
3.2.12 Série ISO 9000	72

3.2.13. ISO IEC 12207 ou NBR ISO/IEC 12207 (Tecnologia de informação – Processos de ciclo de vida de software)	75
4 SOFTCHECK: UMA LISTA DE CHECAGEM	78
4.1 INTRODUÇÃO	78
4.2 SOFTCHECK	79
4.2.1 Estrutura Geral do SOFTCHECK	80
4.2.2 Estrutura Específica dos Módulos	81
4.3 NOTAÇÃO DO SOFTCHECK	82
4.4 SCM1 (MÓDULO1) – PACOTES DE SOFTWARE – TESTES E REQUISITOS DE QUALIDADE	85
S1 – Requisitos de Qualidade	85
S1.1 – Descrição do Produto	86
S1.2 – Documentação do Usuário	97
S1.3 – Programas e Dados	100
S2 – Instruções para Teste	109
S2.1 – Pré-requisitos de teste	109
S2.2 – Atividades de Teste	110
S2.3 – Registros de Teste	116
S2.4 – Relatório de Teste	118
S2.5 – Teste de Acompanhamento	119
S3 Exemplo de uma Descrição de um Produto	121
4.5 SCM2 (SOFTCHECK MÓDULO2): UTILITÁRIOS	123
S1 Glossário	123
S1.2 Termos e Definições	124
4.6 COMPARAÇÃO ENTRE MODELOS APRESENTADOS E O SOFTCHECK	132
4.7 DISCUSSÕES	137
5 CONCLUSÃO	140
5.1 CONTRIBUIÇÕES	141
5.2 SUGESTÃO PARA TRABALHOS FUTUROS	142
<i>BIBLIOGRAFIA</i>	<i>144</i>
<i>BIBLIOGRAFIA REFERENCIADA</i>	<i>147</i>

RESUMO

Este estudo relata a concepção de uma lista de checagem de *software*, definida como um conjunto de questões, cuja finalidade é verificar se um determinado objetivo pode ser constatado como alcançado ou não. Tal lista pode atuar como ferramenta de apoio a profissionais e discentes da área de ciência da computação, mais precisamente a classe de desenvolvedores de pacotes de *software*.

O trabalho permite também que outros pesquisadores tenham acesso a um vasto referencial quantitativo no que tange a normas e modelos direcionados ou não aos produtos de *software*, tornando-se uma fonte significativa para pesquisas futuras.

O ponto fundamental deste estudo se estabelece na elaboração de uma lista de checagem que envolve dois módulos (ou partes). O primeiro, que sugere características de testes e requisitos da qualidade para o produto final de *software* por meio da norma NBR ISO/IEC 12119. E o outro, fornece um glossário de termos técnicos.

Finalmente, no resultado, obtém-se uma lista de checagem de *software*, concebida durante esta pesquisa, denominada SOFTCHECK – *Software's Checklist* (Lista de checagem de *software*), que terá por objetivo criar, em discentes e profissionais, uma cultura de utilização de padrões no desenvolvimento de *software*, ampliando seus conhecimentos e proporcionando a melhoria do processo onde verifica-se o problema. Além disso, estimula o emprego de requisitos não assistidos pelos desenvolvedores, mostrando o que a não incorporação do item acarretaria em termos de: insatisfação do usuário ao adquirir o produto sem informações de uso, com informações inconsistentes por ausência de testes, entre outros.

Palavras-chave: Testes, requisitos de qualidade, normas, modelos, lista de checagem de *software* (SOFTCHECK).

ABSTRACT

This work covers the creation of a checklist for software, which is defined as a set of questions with the purpose of verifying if an objective can be considered as reached or not. The list can serve as a tool to help professionals and students of computer science, more specially for softwares' developers.

The study also permits the researches to have access to a large reference related to rules and models applicable or not to softwares, becoming in this way a significant source for future researches.

The fundamental point is estabilished on the creation of a checklist which has two modules. The first one suggest characteristics for tests and quality requeriments for the final software, by means of the rule NBR ISO/IEC 12119. The second provide a glossary with technical terms.

Finally, as a result, a checklist for software is obtained, created during the research, which list is named SOFTCHEK – Software's Checklist.

The list's goal is to create an environment to encourage the use of standards while developing software, giving to students and professional. The chance to broaden their knowledge and also provide the improvement of the problem solving process.

Besides, it stimulates the use of requeriments never thought by developers, demonstrating that not joining those items could result in user's dissatisfaction, lack of using directions, non reliable information brought about test absence, and so on.

Key-Words: Test, quality requeriment, rules, models, checklist (SOFTCHECK).

1 INTRODUÇÃO

1.1 APRESENTAÇÃO

A padronização acontece desde os primórdios e partiu da comunicação. O fato é que indivíduos necessitavam de um “protocolo” comum para serem compreendidos uns aos outros. Retornando ao presente, normas foram desenvolvidas inicialmente objetivando o intercâmbio de componentes eletrônicos e outros. Desde então a padronização vem sendo presença constante na evolução histórica das grandes descobertas tecnológicas. A padronização busca a uniformização do maior número possível das formas que se prestam aos sistemas de comunicação, indo da comunicação entre os componentes mecânicos e eletrônicos, aos padrões de comunicação entre pessoas, máquinas e até mesmo entre diferentes *softwares*.

O objetivo destes padrões é a redução nos desgastes gerados pelos processos legais, oriundos de discórdias por causa do que foi ou não formalizado entre contratante e contratado, da dependência de indivíduos-chave dentro da organização e, também em desgastes ocasionados por dificuldades na operação de determinados dispositivos cibernéticos.

Com esta finalidade, foram constituídas diversas instituições no cenário mundial. Entre as que possuem prospecção internacional podem ser citadas como exemplo a DIN (Deutsches Institut für Normung – Instituto Alemão para Normalização) e ANSI (American National Standard Institute – Instituto de Padronização Nacional Americano). O Brasil também possui um fórum nacional de normalização, a ABNT (Associação Brasileira de Normas Técnicas).

Entre os padrões nacionais, existem movimentos pela universalização desses padrões em nível mundial. Para esse fim, foi criada em 1947 a International Organization for Standardization uma federação mundial com, atualmente, mais de 100 (cem) organizações nacionais de padronizações (entre elas a ABNT), representando mais de 90 (noventa) países, responsáveis por mais de 95% da população industrial mundial. Com sede em Genebra – Suíça, e tendo como principal atividade a elaboração de padrões para especificações e métodos de trabalho nas mais variadas áreas (menos na área da eletrotécnica e eletrônica, estando essa área sob a responsabilidade da

International Electrotechnical Commission (IEC)), a International Organization for Standardization, para referir-se a si própria, adotou o nome "ISO".

O principal objetivo da ISO é o desenvolvimento de padrões mundiais, com vistas à facilitação do intercâmbio internacional de produtos e serviços e à criação de uma cooperação intelectual, científica, econômica e técnica entre as nações.

No Brasil, o órgão PBQP (Programa Brasileiro de Qualidade e Produtividade), implantado em 1993, na cidade de Curitiba-PR, além de ativo nos estudos de elaboração e desenvolvimento de novas normas, também possui como finalidade auxiliar grandes, médias e pequenas empresas na utilização de normas e modelos de qualidade e produtividade. Na realidade, o PBQP tenta modificar seus dados levantados até a última publicação de 1999, onde na maior parte das empresas cadastradas junto ao órgão, o índice de conhecimento e utilização de tais processos são baixíssimos. Um exemplo, é a utilização do modelo de qualidade TQM (Total Quality Management), um dos mais conhecidos mundialmente, exibe em 1997, que, de uma amostragem de 588 empresas, somente 17% possuem um processo implantado, contra 83% em estudo (ou implantação) ou não implantado, sendo 46,3% na última condição. Para outros modelos a situação é ainda mais crítica onde a utilização avançada e que se preocupa somente com a melhoria contínua, atinge somente 0,6 % dentre 94% em outros estágios.

Deste fato, se dá a motivação deste trabalho, que visa a disseminação do conhecimento e/ou a utilização de padrões e modelos com maior facilidade, tanto na acessibilidade, como na interpretação dos documentos e outros. E, é por meio de uma lista de questões baseada em normas técnicas que o trabalho pretende atingir suas metas.

1.2 PROBLEMÁTICA

Sistemas de normalização têm se mostrado cada dia mais presentes no mercado mundial, que a exigência de produtos de qualidade, originais (inéditos) e que sua concepção acompanhe a evolução tecnológica.

Apesar da grande quantidade de normas e modelos disponíveis no mercado, sua utilização atinge marcos baixíssimos. Esta condição possui como principal causa a extensibilidade e interpretação das fontes. A primeira deve-se ao fato de que a padronização é um processo complexo, devido ao pouco tempo normalmente disponível para a leitura, interpretação e adequação ao processo em vigor. E, a segunda, deve-se à

subjetividade da interpretação dos usuários, onde cada indivíduo as interpreta de acordo com suas experiências.

Outra consideração a ser feita está relacionada às experiências detidas no Brasil com o uso de tais normas e modelos. Pesquisas realizadas pelo PBQP evidenciam que a quantidade de empresas desenvolvedoras de *software* no Brasil, usuárias de qualquer modelo ou norma, ocupa uma fatia insignificante, contra uma grande maioria que não utiliza método algum para garantir qualidade ao menos sobre aspectos básicos e previsíveis.

Normas técnicas vêm sendo formuladas e reformuladas a mais uma de década e ainda assim, seu uso se encontra restrito a poucos profissionais. Esta limitação não é ocasionada somente por dificuldades de acessibilidade mas, muitas vezes, no desconhecimento e na falta de conscientização da utilização de padrões e modelos.

A implantação de padrões visando qualidade, implicaria em mudanças não somente no “chão de fábrica”, mas também na empresa como um todo. Seria necessária uma equipe multidisciplinar, composta de, por exemplo, um profissional encarregado da implantação e manutenção ou melhoria do projeto de padronização, e pessoas diretamente envolvidas com o desenvolvimento do produto. Além disso, é necessária uma mudança na cultura da empresa no que se refere ao comportamento das pessoas e ao processo produtivo, causando dificuldades na conscientização, já que o retorno é sempre a longo prazo.

Outro problema que a ausência de padrões ocasiona é a dificuldade na substituição de pessoas-chave. Tal situação deve-se ao fato de que os substitutos perdem tempo demasiado na tentativa de compreensão de um processo sem padronização, o que não aconteceria em casos de processos padronizados, onde a continuidade do projeto não sofreria grandes influências.

Não obstante de todas as complicações mencionadas acima, os profissionais de engenharia de *software* ainda deparam-se com usuários (contratantes de serviços) cada vez mais intolerantes, exigentes, conhecedores dos recursos oferecidos pela informática e das tecnologias a ela ligada, colocando muitas vezes, o desenvolvedor em situações constrangedoras. Como conceber um produto com qualidade, inovador, de preço acessível e principalmente em um curto espaço de tempo? A solução encontrada para esta questão é a padronização dos processos, que aliada a inovação tecnológica, atende

às exigências dos diversos usuários, possibilitando a produção de *softwares* com mais qualidade e em menos tempo.

A variável tempo nos dias atuais é um requisito de muita importância, estando acima somente o requisito da qualidade. No entanto, é necessário salientar que uma alta produtividade não significa, na maioria das vezes, senão todas, que os produtos possuam qualidade. Na realidade, produtos desenvolvidos em grande escala possuem tendência a perder a qualidade. Havendo, portanto, necessidade de controlar as atividades não apenas na produção, mas no caso de desenvolvimento de sistemas de *software*, também do “capital intelectual”, que deverá consumir uma grande fatia de tempo para o seu controle. Esta última variável se mantém no pico das dificuldades, pois o ser humano é muito “melindroso” e exige um controle rigoroso sobre suas necessidades primordiais dentro de seu ambiente, para que o quesito produtividade cresça, mas, que a qualidade permaneça.

Além das implicações envolvendo o profissional, existe também a questão da disseminação do conhecimento a nível acadêmico. Novatos ou futuros profissionais, não possuem o hábito de padronizar seus processos. Normalmente, seus sistemas não possuem documentação e nem uma modelagem de objetos lógica, gerando *softwares* que dificultam a manutenção por outros profissionais.

Todas estas considerações: extensibilidade, interpretação das fontes, desconhecimento, falta de conscientização, dificuldades de acessibilidade, variável tempo, aprendizagem de discentes e a crescente exigência dos usuários levam a identificação da necessidade da geração de ferramentas para auxiliar na qualidade de produtos e produtividade dos indivíduos.

1.3 LIMITAÇÕES DO TEMA

A estrutura do projeto foi elaborada para abranger vários modelos e normas a fim de que usuários deste trabalho tivessem a oportunidade de manter contato e identificar-se com uma delas com a finalidade de utilizá-las como fonte geradora de bons processos de produção de *software*.

No entanto, encontram-se fora do escopo deste trabalho:

a) outras normas e modelos, que foram deixadas para trabalhos futuros, considerou-se apenas as normas: NBR ISO/IEC 12119, referente à Tecnologia de

Informação – Pacotes de Software – Teste e requisitos de qualidade e a NBR ISO 8402, que trata a Gestão da qualidade e garantia da qualidade – Terminologia;

b) a avaliação da qualidade das questões, a verificação quanto a completeza, entre outros;

c) testes com usuários para validação das questões no que se refere à eficácia e/ou eficiência;

d) implementação das questões da lista, em uma determinada linguagem de programação.

1.4 OBJETIVOS

1.4.1 Objetivo Geral

O objetivo deste trabalho é propor uma ferramenta (lista de checagem) de auxílio à futuros profissionais, novatos e experientes de engenharia de *software* tratando a importância, facilidade de uso e disseminação de conhecimento sobre a utilização de normas, proporcionado por uma lista de verificação.

1.4.2 Objetivos Específicos

- a) Verificar e analisar normas, modelos e recomendações existentes;
- b) Eleger aquelas que serão utilizadas como base para a confecção da lista de checagem de *software*;
- c) Propor uma ferramenta inédita, baseada nas fontes eleitas, constituindo assim, uma lista de checagem pioneira, capaz de auxiliar desenvolvedores de *software* a padronizar seus processos de testes e requisitos de qualidade.

1.5 JUSTIFICATIVA

O crescimento do uso e conhecimento de normas e padrões tem sido lento e insignificante em relação ao número de empresas desenvolvedoras de *software* com grande participação no mercado nacional.

A qualidade está cada vez mais comprometida. Vários projetos são concebidos por profissionais desqualificados a frente do desenvolvimento, sem a utilização de processos formais e padrões. O resultante destas afirmações evidenciam grandes

rejeições sofridas nas exportações, favorecendo importações de produtos e pessoal, desestabilizando profissionais e empresas do mercado nacional.

O essencial para reverter tal situação é a padronização. Esta possibilita o desenvolvimento de produtos com qualidade, inovadores, com preços acessíveis e em um curto espaço de tempo, implicando na melhoria dos processos de produção de uma infinidade de tipos de produtos.

Investir em qualidade e pensar em produtividade acarreta uma série de mudanças dentro do processo. Mudanças que exigem uma adaptação às novas metodologias, implicando no conhecimento, teste, avaliação e, em principal, na aceitação do método como um processo que possibilita o alcance de metas de produtividade com qualidade.

Os tópicos mencionados acima ocasionaram o desenvolvimento deste estudo, que culminou em uma lista de checagem. Tal lista foi definida como um conjunto de questões propostas com a finalidade de verificar se um determinado objetivo pode ser constatado como alcançado ou não. Denominado como SOFTCHECK (*Software's Checklist* – Lista de checagem de *software*), a lista proposta neste trabalho encontra-se na forma de questões de verificações, apresentando recomendações quando itens de padronização não são utilizados. Estas questões proporcionam: a diminuição da subjetividade e complexidade dos itens, a modelagem de processos, a padronização de objetos, facilidade no acesso e uso de normas, auxílio na disseminação do conhecimento, verificação quanto a conformidade com padrões internacionais, redução no tempo para a elaboração de documentações adequadas e instruções de teste. E, todas estas considerações, visam atender satisfatoriamente a crescente exigência dos usuários.

Diminuindo a subjetividade e complexidade por meio da adequação dos itens constantes nas normas técnicas em questões/perguntas, objetiva-se uma redução dos problemas de interpretação das normas e modelos após a utilização do SOFTCHECK, visto que, tais problemas originam-se da apresentação das normas em forma de contrato.

Por sua vez, a modelagem de processos e padronização de objetos tende a uma redução de problemas relacionados: ao tempo de desenvolvimento de manuais, instruções para testes, entre outros; a substituição de pessoas-chave, refere-se à condução de fases finais por pessoas que não eram ativas nas iniciais; à conscientização dos discentes, verificada pelas recomendações; além de proporcionar o aumento da produtividade, pois, a partir da padronização de processos e adequação dos problemas,

tem-se todos os passos definidos, restando apenas a melhoria contínua: atualização tecnológica, gerencial, etc; que quando alcançados visam uma alta produção, de qualidade não comprometida.

Partindo de problemas como dificuldade de acesso, desconhecimento e a baixa disseminação de normas e modelos, o SOFTCHECK se apresentará como facilitador na acessibilidade e utilização destes, caracterizando-se como uma fonte de conhecimento que poderá estar disponível aos interessados em desenvolvimento de *softwares* com qualidade, sempre que se fizer necessário.

Finalmente, a conscientização é um fator subjetivo. Assim sendo, para alcançá-la é necessária uma mudança cultural de cada indivíduo. Esta mudança é almejada pelo SOFTCHECK por meio de recomendações, onde são especificados os problemas decorrentes da não empregabilidade de algum requisito.

Portanto, o SOFTCHECK pretende ser um instrumento de relevância para toda a comunidade de desenvolvedores de *software*, pois este assiste aos problemas listados no item 1.2 (problemática) do capítulo introdutório, agindo como modificador na situação atual de não empregabilidade de padrões e recomendações em processos da área de engenharia de software, mais especificamente, testes e requisitos da qualidade para pacotes de software.

1.6 ESTRUTURA DO DOCUMENTO

Neste documento são apresentadas as várias formas de avaliar, alcançar e garantir qualidade e produtividade em processos e produtos, abrindo margens ao desenvolvimento de uma ferramenta de auxílio para facilitar o uso de tais técnicas.

No capítulo inicial, introduziu-se a padronização como fator relevante para alcance da qualidade e produtividade de produtos, bens ou serviços. O segundo capítulo mostra as etapas a serem seguidas para o desenvolvimento deste estudo.

No capítulo seguinte, apresenta-se a revisão literária. Na primeira parte, são tratadas as diretrizes de qualidade de software como fator chave para a obtenção de bons resultados no processo de desenvolvimento. E, na segunda, encontra-se uma descrição das principais normas e modelos para aplicações gerais e específicas de software.

A proposta deste estudo é descrita após desenvolvida, no quarto capítulo, onde enfoca-se a geração da lista de questões de verificação, e a lista em si, além de uma

notação desenvolvida para atender as necessidades no momento da estruturação do projeto.

Finalmente, no capítulo conclusivo, é apresentado um comparativo entre este estudo e outros trabalhos correlatos, evidenciando vantagens e desvantagens. E encerrando este capítulo, são destacadas as contribuições e sugestões para trabalhos futuros.

2 METODOLOGIA

O trabalho proposto será desenvolvido em duas etapas: levantamento e eleição das normas e modelos, e geração de questões de verificação (lista de checagem de *software* – SOFTCHECK), acompanhadas de recomendações à empregabilidade da questão, em caso de respostas negativas.

2.1 LEVANTAMENTO E ELEIÇÃO

Fontes sobre padrões de qualidade e produtividade foram pesquisadas e relatadas neste estudo. A preocupação inicial se concentrou em fornecer um referencial vasto a fim de suprir necessidades de estudos na área. Com isto, incluiu-se neste trabalho, não apenas fontes ligadas ao software mas também de características gerais.

Para a concepção da lista de checagem, foram selecionadas algumas dentre as várias normas e modelos relatadas. Como critério na eleição das mesmas, foram analisados os requisitos de abrangência, acessibilidade, completude e disponibilidade das fontes em questão. No entanto, devido ao “modo” em que as normas estão dispostas, normalmente, enfocando pontos específicos, houve a necessidade de uma combinação de ferramentas. Tal combinação possibilita, após a limitação deste trabalho, a obtenção de uma ferramenta abrangendo a questão da finalização do produto - testes e requisitos de qualidade para pacotes de software, além de, incluir uma norma de terminologia, originando um glossário de termos técnicos.

O estudo, está disposto da seguinte maneira:

- a) Pesquisa bibliográfica referenciando fontes destinadas à produtividade e qualidade como um todo, não somente as específicas para o *software*.
- b) Identificação das fontes que tratam do assunto de qualidade e produtividade na área de engenharia de *software*;
- c) Organização de normas e modelos em categorias, considerando suas características quanto à empregabilidade;
- d) Delimitação e eleição de normas para tratar os processos de testes e requisitos de qualidade para pacotes de software e termos correspondentes à gestão e garantia da qualidade,
- e) Estruturação das fontes em dois módulos:
 - 1) Pacotes de Software – Teste e Requisitos de Qualidade;

2) Utilitários

- f) Seleção de itens aplicáveis em lista de checagem de *software*;

As questões da lista de checagem de *software* serão geradas a partir de:

- a) Normas Internacionais, versão traduzida para o português pela ABNT:
NBR ISO/IEC 12119 – Tecnologia de informação – Pacotes de *software*
– Teste e requisitos de qualidade e NBR ISO 8402 – Gestão da qualidade e garantia da qualidade – Terminologia;
- b) Experiências com o desenvolvimento de um artigo¹, que tratou a avaliação ergonômica de um sistema multimídia por meio da utilização de um *checklist* (lista de verificação) especializado denominado Ergolist².

2.2 GERAÇÃO DA LISTA DE CHECAGEM DE *SOFTWARE* - SOFTCHECK

- a) Desenvolvimento de uma notação própria para a lista, permitindo a supressão de palavras que teriam alta taxa de utilização;
- b) Transformação dos itens constantes nas fontes em questões/perguntas, anexando recomendações quando da não empregabilidade das mesmas;
- c) Revisão da consistência das questões, envolvendo ajuste, agrupamento e eliminação de questões similares.

¹ Cybis, Walter A.; Pereira, Danilo; Silva, Sâmela S. S. *Avaliação de Software hipermídia pela ferramenta "Ergolist": Uma estratégia válida*. Abergó'99. Salvador, Bahia.

² Site na internet (<http://www.labiutil.inf.ufsc.br/ergolist>), destinado a apoiar inspeção da usabilidade de interfaces com o usuário, desenvolvido pelo laboratório de utilizabilidade da Universidade Federal de Santa Catarina.

3 REVISÃO DE LITERATURA

3.1 QUALIDADE DE SOFTWARE

3.1.1 Introdução

Qualidade é um tema que vem se projetando cada vez mais no mercado nacional e internacional. Este crescimento teve maior importância para empresas e profissionais no Brasil devido, em principal, a dois acontecimentos: a pressão da competição (globalização), aliada a uma maior consciência dos consumidores [WEB99]. O fenômeno da globalização, tem resultado em uma grande mudança de posicionamento das empresas produtoras de bens ou serviços. Tendo em vista a liberalização dos mercados (importações e exportações), exigindo o crescimento e a modernização da indústria e da prestação de serviços, observam-se as influências refletidas no mercado nacional a ponto de modificar o processo de desenvolvimento, exigindo um padrão de qualidade equiparado ou melhor ao internacional; talvez assim, caminhando para uma nova fase de grandes descobertas e avanços tecnológicos.

A conscientização dos consumidores, se resume no amadurecimento das massas populares em exigir que os produtos e serviços que estão adquirindo possuam garantia de que, o que eles (os produtos) se propõe a fazer, realmente o fazem.

Esta conscientização de qualidade dos produtos vem sendo verificada a alguns poucos anos nos produtos de software, devido às suas peculiaridades. O software (programa de computador) possui características que o fazem diferentes de outras realizações do ser humano. Quando se constrói um equipamento (hardware), o processo criativo é traduzido em uma forma física. No caso do software, o processo é lógico.

Software é algo ainda novo, uma vez que o primeiro computador eletrônico digital só apareceu em 1946. Pode-se dizer que toda a experiência acumulada no setor de software (científica, tecnológica, industrial, comercial, nacional e internacional) soma pouco mais de 50 anos [WEB94,WEB99].

As atividades relacionadas ao desenvolvimento de software têm sofrido profundas transformações nos últimos 25 anos. Durante esse tempo, sob uma perspectiva histórica, pôde-se notar três momentos importantes que caracterizaram mudanças de posicionamento na maneira de se tratar o software [ABE95]. O primeiro momento, que tem seu ponto alto durante os anos 60/70, caracteriza-se principalmente pelo enfoque

voltado ao código ou produto. A atividade de desenvolvimento concentra-se essencialmente na codificação e teste dos programas elaborados. A qualidade do produto é decorrência da habilidade do programador, das estratégias de teste unitário e de integração empregadas. Nesta época, o desenvolvedor (programador) possuía o controle total de todo o processo de produção do software. O papel do usuário, se restringia à “fase de utilização” do programa, isto é, não participava ativamente de nenhuma das fases anteriores de desenvolvimento, seu contato com o sistema se originava na conclusão após os testes. Qualquer problema grave que fosse averiguado, seria corrigido na fase de manutenção.

Aos poucos, a ênfase se desloca do produto para o processo. A motivação para isso vem dos problemas surgidos no desenvolvimento de software, cuja principal causa origina-se da ausência de meios sistemáticos para a formulação de especificações de requisitos e maneiras de transformá-las em código. O final dos anos 70 e em particular a década de 80 tornou-se uma época bastante profícua no surgimento de modelos de desenvolvimento, onde métodos para análise de requisitos e para o projeto de software constituem seu principal enfoque. A qualidade é fruto dos resultados obtidos em cada etapa do desenvolvimento. Neste período, o usuário obteve maior destaque colaborando em principal nas especificações de requisitos, onde técnicas de coletas de dados como entrevistas, formulários e pesquisas em campo se destacaram como sendo uma fase relevante sob o ponto de vista das necessidades dos usuários. A participação do usuário no início do processo de desenvolvimento de software, ajudando o desenvolvedor a visualizar o domínio do problema a ser informatizado foi de grande valia para ambas as partes. De um lado, se encontra um profissional que desconhece o processo de produção manual do produto em questão e do outro lado, existe um usuário que necessita de um sistema que o libere do trabalho “mecânico” exaustivo que as tarefas rotineiras lhe impõe. Um exemplo poderia ser o processo da folha de pagamento, que através de um simples procedimento contendo uma fórmula para calcular o salário dos funcionários, teria capacidade de calcular em segundos o que demoraria dias para o ser humano.

Entretanto, com as vantagens, também as desvantagens se instalam nesta mudança. Os usuários, começaram a omitir informações com medo da dispensa pela não necessidade de seus serviços. E, por motivo desta omissão é que nos deparamos ao final dos anos 80, influenciado principalmente pelas idéias de qualidade total então

apregoadas, à tratar o desenvolvimento de software com uma visão mais geral, que envolve não somente os aspectos técnicos do desenvolvimento, mas, principalmente, a qualidade como um instrumento global que envolve a organização da empresa como um todo. Esta mudança foi marcada pela introdução dos líderes de equipes para a colaboração com a informatização. Agora, os profissionais consultados possuem grande interesse na realização dos processos de desenvolvimento, pois isto resultaria em maior produtividade e conseqüentemente destaque ao gerente da área. No entanto, não se resume apenas no intento de utilizar as ambições dos profissionais em questão, e sim na utilização dos seus conhecimentos do domínio do processo com visão individual e geral, pois não devemos nos esquecer que usuários podem ter dificuldades adversas, ou em diferentes níveis. Um caso importante de “parceria” foi relatado em [FRI94], onde duas grandes empresas, a POLO de Software de Curitiba S/A e a Companhia de Cimento Portland Rio Branco, também de Curitiba, se reuniram para desenvolver um sistema. A POLO foi contratada para prestação de serviços de reengenharia e desenvolvimento do Sistema de Planejamento Financeiro da Portland, do grupo Votorantin. Um ponto relevante, dentre muitos relatados, é a constituição da equipe técnica, a qual contou com onze profissionais da empresa contratada (um gerente de projetos, um consultor em sistemas financeiros, um analista líder, dois analistas de sistemas sênior, cinco programadores e um documentador) e também, cinco profissionais da empresa contratante (um gerente usuário patrocinador, três analistas do usuário e um analista de sistemas sênior). Este relato evidencia as características apresentadas acima, onde um grupo de profissionais da empresa contratante foi designado a trabalhar em conjunto com a equipe contratada para suavizar as questões de domínio do sistema, e dar ênfase ao que realmente possuía grande importância que era a carência de profissionais capacitados na nova tecnologia que seria utilizada, poupando tempo e esforços desnecessários.

Em virtude do surgimento desses novos elementos no desenvolvimento de software e da mudança na postura de como tratá-los, novas métricas e ajustes naquelas já existentes fizeram-se necessárias a fim de viabilizar o emprego de métricas dentro desse contexto evolutivo.

Neste capítulo apresentam-se fatores que influenciam a qualidade do software (sessões 3.1.2 à 3.1.7), a relação entre a qualidade do produto e qualidade do processo

(sessão 3.1.8) e uma descrição sobre os principais procedimentos e implicações para a certificação de qualidade (sessão 3.1.9).

3.1.2 Qualidade de Software e Garantia de Qualidade de Software

Muitas definições de qualidade de software têm sido propostas na literatura. Segundo a atual norma brasileira sobre o assunto [NBR94a], a qualidade é: *A totalidade das características de uma entidade que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas.*

Onde a entidade é o produto do qual estamos falando, que pode ser um bem ou serviço. As necessidades explícitas são as próprias condições e objetivos propostos pelo produtor. As necessidades implícitas incluem as diferenças entre os usuários, a evolução no tempo, as implicações éticas, as questões de segurança e outras visões subjetivas[BAR97].

Outras características que definem a qualidade de software é apresentada por Juran[MIL93]: *Software de qualidade é aquele que não apenas satisfaz as exigências, mas também é implementado a tempo e de acordo com o orçamento.*

Normalmente as empresas de software não atendem à estes últimos requisitos[FIO94], o que gera descredibilidade ao contratado e impaciência por parte do cliente.

Em outra definição de qualidade, sendo a mesma utilizada para designar produtividade em [WEB99], possui uma estreita relação com [SOM96] e Watts S. Humphrey (em sua contribuição para a obra [WEB99]):

A primeira descreve a qualidade de software como: *uma filosofia de gestão, que visa conduzir as organizações a uma postura de melhoria de seus processos, por meio do compromisso de seus dirigentes e empregados.* Tal postura assegura produtos e serviços com desempenho, preço e disponibilidade adequados, totalmente orientados para as aspirações do cliente. A segunda apesar de não estar claramente apregoada no contexto, conduz um entendimento de que software com qualidade depende de uma série de fatores (segurança, confiabilidade, rastreabilidade, robustez e outros), e que todos são alcançados e mantidos por meio de um bom gerenciamento da qualidade (o que está relacionado ao posicionamento mencionado acima). Faz menção ainda, a uma possível “cultura da qualidade” onde todos os responsáveis pelo desenvolvimento do

produto estejam comprometidos com a qualidade. Neste último ponto, de acordo com Humphrey, está o fator chave para a qualidade de software: “uma nova cultura de software”. Esta mudança na cultura de software está relacionada ao uso de todas as ferramentas disponíveis na área de desenvolvimento de software para tratar o processo de melhoria, e, em principal, na conscientização e na gerência do capital humano (gerentes e engenheiros de software).

Para finalizar, Pressman [PRE95], define qualidade de software como: *Conformidade a requisitos funcionais e de desenvolvimento explicitamente declarados a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo o software profissionalmente desenvolvido.*

Essa definição sem dúvida poderia ser modificada ou ampliada. Sob o ponto de vista de [PRE95], a definição apresentada anteriormente serve para enfatizar 3 pontos importantes:

1. Os requisitos de software são a base a partir da qual a qualidade é medida. A falta de conformidade aos requisitos significa falta de qualidade.
2. Padrões especificados definem um conjunto de critérios de desenvolvimento que orientam a maneira segundo o qual o software passa pelo trabalho de engenharia. Se os critérios não forem seguidos, o resultado quase que seguramente será a falta de qualidade.
3. Há um conjunto de requisitos implícitos que freqüentemente não são mencionados (por exemplo, o desejo de uma boa manutenibilidade). Se o software se adequar aos seus requisitos explícitos, mas deixar de cumprir seus requisitos implícitos, a qualidade de software será suspeita.

Apresentar uma conclusão sólida de definição sobre qualidade de software ainda é um tanto difícil, pois apesar dos muitos estudos realizados, não existe um acordo entre os profissionais para designar o que é qualidade de software. No entanto, com base nas questões levantadas acima, há alguns pontos críticos que podem ser destacados, veja Quadro 1.

Quadro 1 – Definições para qualidade de software.

Fonte	Pontos Críticos de Definições de Qualidade de Software
NBR ISO 8402	Atender aos requisitos, contemplando os explícitos e implícitos
Juran [MIL93]	Atender aos requisitos, enfatizando cronograma e custos
Weber et. Al. [WEB99]	Filosofia de gestão, enfatizando desempenho, preço e com grande enfoque na orientação ao cliente (atender aos requisitos).
Sommerville [SOM96]	Atender a requisitos e muitos outros fatores, contemplando o gerenciamento da qualidade (como WEB99) e uma “cultura de qualidade” (como Humphrey).
Humphrey [WEB99]	Mudança na cultura de software do capital humano, contemplando gerenciamento de qualidade (visando a melhoria).
Pressman [PRE95]	Atender a requisitos, contemplando os explícitos e implícitos e definir padrões especificados.

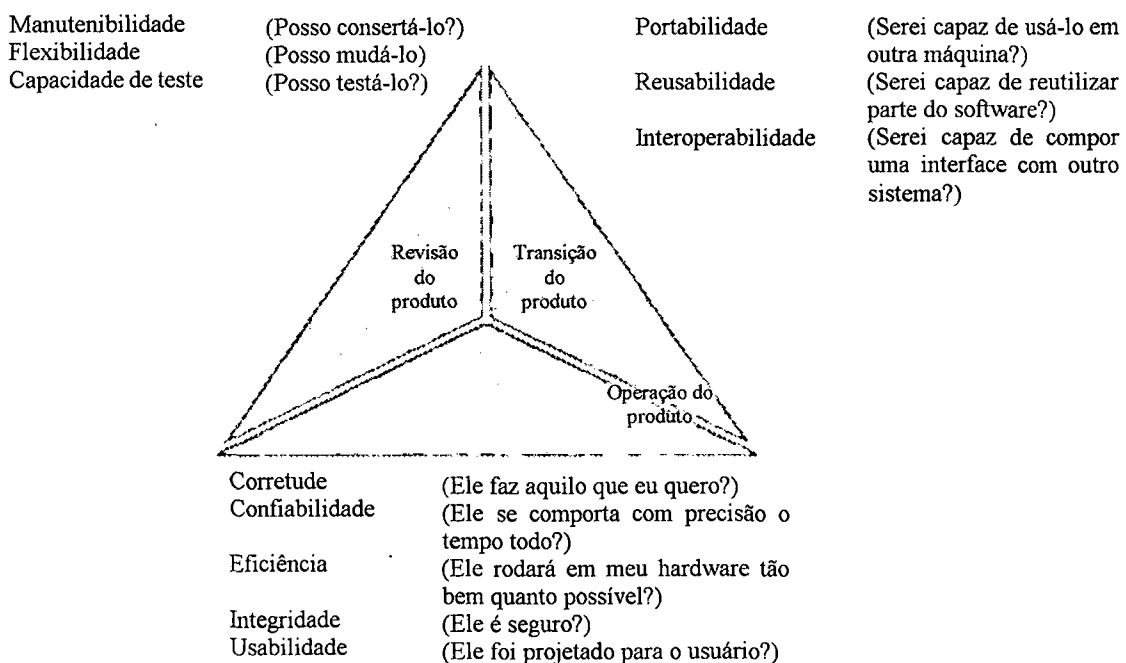
Podemos dizer que uma cultura de software, poderia nos levar aos outros requisitos. No entanto, verificando nos passos históricos, a cultura de software atual levou quase quatro décadas desde os anos 60, onde iniciaram-se os processos de testes, para chegar onde se encontra. Está claro que para o reprojeto da atual cultura haverá fundamentações sólidas que já se encontram disponíveis para construir suas bases, isto é, levando-se em conta os “instrumentos” existentes (métodos, técnicas, sistemas de avaliações e outros), a complexidade que atingiu a marca de quatro décadas para ser construída pode ser, tecnicamente, muito reduzida. Mas, ainda assim, a previsão é que a realização desta nova cultura de software também se prolongue por anos até o total reprojeto, pois as maiores mudanças estão relacionadas à cultura do capital humano, isto implica em uma mudança organizacional nas empresas sobre as concepções de gerentes e educacional na formação acadêmica de engenheiros.

O fato é que devemos “atacar” dos dois lados: tentar mudar a cultura de software e também fazer com que os outros elementos que estão em constante evolução, tais como as normas, modelos, ferramentas e outros, trilhem o caminho para esta progressão, já que a qualidade de software também pode ser dita como uma combinação complexa de fatores que variarão de acordo com diferentes aplicações, tecnologias utilizadas e clientes que as solicitam.

3.1.2.1. Fatores de Qualidade de Software

Os fatores que afetam a qualidade de software podem ser categorizados em dois grupos amplos: (1) fatores que podem ser medidos diretamente (por exemplo, erros/KLOC/unidade de tempo) e (2) fatores que podem ser medidos apenas indiretamente (por exemplo, usabilidade ou manutenibilidade). Em cada caso, deve ocorrer medição. Precisamos comparar o software (documentos, programa e dados) com algum dado e chegar a uma indicação de qualidade. McCall e seus colegas [MCC77] propuseram uma categorização útil dos fatores que afetam a qualidade de software. Esses fatores da qualidade de software, mostrados na Figura 1, focalizaram três aspectos importantes de um software: suas características operacionais, sua manutenibilidade de mudanças e sua adaptabilidade a novos caminhos.

Figura 1 - Fatores de qualidade de software de McCall.



Em relação aos fatores anotados na Figura 1, McCall apresenta as seguintes descrições (Quadro2):

Quadro 2 – Fatores de qualidade de software

Descrições dos fatores definidos por McCall
Corretude: à medida que um programa satisfaz sua especificação e cumpre os objetivos visados pelo cliente.
Confiabilidade ou Robustez: à medida que se pode esperar que um programa execute sua função pretendida com a precisão exigida, mesmo em situações anormais.
Eficiência: a quantidade de recursos de computação e de código exigida para que um programa execute sua função.
Integridade: à medida que o acesso ao software ou a dados por pessoa não-autorizadas pode ser controlado.
Usabilidade: o esforço para aprender, operar, preparar a entrada e interpretar a saída de um programa.
Manutenibilidade: o esforço exigido para localizar e reparar erros num programa.
Flexibilidade: o esforço exigido para modificar um programa operacional.
Testabilidade: o esforço exigido para testar um programa a fim de garantir que ele execute sua função pretendida.
Portabilidade: o esforço exigido para transferir o programa de um ambiente de sistema de hardware e/ou software para outro.
Reusabilidade: à medida que um programa (ou partes de um programa) pode ser reusado em outras aplicações – relacionada ao empacotamento e escopo das funções que o programa executa.
Interoperabilidade: o esforço exigido para se acoplar um sistema a outro.

Fonte: [MCC77]

É difícil, e em certos casos impossível, desenvolver medidas diretas dos fatores de qualidade acima. Portanto, um conjunto de métricas é definido e usado para desenvolver expressões para cada um dos fatores, de acordo com a seguinte relação:

$$F_q = c_1 \times m_1 + c_2 \times m_2 + \dots + c_n \times m_n$$

Onde F_q é um fator da qualidade de software, c_n são coeficientes de regressão e m_n , métricas que afetam o fator de qualidade. Infelizmente, muitas das métricas definidas por McCall só podem ser medidas subjetivamente. As métricas podem estar na forma de uma lista de conferência (*checklist*), que é usada para “graduar” atributos específicos do software [CAV78]. O esquema de graduação proposto por McCall é uma escala de 0 (baixo) a 10 (elevado). As seguintes métricas são usadas no esquema de graduação (Quadro 3):

Quadro 3 – Métricas de graduação definidas por McCall

Métricas de graduação
Auditabilidade: a facilidade com que se pode checar a conformidade aos padrões
Acurácia: a precisão das computações e do controle.
Comunidade de comunicação (Communication Commonality): o grau em que interfaces padrões, protocolos e larguras de banda (bandwidths) são usados.
Inteireza: o quanto a implementação total da função requerida foi conseguida.
Concisão: a compactação do programa em termos de linhas de código.
Consistência: o uso de técnicas de projeto e documentação uniformes ao longo do projeto de desenvolvimento de software.
Comunidade de dados (Data Commonality): o uso de estruturas e tipos de dados padrões ao longo do programa.
Tolerância a erros: o dano que ocorre quando o programa encontra um erro.
Eficiência de execução: o desempenho de run-time de um programa.
Expansibilidade: o quanto o projeto de arquitetura, procedimental e de dados podem ser ampliados.
Generalidade: a amplitude de aplicação em potencial de componentes do programa.
Independência de hardware: o quanto o software é desvinculado do hardware em que opera.
Instrumentação: o quanto o programa monitora sua própria operação e identifica erros que venham a ocorrer.
Modularidade: a independência funcional dos componentes do programa.
Operabilidade: a facilidade de operação de um programa.
Segurança: a disponibilidade de mecanismos que controlem ou protejam programas e dados.
Autodocumentação: o quanto o código-fonte apresenta documentação significativa.
Simplicidade: o quanto um programa pode ser entendido sem dificuldade.
Independência do software básico: o quanto um programa é independente de particularidades não-padroneizadas de linguagens de programação não-padrão, das características de sistemas operacionais e de outras sujeições ambientais.
Rastreabilidade: a capacidade de rastrear uma representação de projeto ou componente de programa até os requisitos.
Treinamento: o quanto o software auxilia no sentido de ajudar novos usuários a aplicarem o sistema.

Fonte: [MCC77]

Estes fatores propostos por McCall [MCC77], representam uma dentre uma série de “listas de conferências” sugeridas para a qualidade de software. No capítulo seguinte, será apresentada a norma NBR 13596 (Tecnologia de informação – Avaliação de produto de software – Características de qualidade e diretrizes para o seu uso, 1996) que discorre sobre características de qualidade de software bastante parecidas com as definidas por McCall, abrangendo também diretrizes para o seu uso.

3.1.2.2 Garantia de Qualidade de Software

Philip Crosby [CRO79], define a garantia de qualidade de software (Software Quality Assurance – SQA) como uma atividade “guarda-chuva” que é aplicada ao longo do processo de engenharia de software. A SQA abrange (1) métodos e ferramentas de análise, projeto, codificação e teste; (2) revisões técnicas formais que são aplicadas durante cada fase de engenharia de software; (3) uma estratégia de teste de múltiplas fases; (4) controle da documentação de software e das mudanças feitas nela; (5) um procedimento para garantir a adequação aos padrões de desenvolvimento de software (quando aplicáveis) e (6) mecanismos de medição e divulgação.

Na realidade, a meta a ser alcançada quando passamos por todos os estágios do ciclo de vida de desenvolvimento de software, seguindo métodos técnicos, visamos a obtenção da qualidade. No entanto, o desenvolvimento de software talvez seja, dentre as atividades intelectuais realizadas pelo homem, a que mais tenha negligenciado a prática intrínseca de medidas, onde níveis de precisão cada vez mais apurados vêm sendo exigidos. As histórias de projetos de softwares fracassados que aparecem na literatura poderiam ter tido um final diferente se, entre outras providências, a realização de medidas fosse tomada como algo que traz resultados efetivos ao processo e não considerada, primordialmente, como uma atividade consumidora de tempo e recursos [PRE95]. Além disso, não basta medir a qualidade aleatoriamente, há a necessidade de planejar e executar o padrão planejado sistematicamente. Neste ponto surge os termos gerenciamento da qualidade e garantia da qualidade.

A garantia da qualidade de software (SQA) é um “padrão sistemático e planejado de ações” [SCH87] que são exigidas para garantir a qualidade de software. O planejamento de ações e conseqüentemente de padrões de software de uma determinada empresa são adaptados e realizados normalmente por gerentes da qualidade de software, sendo responsáveis por três tipos de atividades [SOM96]:

1. *Garantia de qualidade*: eles precisam estabelecer padrões e procedimentos organizacionais que conduzem ao alto nível de qualidade de software.
2. *Planejamento da qualidade*: eles devem selecionar padrões e procedimentos adequados e adaptá-los à especificação do projeto de software.
3. *Controle da qualidade*: eles devem assegurar que padrões e procedimentos sejam seguidos pela equipe de desenvolvimento de software.

A garantia de qualidade se preocupa em definir quais os objetivos da organização para alcançar a qualidade. Esta atividade envolve a definição ou seleção de padrões que deveriam ser aplicados no desenvolvimento do processo ou produto de software. Após esta seleção ou definição, estes padrões devem ser embutidos nos procedimentos ou processos que serão aplicados durante o desenvolvimento. Deve-se tomar o cuidado de verificar se estes processos serão suportados pelas ferramentas existentes, que foram compradas ou desenvolvidas especialmente durante o processo de garantia de qualidade.

A meta do controle de qualidade é supervisionar o processo de desenvolvimento para assegurar que os padrões e procedimentos de garantia de qualidade estejam sendo seguidos. O processo de controle de qualidade possui seus próprios conjuntos de procedimentos e relatórios que devem ser definidos e aplicados durante o processo de desenvolvimento.

Existem vários padrões que podem ser utilizados no desenvolvimento de um sistema de gerenciamento de qualidade e outros para avaliação deste gerenciamento. A série ISO 9000, se destaca dentre as mais utilizadas em diversos tipos de organizações. A ISO 9001 [NBR94b] é o mais genérico destes padrões e é aplicada em organizações que projetam, desenvolvem e mantêm produtos. Para especificar as características de qualidade de software foi desenvolvido em 1993 um guia para aplicação da ISO 9001 ao desenvolvimento, fornecimento e manutenção de software denominada ISO 9000-3 [NBR93]. Mais detalhes sobre estes padrões serão apresentados no capítulo seguinte.

O planejamento da qualidade deveria ser iniciado no começo do estágio no processo de software. O plano da qualidade deverá expor os requisitos de qualidade do produto, identificando quais atributos possuem maior importância para o produto que está sendo desenvolvido. Por exemplo, pode ser que a eficiência seja primordial e que, conseqüentemente outros fatores tenham que ser sacrificados para alcançá-la. Ao passo que, se este requisito constar nos planos, obviamente os engenheiros terão que segui-lo para alcançar o fator mais importante. Outra atividade do plano de qualidade está em definir um processo de avaliação de qualidade, para verificar se os procedimentos utilizados surtiram algum efeito positivo no desenvolvimento do software, que normalmente seria refletido nas atividades de manutenção.

Administrar os processos de gerenciamento da qualidade é uma atividade que toma um tempo considerável do gerente. Deste modo, sugere-se que a equipe de

gerenciamento da qualidade seja separada da equipe de gerenciamento de projeto. No entanto, o gerente de projeto de um determinado projeto, pode ser o gerente da qualidade de outro projeto. Esta separação – transferência de responsabilidades de gerenciamento de qualidade para uma equipe independente – além de diminuir uma possível sobrecarga dos gerentes, visa o não comprometimento do cronograma e dos custos dentro do projeto, pois, na implantação dos planos de qualidade, há uma alteração no cronograma e nos custos, que muitas vezes pode ser considerável (por exemplo, na adaptação organizacional e primeira implantação do sistema). Isto pode levar o gerente (gerente da qualidade e processo) a desconsiderar os planos de qualidade para atender a menores custos e cronogramas requeridos pelo cliente. De outra forma, considerando equipes diferentes para gerenciamento do projeto e gerenciamento da qualidade, haverá um grande esforço por parte dos gerentes de qualidade em manter o planejamento, pois são responsáveis somente por garantir a qualidade. Assim sendo, buscarão novas estratégias para atender aos requisitos do cliente (custo e cronograma menores).

A garantia de qualidade de software (SQA) compreende uma variedade de tarefas associadas a sete grandes atividades como definido acima por Crosby e melhorado por Pressman, que incluiu uma sétima atividade relacionada a manutenção de registros e reportagem (documentação), que serão descritas nas próximas sessões.

3.1.2.3 Atividades de Garantia de Qualidade de Software (SQA)

As atividades de SQA incluem todas as fases do ciclo de desenvolvimento de software e atividades de gerenciamento da qualidade.

Como foi dito acima, na sessão de garantia de qualidade de software (3.1.2.2), as atividades de desenvolvimento de software por meio dos métodos e ferramentas voltadas ao gerenciamento do projeto objetivam, quando utilizadas adequadamente, à qualidade de software. Conclui-se então que a qualidade de software é projetada num produto ou sistema, e não, imposta após o fato[PRE95]. A SQA inicia-se de fato em paralelo com os métodos técnicos. Assim que as primeiras fases (especificação/protótipo e projeto) dos métodos técnicos tiverem sido criados, há a necessidade de uma avaliação para averiguar a qualidade. Esta avaliação pode ser feita por meio de uma revisão técnica formal (*Formal Technical Review – FTR*). Trata-se de

um encontro estilizado realizado pelo pessoal técnico com o propósito único de descobrir problemas da qualidade [PRE95]. A atividade de revisão pode ser, também, aplicada após o término de cada fase do gerenciamento de projeto, dependerá da estratégia de planejamento montada pela equipe de qualidade ou do método (padrões) de avaliação de processos utilizado para o gerenciamento da qualidade.

A atividade de teste de software combina uma estratégia de múltiplos passos com uma série de métodos de projeto de casos de testes que ajudam a garantir uma detecção de erros efetiva. Alguns desenvolvedores pressupõem que uma boa atividade de teste revelará a maioria dos erros, abrandando a necessidade de outras atividades SQA [PRE95]. No entanto, em algumas literaturas, a realidade mostra que mesmo bem realizada, a atividade de teste não é tão efetiva quanto gostaríamos que fosse para todas as classes de erros [JON81].

Os padrões e procedimentos formais aplicados no processo de engenharia de software variam de empresa para empresa com relação aos tipos e grau que são aplicados. Em muitos casos os padrões são determinados pelos clientes, devido aos requisitos exigidos no contrato elaborado; por imposições reguladoras ou auto-impostos pelo gerenciamento de qualidade. A partir do momento da existência de padrões formais (escritos), há a necessidade de estabelecer uma atividade SQA para garantir que eles sejam cumpridos. Uma avaliação do cumprimento dos padrões pode ser feita como parte de uma revisão técnica formal, ou, em situações em que a verificação independente do atendimento aos padrões for exigida, o grupo SQA pode realizar sua própria auditoria [PRE95].

As mudanças realizadas para a melhoria do software, por solicitação do cliente ou verificação do gerente de qualidade, são bidirecionais no que diz respeito a seus efeitos colaterais. Em primeira instância ela é vista como uma atividade benigna, na medida em que ela contribui diretamente para a qualidade do software ao formalizar pedidos de mudança (atendendo solicitações de clientes, por exemplo). No entanto, verifica-se a necessidade do controle destas mudanças, pois, elas possuem um grande potencial para introduzir erros ou criar efeitos colaterais que propagam erros, influenciando negativamente o projeto. E, uma medida que seria para melhorar a qualidade do software, pode vir a degrading com todo o trabalho realizado até o momento. O

controle de mudanças é aplicado durante a fase de desenvolvimento do software e, posteriormente, durante a fase de manutenção do software.

A medição é uma atividade que faz parte de qualquer disciplina de engenharia, e a pouco, como vimos nas sessões anteriores, têm sido muito requisitada para os produtos de software. As métricas são realizadas para avaliar o impacto das mudanças metodológicas e procedimentais da qualidade dentro do desenvolvimento do produto ou processo. As métricas de software abrangem um amplo conjunto de medidas técnicas e orientadas para a administração [PRE95], como será visto mais adiante.

A anotação e manutenção de registros (documentação) para a garantia de qualidade de software oferecem procedimentos para a coleta e disseminação de informações de SQA. O resultado de revisões, auditorias, controle de mudanças, testes e outras atividades SQA devem tornar-se parte do registro histórico de um projeto e devem ser levados ao conhecimento do pessoal de desenvolvimento, tendo-se como base a necessidade de conhecimento.

Os tópicos aqui relacionados (FTR, testes, padrões e procedimentos formais, controle de mudanças e documentação) serão discutidos com maiores detalhes nas próximas sessões.

3.2.3 Revisões de Qualidade

As revisões de software são comparadas em muitas bibliografias como sendo um “filtro” para o processo de engenharia de software. Esta comparação é levado a efeito, porque esta é uma atividade que pode ser utilizada ao findar cada fase do desenvolvimento de software, identificando possíveis defeitos que possam ser eliminados antes de iniciar a próxima etapa. Freedman e Weinberg [FRE90, PRE95] discutem a necessidade de fazer revisões da seguinte maneira:

O trabalho técnico precisa de revisão pelo mesmo motivo que os lápis precisam de borrachas: Errar é humano. A Segunda razão pela qual precisamos de revisões técnicas é que, não obstante as pessoas captem bem alguns de seus próprios erros, grandes erros escapam mais facilmente a quem lhes deu origem do que a outras pessoas. O processo de revisão é, por conseguinte, a resposta à oração de Robert Burns:

Oh, quem nos dera o poder de vermos a nós mesmos da mesma forma que os outros nos vêem

Uma revisão – qualquer revisão – é uma maneira de usar a diversidade de um grupo de pessoas para:

1. Apontar melhorias necessárias ao produto de uma única pessoa ou equipe.
2. Confirmar as partes de um produto em que uma melhoria não é desejada ou não é necessária.
3. Realizar um trabalho técnico com uma qualidade mais uniforme ou, pelo menos, mais previsível do que aquilo que pode ser realizado sem revisões, de forma a tornar esse trabalho técnico mais administrável.

As revisões normalmente envolvem um grupo de pessoas examinando parte ou todo o processo de software, sistema ou documentação a ele associada para descobrir potenciais problemas. As conclusões da revisão são formalmente registradas e enviadas para o responsável pela correção dos problemas [SOM96].

Existem vários tipos diferentes de revisões. Um encontro informal em torno da máquina de café é uma forma de revisão, se problemas técnicos forem discutidos; ou ainda, uma apresentação formal do projeto de software a uma platéia de clientes, à administração e ao pessoal técnico, também se classifica como uma forma de revisão [PRE95]. As revisões técnicas formais são às vezes denominadas de *walkthrough* (percorrer todos os caminhos) - as quais fazem parte do processo de gerenciamento da qualidade.

Revisões de qualidade possuem três funções [SOM96]:

- (1) Uma função de qualidade em que ela é parte dos procedimentos para a garantia da qualidade geral usada na organização.
- (2) Uma função de gerenciamento do projeto, no qual ela fornece informações para a gerência sobre o progresso do processo. Documentos podem estar com o sinal ausente (*signed off*) na revisão indicando que o desenvolvimento adicional pode continuar. No entanto, um documento com sinal ausente, não significa necessariamente que está livre de falhas. Até certo ponto, o gerente pode decidir que problemas identificados durante a revisão são relativamente simples para resolver.
- (3) Uma função de treinamento, em que os engenheiros explicam seus projetos e justificam suas decisões de projeto. Novatos para o projeto ou projetistas que

terão contato com o sistema que está sendo projetado podem assistir a revisão como observadores. Isto oferece uma excelente oportunidade para aprender sobre o projeto do sistema.

Os objetivos gerais das revisões são: detectar de erros e inconsistências e apontá-las para o projetista ou autor do documento. A atividade de revisão é muito abrangente. Sua utilização não está limitada a especificações, projetos ou código. Documentos tais como: modelos de processos, planos de teste, procedimento de gerenciamento de configuração, padrões de processos, e manuais de usuários também podem ser revisados.

Na realidade, o sucesso de uma FTR é estabelecido pela liderança do revisor chefe (líder). Isto se deve ao fato de que é ele que irá planejar, preparando cronograma, orientando os demais membros quanto aos procedimentos a serem realizados (fora o treinamento, pelo qual todo revisor deve passar); e conduzir, que é uma atividade que requer muita perícia, isto é, experiência, muitas vezes um certo nível de obstinação, para que não haja dispersão e aptidão, pois muitas bibliografias defendem que ser líder é uma característica nata, isto é, não é adquirida com o passar dos anos, se nasce com ela. Apesar desta afirmação ser muito radical, mesmo assim, afirmações de que existem indivíduos que nunca seriam líderes por mais treinamento que possam ter, são válidas.

3.1.3.1 Reuniões de Revisão

Independentemente do formato da revisão técnica formal escolhida, três considerações podem ser feitas para aumentar o desempenho das revisões [PRE95]. A primeira, sugere que a equipe de revisão não deve ser muito grande, deverá se limitar a uma quantidade de três a cinco pessoas. Particularmente, os membros poderão ter funções diferentes dentro da equipe. Basicamente, possuem duas tarefas diferentes para os membros: liderança da equipe, registro da atividade. Na realidade todos são revisores, mas no mínimo dois revisores possuem estas funções extra. Pelo menos um membro será o líder da equipe, que deverá se responsabilizar pelas decisões a serem tomadas e pela condução da atividade de revisão. Outro membro deverá se responsabilizar pela documentação da atividade, tomando nota de todas as decisões aprovadas ou não pelo grupo.

Outra consideração se refere à necessidade de uma preparação antecipada para a revisão, para que os integrantes da equipe de revisão possam ler e entender os documentos. Cada membro então, deverá fazer anotações de seus comentários sobre o que será revisado. No entanto, ela não deve consumir mais que duas horas de trabalho por pessoa. Todos os comentários anotados deveriam ser considerados e poderiam estar classificados em uma das três categorias abaixo, após a revisão estar concluída:

- (1) *Sem ação*: o comentário estava realmente incorreto ou a equipe decidiu que o custo de corrigir o problema não era justificável.
- (2) *Referir para corrigir*: a revisão detectou o problema e pediu que o originador do documento ou projeto arrume esse problema.
- (3) *Reconsiderar todo o projeto*: o melhor caminho para resolver um problema identificado é mudar todo o projeto de sistema ou processo. O líder da revisão normalmente estabelecerá uma reunião entre os engenheiros envolvidos para reconsiderar o problema.

E, a última, revela que a duração da reunião de revisão deveria normalmente ser curta, um período inferior a 2 horas. Apesar disso, se for possível, o líder deveria percorrer todo o documento com a equipe de revisão (*Walkthrough*).

A reunião de revisão deve concentrar-se numa parte específica (e pequena) do software global. Por exemplo, em vez de tentar revisar todo o projeto, *walkthroughs* são levados a efeito para cada módulo ou pequeno grupo de módulos. Ao estreitar o foco, a FTR tem uma probabilidade maior de revelar erros.

Os passos da reunião de revisão são [PRE95]:

1. O desenvolvedor do produto a ser analisado informa ao líder do projeto que o produto está completo e que uma revisão é necessária.
2. O líder do projeto entra em contato com o líder de revisão, que avalia o produto quanto ao seu acabamento, gera cópias de materiais sobre os produtos e as distribui a dois ou três revisores para uma preparação antecipada.
3. O líder de revisão também revê o produto e estabelece uma agenda para o encontro de revisão, que é programada tipicamente para o dia seguinte.
4. O encontro conta com a participação do líder de revisão, todos os revisores e o produtor. Um dos revisores assume o papel de secretário, ou seja, a pessoa que

registra (por escrito) todas as questões importantes levantadas durante a revisão.

5. A FTR inicia-se com uma discussão da agenda e com uma breve introdução feita pelo produtor.
6. O produtor passa então a “caminhar através” do produto, explicando o material, enquanto os revisores levantam questões baseadas na preparação antecipada.
7. Quando erros válidos são descobertos, o secretário anota-os.
8. Ao final da revisão, todos os participantes da FTR devem decidir se (1) aceitam o produto sem modificações adicionais; (2) rejeitam o produto devido a erros graves (uma vez corrigidos, outra revisão deve ser realizada); (3) aceitam o produto provisoriamente (erros menores foram encontrados e devem ser corrigidos, mas nenhuma revisão adicional será exigida). Sendo a decisão tomada, todos os participantes da FTR preenchem um documento indicando sua participação na revisão e sua contribuição com as descobertas da equipe de revisão.

3.1.3.2 Diretrizes de Revisão

As diretrizes de uma FTR devem ser planejadas antecipadamente, pois é fato que uma revisão descontrolada muitas vezes pode ser pior do que nenhuma revisão.

Os seguintes dados representam um conjunto mínimo de diretrizes para as revisões técnicas formais [PRE95]:

1. Revise o produto, não o produtor. Uma FTR envolve pessoas e egos. O líder da equipe deve conduzir o encontro de revisão de maneira que os erros sejam apontados gentilmente, com um tom descontraído e construtivo; a intenção não deve embaraçar nem menosprezar o produtor. Ao menor sinal de que a revisão tenha saído do controle, ela deverá ser interrompida;
2. Fixe e mantenha uma agenda;
3. Limite o debate e a refutação;
4. Enuncie as áreas problemáticas, mas não tente resolver cada problema anotado;
5. Faça anotações por escrito;

6. Limite o número de participantes e insista numa preparação antecipada;
7. Desenvolva uma lista de conferência (checklist) para cada produto que provavelmente será revisto;
8. Atribua recursos e uma programação de tempo para as FTRs;
9. Realize um treinamento significativo para todos os revisores;
10. Reveja suas antigas revisões.

3.1.4 Atividades de Teste

Atividades de testes são consideradas como um elemento crítico da garantia de qualidade de software e representa a última revisão de especificação, projeto e codificação. Ela envolve as seguintes atividades: planejamento, projeto de casos de teste, execução dos casos de teste e análise (avaliação) dos resultados dos testes. O planejamento da atividade de teste deve resultar num Plano de Teste, no qual são estimados recursos e são definidas estratégias, métodos e técnicas de teste, caracterizando-se um critério de aceitação do software em desenvolvimento. Quanto ao projeto de casos de teste, o ideal seria gerar casos de teste para testar exaustivamente o software, mas isso é impraticável devido a restrições de tempo e custo [COL99, YOU90].

É comum que as atividades de testes possam representar 40% do esforço de projeto total, apesar de ter sido seguido cuidadosamente todas as fases do processo de desenvolvimento de software.

Existem muitas técnicas e estratégias de testes de software no mercado, cada uma relacionada ao tipo de projeto, em como ele foi desenvolvido e ao estado em que se encontra o desenvolvimento do sistema, já que estas atividades podem ser implantadas em qualquer fase do processo de desenvolvimento de software. Algumas técnicas e estratégias de testes são apresentadas na Quadro 4.

Os testes, de maneira grosseira, são tidos como uma atividade “destrutora”, pois o processo de desenvolvimento inicia-se com o engenheiro tentando construir software cercado de cuidados para não cometer erros. Com a atividade de teste, o engenheiro cria uma série de casos de teste na tentativa de “demolir” o software que ele construiu. De fato, a atividade de teste é um passo do processo de engenharia de software que poderia ser visto (psicologicamente, pelo menos) como destrutivo, em vez de

construtivo [PRE95]. Mas, no entanto, na moral da história, estas atividades são construtivas ao pensarmos que se elas não existissem, nossos esforços seriam muitas vezes maior na fase de manutenção, e a incredibilidade perante ao usuário sim, seria um fato “demolidor”. Além disso, com esta atividade podemos aprender com os erros encontrados, tentando uma melhoria e ajustando o processo de desenvolvimento, produzindo padrões com as experiências obtidas.

Quadro 4 – Principais técnicas e estratégias de testes de softwares

Técnicas de testes de software	
Tipos de testes	Descrição
Teste de caixa branca	Verificar internamente os detalhes procedimentais. Testa os caminhos lógicos do sistema minuciosamente.
Teste de estrutura de controle	Verifica o comportamento das estruturas de controle, isto é, põe à prova as condições lógicas, caminhos lógicos e a validade das construções de laços contidas num módulo ou programa. Existem muitas variações de teste de estrutura de controle, dentre elas destacam-se os testes de caminho básicos, teste de condição, teste de fluxo de dados e testes de laços.
Teste caixa preta	Refere-se aos testes que são realizados nas interfaces dos softwares. Os testes são realizados com dados externos (entradas e saídas) sem se preocupar muito com a estrutura lógica interna do software. Alguns exemplos de métodos desta técnica são: o particionamento de equivalência, análise de valor limite, técnicas de grafo de causa-efeito e testes de comparação.
Estratégias de testes de software	
Tipos de estratégias	Descrição
Teste de unidades	Este teste se concentra no esforço de verificação da menor unidade de projeto de software – o módulo. Baseia-se na técnica caixa branca.
Teste de integração	É uma técnica sistemática para a construção da estrutura de programa, realizando-se, ao mesmo tempo, testes para descobrir erros associados a interfaces. Pois, após o teste das unidades, há a necessidade de uma integração destas unidades individuais e conseqüentemente um teste para verificar se esta integração foi adequadamente realizada, e se não projetou novos erros. Exemplo de abordagens utilizadas para a construção de estruturas de programas são as integrações top-down e bottom-up.

Estratégias de testes de software	
Tipos de estratégias	Descrição
Teste de validação	São testes executados após os testes de integração, onde o software está completamente montado como um pacote, erros de interface foram descobertos e corrigidos. Esta tarefa é realizada por meio de uma série de testes de caixa preta que deverão demonstrar a conformidade com os requisitos.
Teste de sistema	Um nível mais abstrato de teste se encontra nos testes de sistemas, onde outros elementos do sistema como o novo hardware, informações e uma série de testes de integração e validação foram levados a efeito. O teste de sistema é, na verdade, uma série de diferentes testes, cujo propósito primordial é pôr completamente à prova o sistema baseado em computador, verificando se todos os elementos do sistema foram adequadamente integrados e realizam as funções atribuídas. Incluem-se dentre os testes executados para esta tarefa os testes de recuperação, de segurança, de estresse e de desempenho.

Fonte: [SOM96, PRE95, YOU90].

3.1.5 Atividades de Padrões e Procedimentos

Um dos mais importantes papéis da equipe de garantia de qualidade é o desenvolvimento de padrões de processos e produtos. Padrões de produto definem características que todos os componentes de produtos deveriam exibir; padrões de processos definem como o processo de software deveria ser conduzido. Um exemplo de um padrão de produto é uma forma de revisão que define a informação a ser coletada durante a revisão. E, um exemplo de padrão de processo é uma definição procedural de como a revisão de projeto deveria ser organizada. Mas adiante, será relatado com mais detalhes a implicações de qualidade de produto e processo.

Padrões são importantes por inúmeras razões [SOM96]:

- (1) Eles fornecem um melhor encapsulamento, no mínimo mais apropriado e prático. O conhecimento para a produção de um padrão é adquirido após muitas tentativas e erros organizadas de modo a não repetir erros passados.
- (2) Ele fornece a base sobre a qual o processo de garantia de qualidade será implementado. Visto que, os padrões possuem um encapsulamento mais prático, a garantia de qualidade torna-se uma atividade que deverá assegurar que os padrões sejam seguidos apropriadamente.

(3) Eles ajudam na continuidade onde o trabalho realizado por uma pessoa é conduzido e continuado por outra. Padrões asseguram que todos os engenheiros dentro de uma organização adotem as mesmas práticas. Conseqüentemente, o esforço de aprender quando estiver iniciando um novo projeto é reduzido.

Existem muitos padrões produzidos por diferentes corporações nacionais e internacionais. A realidade, é que a dificuldade e o tempo gasto é muito grande no seu desenvolvimento. E estas corporações, normalmente, se ocupam exclusivamente deste tipo de produção. Dentre as corporações mais conhecidas do ramo estão a US DoD, ANSI, BSI, NATO, IEEE, ISO e outros.

Construir padrões de processos e produtos, toma tempo e é necessário anos de experiências do tipo tentativa e erro. A melhor maneira de implanta-los em uma organização sem que esta sofra muito com as mudanças necessárias, é partir para a utilização de padrões já elaborados, e obviamente testados em muitas organizações, comprovando sua eficácia. Ainda assim, existem vários problemas: um deles, diz respeito à escolha do padrão correto para determinada organização e, outro se encontra em executar os padrões, pois estes estão num nível genérico e não específico, o que deixa a maior responsabilidade na adaptação que será feita pela equipe de qualidade.

Estes padrões abrangem assuntos tais como: engenharia de software, linguagens de programação, notações (símbolos gráficos, procedimentos para derivar e escrever requisitos de software), procedimentos de garantia de qualidade e processos de verificação e validação [IEE94].

Uma outra questão que poderia ser levada a efeito é a relutância dos engenheiros de software na utilização de padrões. Eles alegam, muitas vezes, que padrões são burocráticos e irrelevantes para a atividade técnica de engenharia de software. Isto é particularmente provável, quando os padrões requerem preenchimento de formulários tediosos e registro de trabalho. O fato é que, os desenvolvedores destes padrões estão muitas vezes afastados do processo de desenvolvimento de software e que estão desatualizados quanto às práticas modernas.

Para evitar estes problemas, a organização de garantia de qualidade que estabelece os padrões deve estar adequadamente equipada e deve levar em conta os seguintes passos [SOM96]:

- (1) Envolver engenheiros de software no desenvolvimento de padrões de produtos. O que levaria a um entendimento e confiança nos padrões por parte dos engenheiros.
- (2) Revisar e modificar padrões regularmente para refletir mudanças tecnológicas. Não basta ter um manual de padrões eles precisam se desenvolver de acordo com a evolução tecnológica.
- (3) Disponibilização de ferramentas de apoio para formatar documentos, por exemplo, padrões de escritório são as causas de muitas queixas por causa dos trabalhos tediosos envolvidos na sua implementação.

Gerentes de projeto e de qualidade podem evitar muitas dificuldades, simplesmente adequando os padrões no início do projeto quanto às circunstâncias individuais. Entretanto, padrões que relatam a qualidade do produto e o processo de pós-entrega deveriam ser mudados somente após cuidadosa consideração.

3.1.5.1 Padrões de Documentação

Os padrões de documentação são uma forma concreta de representar produto e processo de software. Documentos produzidos com padrões apropriados tem uma aparência, estrutura e qualidade compatíveis.

Há três tipos de padrões de documentação [SOM96]:

- (1) Padrões de processo de documentação. Estes padrões definem os processos os quais deveriam ser seguidos para produção da documentação.
- (2) Padrões de documentos. Estes são padrões que regulamentam os próprios documentos.
- (3) Padrões de troca/intercâmbio de documentos. Este padrão refere-se a importância da troca de cópias de documento cada vez mais importantes feitas por meio de correio eletrônico e a armazenagem de documentos em base de dados.

Procedimentos envolvidos no desenvolvimento de um documento e as ferramentas utilizadas para a produção deste documento, refere-se a uma padronização de processo de documentação. Nesta fase também há a necessidade da definição de procedimentos de checagem e refinamento que assegurem que os documentos possuem alta qualidade.

Os padrões de documentação construídos devem suportar todos os tipos de documentos e as checagens utilizadas no processo de qualidade formal, devem ser restringidas a documentos formais tais como: os usados em um novo desenvolvimento ou aqueles que serão liberados para clientes.

O padrões de documentos são aplicados em todos os documentos produzidos no decorrer do desenvolvimento do software. A aparência e a estrutura deve ser compatível entre classes de mesmo tipo, obtendo um “estilo da casa” ao longo do processo. Mesmo em situações em que os padrões devam ser adaptados às necessidades específicas do projeto, é melhor manter sempre o mesmo estilo.

Exemplos de padrões de documento que podem ser facilmente desenvolvidos são [SOM96]:

- (1) Padrões de identificação de documentos. Em meio a grandes projetos, que produzem milhares de documentos, há a necessidade de que cada documento seja unicamente identificado.
- (2) Padrões de estrutura de documentos. Cada classe de documentos produzidos durante o projeto de desenvolvimento de software deveria seguir alguma estrutura padrão, definindo, por exemplo, as convenções usadas para numeração de página, cabeçalho de página, rodapé e outros.
- (3) Padrões de apresentação. Definem o “estilo da casa” quanto a: fontes e estilos usados, logotipos e nomes de companhias, o uso de cores para destacar a estrutura do documento e assim por diante.
- (4) Padrões de atualização de documentos: Quando um documento é mudado para refletir mudanças no sistema, uma maneira compatível de indicar estas mudanças deveria ser utilizada. Um exemplo, seria a mudança de diferentes cores de capas para indicar uma nova versão do documento ou o uso de barras de mudança para indicar a mudança em um parágrafo.

E, por último, os padrões de intercâmbio de documentos os quais antigamente eram enviados de um engenheiro para o outro em forma de cópias de papel, atualmente são realizados eletronicamente. O uso de padrões de intercâmbio permite que documentos que sejam transferidos eletronicamente sejam recriados em sua forma original.

Exemplos de padrões de intercâmbio incluem o uso de processadores de texto compatíveis, se necessário com uma limitação de fontes e estilos de textos usados por causa das diferenças entre as capacidades de impressoras e monitores de vídeo e outros. O padrão mais utilizado é o formato RTF, devido à sua ampla capacidade de compatibilidade com outros sistemas, desenvolvido pela Microsoft. Um outro, que se encontra cada vez mais adentrando no mercado é o Adobe Acrobat, usado para visualizar formato texto.

3.1.6 Anotação e Manutenção de Registros (documentação)

Existem vários aspectos positivos em se manter um sistema de documentação durante o processo de desenvolvimento do software. O principal deles, se encontra na grande ajuda no momento da manutenção do produto desenvolvido. O sistema de documentação é uma prática simples, até certo ponto, sob o ponto de vista que é apenas uma prática de anotar todos os passos que foram dados durante a implementação do sistema, especificação dos requisitos e ao final os planos de testes de aceitação. No entanto, não basta que seja feita uma documentação de acordo com padrões, o ponto crítico está em mantê-la atualizada devido às constantes solicitações de mudanças como serão apresentadas na sessão seguinte.

A documentação do sistema deveria ser estruturada, com visões gerais destacando o leitor com descrições mais formais e detalhadas de cada aspecto do sistema. É importante que os documentos sejam claros e legíveis, de outra maneira, eles não serão usados [SOM96].

Um benefício significativo são as ferramentas CASE que já possuem um suporte para o desenvolvimento da documentação aliado ao processo de desenvolvimento poupando tempo e esforços a baixos custos.

3.1.7 Controle de Mudanças

As mudanças expressivas dentro de sistemas de software não referem-se somente a solicitações de clientes, mas também às mudanças refletidas por necessidades organizacionais e conseqüentemente mudança de requisitos durante o ciclo de vida de um sistema [SOM96].

O controle de mudanças combina procedimentos humanos e ferramentas automatizadas. O maior problema em controlar mudanças está em projetar e analisar os custos e benefícios que serão alcançados ou não por elas. O planejamento de mudanças parte de uma solicitação para que algo seja modificado dentro do ciclo de desenvolvimento ou da manutenção do sistema. Elas então serão aprovadas, implementadas e validadas.

O primeiro passo para o processo de gerenciamento de mudanças é o preenchimento de um formulário de requisição de mudança (CRF – *change request form*). Após a requisição da mudança o passo seguinte é a análise da mudança requerida quanto ao mérito técnico, potenciais efeitos colaterais, impacto global sobre outros objetos de configuração, funções do sistema e o custo projetado da mudança. Os resultados então são analisados e, caso as mudanças sejam solicitações válidas, avaliações para a realização das mudanças são estimadas e as solicitações de mudanças são registradas na base de dados. Caso as mudanças sejam aceitas, então elas serão efetuadas, registradas e ligadas a solicitações de mudanças associadas, e submetidas a aprovação requerida pela qualidade. Este processo, após aceitação das mudanças, é executado até que a qualidade de software seja adequada. Uma nova versão do sistema é criada.

Uma vez que um formulário de solicitação de mudança foi submetido, há algumas conclusões que se pode chegar quando uma análise de mudança é avaliada como inválida [SOM96]:

1. Não se trata de uma falha de sistema, e sim, de um mal-entendido cometido pelo usuário;
2. A mudança trata-se de uma falha duplicada ou que já tenha sido referenciada.

Nestes casos a mudança deveria ser rejeitada. O processo acima demonstrado é um dos muitos exemplos de planejamento de mudanças que podem ser realizados.

3.1.8 Métricas

A medição é uma atividade que faz parte de qualquer disciplina de engenharia, e a pouco, como vimos nas sessões anteriores, tem sido muito requisitada para os produtos de software. As métricas são realizadas para avaliar o impacto das mudanças

metodológicas e procedimentais da qualidade dentro do desenvolvimento do produto ou processo, e ainda sua documentação relacionada. Exemplos de medidas são o tamanho de um produto em linhas de código, o Fog index [GUN62], que é uma medida de legibilidade de um manual do produto, o número de falhas relatadas em um produto de software entregue e o número de pessoas-dia requeridas para desenvolver um componente [SOM96, PRE95]. No entanto, apesar da quantidade de métricas e medidas existentes, a subjetividade de medir software torna o controle extremamente difícil de ser sistemático e apesar da “precisão” dos instrumentos, ainda assim nos deparamos com uma grande subjetividade.

Toda esta preocupação em medir a qualidade de software se encontra nos novos conceitos dos usuários cada vez mais exigentes e, para manter um nível mínimo aceitável dentro da competitividade do mercado atual.

As métricas de software abrangem um amplo conjunto de medidas técnicas que será visto mais adiante, no próximo capítulo.

3.1.9 Qualidade do Produto x Qualidade do Processo

Uma das evoluções mais importantes dentro dos estudos da qualidade foi a mudança de posicionamento em relação à verificação da qualidade do produto produzido para o processo de produção. Antigamente, a qualidade se restringia na garantia do próprio artesão de que o produto por ele confeccionado possuía boa qualidade. Com o passar dos anos, o aumento populacional e, conseqüentemente aumento na produção, este passou a não ser um método viável, pois não existia uma ou duas pessoas confeccionando os produtos, mas centenas ou milhares, e ao invés da produção atingir a marca de 5 produtos por dia, atingiria 500 ou mais, se tornando impossível a revisão individual de todos os produtos por meio, talvez, de um gerenciamento individual. Várias medidas foram tomadas para a resolução destes problemas no decorrer dos anos (Quadro 5). E a medida que mais contribuiu foi o controle do procedimento de produção, isto é, do processo de produção. Esta afirmação está baseada na suposição de que o fator chave influenciando a qualidade do produto está na qualidade do processo de desenvolvimento do produto [SOM96]. Estas idéias de melhoria do processo originaram-se do trabalho de um engenheiro americano chamado W. E. Deming que trabalhou em uma indústria japonesa após a Segunda Guerra

Mundial para melhoria da qualidade. Deming (e outros) introduziram a idéia de controle de qualidade estatística [SOM96], isto é, baseado na medida do número de defeitos em um produto e relacionar estes defeitos em um processo. O processo passa por uma melhoria para reduzir o número de defeitos do produto, que se repete até que o resultado do processo seja prognosticável e o número de defeitos reduzidos. O processo será então padronizado e iniciará um novo ciclo de melhoria.

No processo de fabricação a relação de processo/produto é muito óbvia. Melhorar o processo evitando-se defeitos logicamente conduzirá a produtos melhores. No entanto, este fato não é tão óbvio quando o produto é intangível e dependente, até certo ponto, de um processo intelectual que não pode ser automatizado. Qualidade de software não é dependente de um processo de fabricação, mas de um processo de projeto onde o elemento humano é muito importante [SOM96]. Em algumas classes de produtos, o processo utilizado é o mais importante determinante da qualidade do produto e em outros as pessoas envolvidas no processo podem ser mais importantes do que o processo usado, especialmente em casos onde novas aplicações serão utilizadas.

Quadro 5 - Evolução no posicionamento da qualidade do produto x qualidade do processo

Ano	Ação	Descrição
1900	Inspeção pós-produção	Avalia o produto final, depois de pronto
1940	Controle estatístico da produção	Avalia os subprodutos das etapas de produção
1950	Procedimento de produção	Avalia todo o procedimento de produção
1960	Educação das pessoas	Avalia as pessoas envolvidas no processo
1970	Otimização dos processos	Avalia e otimiza cada processo
1980	Projeto robusto	Avalia o projeto de produção
1990	Engenharia simultânea	Avalia a própria concepção do produto

Fonte: [BAR97].

Para produto de software assim como em outras atividades intelectuais tais como: livros, filmes e outros, onde qualidade depende principalmente do projeto, existem quatro fatores que podem afetar a qualidade do produto, como são apresentados na Figura 2 [SOM96].

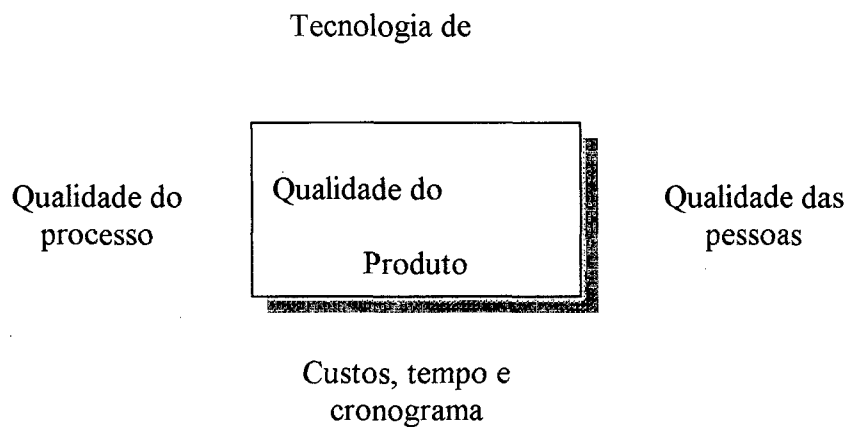
A influência de cada um destes fatores depende do tamanho e do tipo de projeto. Para grandes sistemas que são compostos de sub-sistemas separados, desenvolvidos por diferentes equipes, o principal determinante da qualidade do produto é provavelmente o

processo de software. O maior problema nestes grandes projetos é a integração, o gerenciamento de projeto e as comunicações.

Para pequenos projetos, entretanto, há poucos membros na equipe então, a qualidade da equipe de desenvolvimento, provavelmente, é mais importante do que o processo de desenvolvimento usado. Se a equipe possui um alto nível de habilidade e experiência, a qualidade do produto provavelmente será alta. Se a equipe é inexperiente e não apta, um bom processo pode limitar o prejuízo mas, ele próprio, não conduzirá à alta qualidade do software.

O nível tecnológico pode ajudar e atrapalhar os desenvolvedores de software. Para equipes pequenas a tecnologia é muito importante, pois seus membros não possuem muito tempo para gastar com tediosos procedimentos administrativos. Assim, boas ferramentas seriam de grande ajuda. Em grandes projetos, um nível básico de desenvolvimento tecnológico é essencial para a informação gerencial. Entretanto, ferramentas são menos importantes neste caso, pois os membros das equipes gastam mais tempo em comunicação e entendimento das partes do sistemas do que com as atividades de desenvolvimento.

Figura 2 – Principais fatores da qualidade de produtos.



E na base do triângulo encontra-se um fator absolutamente crítico, pois se um projeto independente do tamanho, é sub-orçado ou planejado com um cronograma de entrega não-realístico, a qualidade do produto será afetada. Um bom processo requer recursos para uma implementação efetiva. Se os recursos são insuficientes, o processo

não pode trabalhar efetivamente. Se os recursos são inadequados, somente pessoas excelentes podem salvar um projeto.

No entanto, muitas vezes, a causa de problemas na qualidade do software não é devido ao pobre gerenciamento, processos inadequados ou a pobre qualidade no treinamento. Até certo ponto, é o fato de que organizações devem competir para sobreviver. Muitos projetos de software são deliberadamente sub-orçados para ganhar a concorrência contratual para desenvolvê-lo. “Preço-para-ganhar” é uma inevitável consequência da competitividade do sistema. Então, não será surpresa se a qualidade do produto baixar.

3.1.10 Certificação de Qualidade

Atualmente a palavra certificação tem sido muito utilizada entre as organizações. A certificação é a emissão de um documento oficial indicando a conformidade com uma determinada norma ou padrão. Portanto, para que haja uma certificação, há a necessidade de requisitos que estejam acordados nacional e internacionalmente por órgãos competentes, e processos para avaliar a execução destes requisitos, é o que denominamos de normas. Existem muitas normas, cobrindo uma gama variada de produtos, atividades ou serviços. A série ISO 9000 [ISO 9000], é um exemplo de normalização. Ela é um conjunto de normas internacionais para sistemas de gerenciamento da qualidade, desenvolvido pela ISO (International Organization for Standardization) e foi o primeiro acordo sobre os requisitos básicos para sistemas da qualidade com alcance mundial [CAR97].

Para a obtenção da certificação, há a necessidade da realização de todo um processo de avaliação e julgamento de acordo com uma determinada norma que foi implantada em um processo de desenvolvimento de produto, bem ou serviço. Embora uma empresa possa auto-avaliar-se ou ser avaliada por seus próprios clientes, o termo “certificação” costuma ser aplicado apenas quando efetuado por uma empresa independente e idônea, normalmente especializada neste tipo de trabalho. No Brasil, o INMETRO é o órgão do governo responsável pelo credenciamento destas instituições que realizam a certificação de sistemas de qualidade.

Esta certificação é obtida por meio de alguns passos a serem seguidos [BAR97]:

1. A empresa estabelece o seu sistema de qualidade;

2. A empresa faz uma solicitação formal a um órgão certificador, incluindo detalhes do negócio da empresa, escopo da certificação solicitada e cópia do manual de qualidade;

3. O órgão certificador faz uma visita à empresa, colhe mais dados e explica o processo de certificação;

4. O órgão certificador verifica se a documentação do sistema de qualidade está de acordo com a norma ISO;

5. O órgão certificador envia uma equipe à empresa com fins de auditoria. Nesta visita, será verificado se todos na empresa cumprem o que está documentado no manual de qualidade;

6. O órgão certificador emite o certificado de qualidade;

7. O órgão certificador realiza visitas periódicas à empresa para assegurar que o sistema continua sendo efetivo.

Muitas empresas têm utilizado normas para gerenciar e melhorar seus processos e têm obtido resultados positivos. Um exemplo é a empresa POLO de Curitiba, que obteve certificação em 1996, e manteve as características de qualidade de seus produtos obtendo uma recertificação em 1999. Ela se utiliza da série ISO 9000, e complementa-as com o modelo de maturidade CMM(Capability Maturity Model – Modelo de Maturidade e Capacidade) e o ciclo PDCA(Plan, Do, Check, Act), que é realizado periodicamente a tarefa C e A [WEB99b].

O exemplo, acima citado mostra que, para produtos de software não basta somente a certificação de acordo com normas e padrões previamente definidos por um órgão competente, como o que acontece com outros produtos que podem ser medidos objetivamente. Uma solução, se bem conduzida, seriam técnicas de melhoria de processos, que poderão levar além da certificação, estabelecendo um mecanismo de melhoria contínua [CAR97].

3.2 NORMAS E MODELOS DE QUALIDADE

Norma refere-se a um conjunto de regras, normalmente consensuais, ou seja, impostas somente mediante um consenso ou acordo entre as partes. Enquanto os

modelos são moldes que podem ser imitados, que normalmente não são derivativos de acordos consensuais.

3.2.1 Introdução

A grande dificuldade em obter e manter a qualidade dos produtos de software encontra-se em como medi-los. Ao contrário de outras disciplinas de engenharia, onde os produtos gerados apresentam características físicas como o peso, a altura, tensão de entrada, tolerância a erros e outros, a medida da qualidade do software é algo relativamente nova e alguns dos critérios utilizados são questionáveis.

Dentro da literatura especializada, é comum encontrar os termos métrica e medida coexistindo em um mesmo contexto, sendo que, em alguns casos, utilizados de maneira intercambiável [MOH79, HIR82, CIO91]. O termo métrica é utilizado no sentido dos objetivos a serem medidos. Por exemplo, métrica de linhas de código, de pontos por função, de complexidade etc. Por sua vez, o termo medida denota a aplicação de uma determinada métrica para a obtenção de dados quantitativos [LEM97].

As métricas são determinadas a priori dentro de um processo de desenvolvimento, enquanto que os dados que devem ser coletados são obtidos por meio de medidas realizadas durante o processo ou sobre o produto resultante.

De maneira independente de quais métricas se está considerando, há um conjunto de características que podem ser-lhes atribuídas de forma comum. Estas características determinam a maneira com que uma medida pode ser realizada, direta ou indiretamente, e se o dado coletado é objetivo ou subjetivo. Considere, por exemplo, que se deseja medir o número de linhas de código de um determinado software. Esta medida fornecerá um dado objetivo representativo do número de linhas contabilizadas. Contudo, a maneira de obter esse número pode variar. Em uma primeira situação, pode-se realmente contar cada linha constituinte do código. Neste caso, diz-se que a medida foi realizada de maneira direta. Em outra situação o número de linhas pode ser obtido pela contagem de páginas de listagem vezes um fator estatístico que represente a média de linhas de código por página de listagem. Neste caso, diz-se que a medida foi tomada de maneira indireta. Essas características podem ser consideradas acessórios a uma dada métrica, uma vez que dependem muito mais da maneira com que uma determinada medida é realizada do que propriamente da métrica em questão [LEM97].

A metrologia do Software corresponde ao conjunto de teorias e práticas relacionadas com as medidas, a qual permite estimar o desempenho e o custo do software, a comparação de projetos e a fixação dos critérios de qualidade a atingir. No entanto, antes de medir, há necessidade de seguir trajetórias já identificadas como positivas, deixando de correr riscos desnecessários. Tais trajetórias, na realidade, são os estudos realizados a fim de abranger experiências sobre qualidade, são a maioria, voltadas para o melhoramento do processo de desenvolvimento, pretendendo-se com isto, chegar a uma boa qualidade do produto ou serviço [BAR97].

Existem vários modelos e normas utilizadas para a avaliação e implantação de processo de qualidade em produtos, processos de desenvolvimento e melhoria da qualidade organizacional, de características gerais e específicas. Pode-se classificar estes modelos e abordagens de acordo com suas funcionalidades, perante o projeto a ser desenvolvido, melhorado ou avaliado, como pode ser visto a seguir [WEB99]:

1. Abordagens para a melhoria da qualidade total nas organizações. Trata-se do gerenciamento como um todo. Nesta abordagem encaixam-se os métodos: EQA, TQM, Baldrige(USA), QMS, PNQ (Brasil),GQM, PDCA;

2. Abordagens para a melhoria da qualidade dos processos de software. Diz respeito ao processo de qualidade que será implantado em conjunto com os métodos de desenvolvimento de software, para avaliar e garantir a qualidade do processo de desenvolvimento de software, resultando em um produto de qualidade. Pode ser enquadrado nesta abordagem modelos tais como o: CMM, SPICE, TRILLIUM, BOOTSTRAP, PSP, ami;

3. Método de melhoria de qualidade do produto de software. É o resultado das atividades realizadas no processo de desenvolvimento do software. Avaliar a qualidade de um produto de software é verificar, por meio de técnicas e atividades operacionais o quanto os requisitos são atendidos. Tais requisitos, de uma maneira geral, são a expressão das necessidades, explicitados em termos quantitativos ou qualitativos, e têm por objetivo definir as características de um software, a fim de permitir o exame de seu atendimento [TSU97]. Desta categoria fazem parte o: STD, STARTS, TickIT, ISO 9001, ISO 9000-3, ISO/IEC 9126, ISO/IEC 14598, ISO/IEC 12119;

4. Normas da qualidade para o desenvolvimento de software. São atividades definidas e planejadas para o desenvolvimento de software com qualidade. Desta categoria pertencem as seguintes normas: BS 7165, DOD-STD-2167A, DOD-STD-2168, ESA, ISO 12207, ISO 9000-3, AQAP 13, AQAP 14;

5. Normas gerais da qualidade. Estas normas podem ser aplicadas na produção de qualquer produto, bem ou serviço. Nesta abordagem enquadram-se a: a série ISO 9000, AQAP 1, AQAP 2;

Atualmente, muitas instituições se preocupam em criar normas para permitir a correta avaliação de qualidade ou em “fabricar” meios que garantam a qualidade, tanto de produtos de software, quanto de processos de desenvolvimento de software. Pois, apesar da variedade de abordagens existentes, ainda assim, existem problemas na implantação e continuação dos processos. Atualmente os modelos e normas são de características gerais, isto é, seu sucesso depende muito da interpretação, planejamento e implantação das pessoas envolvidas no projeto. Novos padrões, alguns dos quais, também serão apresentados neste trabalho, já estão sendo desenvolvidos para cobrir estas lacunas.

Nas sessões subseqüentes, breves descrições dos principais modelos e normas de cada classe de abordagens serão discutidos.

3.2.2 Modelo TQM – Total Quality Management (Gerenciamento da Qualidade Total)

Gerência pela Qualidade Total (TQM – Total Quality Management), é um sistema que combina técnicas de controle da qualidade e modelos organizacionais desenvolvidos nos últimos 40 anos, tanto nos Estados Unidos quanto no Japão [HAZ99].

A TQM possui as seguintes características, as quais podem ser aplicadas a todas as organizações [KRI96]:

- A qualidade é reconhecida como uma vantagem estratégica com o apoio total da alta-administração. O novo pensamento da qualidade deve ter um enfoque preventivo;

- A gerência da qualidade é dirigida como um passo de melhoria em toda a organização, com seus produtos, processos do negócio, processos de apoio e sistemas de informação;
- Todos os empregados devem estar envolvidos com a qualidade, e serem treinados em métodos e técnicas para a melhora da Qualidade Total;
- A melhora da qualidade de um produto está baseada em uma compreensão e controle sistemático do processo e análise dos resultados organizacionais e do mercado;
- O pensamento da qualidade é mantido ativamente nas organizações por meio de treinamento, sistema de recompensas e *benchmarking* interno e externo.

É importante que se entenda a distinção entre TQC e TQM. Enquanto o primeiro conceito relaciona-se mais enfaticamente à satisfação do cliente ou melhor eficiência e eficácia no relacionamento com o cliente, o conceito de TQM, expande a necessidade de se ter eficácia e eficiência no relacionamento de todos os elementos que compõem o modelo da empresa inserida em um contexto mais amplo [TQM96].

Dessa forma o TQC (Total Quality Control - Controle Total da Qualidade), pode ser definido como um conjunto de atividades, envolvendo toda a empresa, que têm como objetivo assegurar o resultado do empreendimento.

O TQM, compreende o gerenciamento das relações entre todos os envolvidos com a existência da empresa, não se restringindo somente ao relacionamento com o cliente.

Na prática, empresas que aplicam o TQM ou o TQC acabam por desenvolver um conjunto de atividades similares e talvez por isso é que muitos autores confundem os termos. Conceitualmente, podemos considerar que o TQC seja uma parte integrante do TQM [TQM96].

No Brasil, o percentual de empresas com programa de qualidade total ou similar formalmente implantado numa pesquisa realizada em [WEB99], mostra que em 1997, apenas 17% de uma amostragem de 588 empresas possuem um processo implantado, enquanto que o percentual de empresas sem programa da qualidade formalmente implantado não sofreu redução significativa no decorrer dos anos, desde 1993 (Tabela 1).

Tabela 1 – Implantação de Programa da Qualidade Total ou Similar no Brasil.

Categorias	1993		1995		1997	
	Nº	%	Nº	%	Nº	%
Implantado	34	12,4	51	11,5	104	17,7
Em estudo	128	46,5	170	38,3	212	36,1
ou implantação						
Não implantado	113	41,1	223	50,2	272	46,3
Base	275	100	444	100	588	100

Fonte: [WEB99]

3.2.3 Modelo PDCA – Plan-Do-Check-Action (Planejar-Executar-Checar-Agir)

O modelo PDCA (P de “plan”, planejar; D de “do”, executar; C de “check”, verificar e A de “act”, agir corretiva e preventivamente), não é um modelo novo. Ele descreve passos conscientes que a maioria das pessoas tomariam enquanto tomam decisões e resolvem problemas [WEB97, PDC94]. O modelo resumido apresenta-se no Quadro 6:

Quadro 6 – Descrições do Modelo PDCA

FASE	DESCRIÇÃO
PLAN (planejar)	Usando os recursos disponíveis, planejar a melhoria, focando em melhorias específicas que causam um impacto mais positivo no processo. Uma lista de melhorias potenciais é gerada. Após a identificação das melhorias, elas são priorizadas.
DO (executar)	Esta fase envolve desenvolvimento e implementação da melhoria baseada na lista priorizada.
CHECK (verificar)	Medidas contínuas, monitoram o desempenho do processo após a implantação da melhoria.
ACTION (agir)	Decisões são tomadas para continuar ou para a melhoria baseada neste impacto. Se a melhoria é continuada, ela pode ser transferida para processos similares.

Fonte: [PDC94]

O ciclo PDCA continua até que o nível desejado de melhoria seja alcançado.

3.2.4 Modelo GQM - Goal-Question-Metric (Metas-Questões-Métricas)

O método GQM [BAS86], fornece um mecanismo que orienta a determinação dos objetivos de interesse da análise e o refinamento de cada objetivo em um conjunto de questões, com o propósito de quantificar os mesmos. O método pode ser definido resumidamente em três passos básicos, mostrados na Quadro 7:

Quadro 7 - Descrições do Modelo GQM

PASSO 1: estabelecer os objetivos da coleta de dados (Goal – Metas, objetivos)
Objetivo: Os objetivos podem ser definidos em termos de: Objeto de Estudo, Propósito, Perspectiva e Ambiente [BAS90].
PASSO 2: desenvolver uma lista de questões de interesse (Question – Questões)
Objetivo: As questões de interesse permitem uma definição mais nítida do objetivo da pesquisa, pois essas forçam os investigadores a considerar a análise de dados antes que eles sejam coletados [ROM87].
PASSO 3: estabelecer as métricas (Metric – Métricas)
Objetivo: As métrica define as medidas associadas a cada questão para lhe dar resposta quantitativa. Essa abordagem é proposta devido à grande variedade de características observáveis o software, cuja forma de uso e interpretação só se tornam claras a partir da definição de modelos e objetivos apropriados para o contexto.

Fonte: [IAV97]

Este método, assim como outros que serão vistos a seguir, são métodos genéricos que seu sucesso dependem da experiência dos indivíduos que o estão utilizando.

3.2.5. CMM – Capability Maturity Model (Modelo de Maturidade da Capacidade)

Este “Modelo de Maturidade da Capacidade” é uma iniciativa do SEI (Software Engineering Institute – Instituto de Engenharia de Software) da CMU (Carnegie Mellon University – Universidade de Carnegie Mellon) para avaliar e melhorar a capacitação de empresas que produzem software. O projeto CMM foi apoiado pelo Departamento de Defesa do Governo dos Estados Unidos, que é um grande consumidor de software e precisava de um modelo formal que permitisse selecionar os seus fornecedores de software de forma adequada. Embora não seja uma norma emitida por uma instituição

internacional (como a ISO ou o IEEE), este modelo tem tido uma grande aceitação mundial, até mesmo fora do mercado americano.

O CMM, é um modelo para medição da maturidade de uma organização no que diz respeito ao processo de desenvolvimento de software. A definição de “maturidade” pode ser melhor entendida por intermédio da Quadro 8 abaixo:

Quadro 8 – Definição de maturidade do Modelo CMM

Organizações maduras	Organizações imaturas
Papéis e responsabilidades bem definidos	Processo improvisado
Existe base histórica	Não existe base histórica
É possível julgar a qualidade do produto	Não há maneira objetiva de julgar a qualidade do produto
A qualidade dos produtos e processos é monitorada	Qualidade e funcionalidade do produto sacrificadas
O processo pode ser atualizado	Não há rigor no processo a ser seguido
Existe comunicação entre o gerente e seu grupo	Resolução de crises imediatas

Fonte: [BAR97]

Neste modelo, as organizações de software são enquadradas em um dos cinco níveis de maturidade definidos, no qual o nível 1, define uma organização como imatura e o nível 5, como as organizações mais maduras, onde cada detalhe do processo de desenvolvimento está definido, quantificado e acompanhado e a organização consegue até absorver mudanças no processo sem prejudicar o desenvolvimento. A Quadro 9, mostra as características da organização e as áreas chaves de processo necessárias para que ela seja classificada em cada um dos níveis, conforme a versão 1.1 do CMM.

Quadro 9 – Níveis de Maturidade e áreas chave de processo do Modelo CMM

Níveis e Características	Áreas Chaves de Processo
<p>Nível 5 – Otimizado Melhoria contínua do processo é possibilitada pela realimentação quantitativa do processo e conduzida a partir de idéias e tecnologias inovativas.</p>	<p>Prevenção de defeitos Gerenciamento de mudanças tecnológicas Gerenciamento de mudanças de processo</p>

Níveis e Características	Áreas Chaves de Processo
<p>Nível 4 – Gerenciado São efetuadas medições detalhadas do processo de software e qualidade do produto. Tanto o processo como o produto são entendidos e controlados quantitativamente.</p>	<p>Gerenciamento quantitativo do processo Gerenciamento da Qualidade de Software</p>
<p>Nível 3 – Definido O processo de software em relação tanto às atividades de gerenciamento, como de engenharia, são documentadas, padronizadas e integradas em um processo padrão para a organização. Todos os projetos usam uma versão aprovada e adaptada do processo padrão de software da organização para o desenvolvimento e manutenção.</p>	<p>Foco no processo da organização Definição do processo da organização Programa de treinamento Engenharia de produto de software Gerenciamento integrado do software Coordenação entre grupos Revisões.</p>
<p>Nível 2 – Repetitivo Processos básicos de gerenciamento de projeto são estabelecidos para controlar custos, cronogramas e funcionalidade. A disciplina necessária de processo permite repetir sucessos anteriores em projetos de aplicação similar.</p>	<p>Gerenciamento de requisitos Planejamento de projeto de software Acompanhamento de projeto de software Gerenciamento de subcontratos Qualidade assegurada de software Gerenciamento de configuração.</p>
<p>Nível 1 – Inicial O processo de software é caracterizado como desorganizado, ocasionalmente até caótico. Poucos processo definidos. Sucesso depende dos esforços individuais e heroísmo.</p>	

Fonte: [PAU95]

Uma empresa no nível 1 não consegue garantir prazo, custo ou funcionalidade. No nível 2, a empresa já consegue produzir bons softwares, no prazo e a um custo previsível. O nível 3, garante um excelente nível de qualidade, tanto no produto quanto no processo de desenvolvimento como um todo. Não há, no mundo muitas empresas que tenham chegado aos níveis 4 e 5 [BAR97].

Uma pesquisa da CMU, sobre avaliações CMM realizados no período 1992-1997, apresentou o seguinte resultado:

Tabela 2 – Perfil de Maturidade das Organizações no período de 1992 a 1997

Caracterização da Amostra	Perfil de Maturidade das Organizações
<ul style="list-style-type: none"> • 700 organizações • 202 empresas • 3429 projetos • 22,9% das organizações no exterior 	<ul style="list-style-type: none"> • nível 1: 58,1% • nível 2: 23,9% • nível 3: 15,1% • nível 4: 2,3% • nível 5: 0,6%

Fonte: [WEB99]

O CMM teve tal repercussão, que diversos outros se baseiam diretamente nele, como é o caso do Bootstrap e do Trillium, e influenciou fortemente outros como o SPICE, que serão descritos posteriormente. A versão 1.1 está em processo de revisão, devendo denominar-se SW-CMM v2.0 em 1999, e as principais alterações ocorrerão nos níveis 4 e 5 [WEB99].

3.3.6 SPICE – Software Process Improvement and Capability Determination (Melhoria do Processo de Software e Determinação da Capacidade)

Em junho de 1991, o subcomitê de engenharia de software da ISO (International Standards Organization – Organização de Padrões Internacionais) aprovou a realização de estudos para analisar as necessidades e os requisitos de um padrão para avaliação do processo de software.

Como resultado desse estudo, em 1993 foi lançado o SPICE. Este modelo constitui-se de um padrão para a avaliação dos processos de software, que integre os modelos CMM, TRILLIUM, BOOTSTRAP e STD (Software Technology Diagnostic – Diagnóstico de tecnologia de software), visando determinar a capacitação de uma organização. O modelo visa ainda, orientar a organização para uma melhoria contínua do processo [BAR97, PAU94, DIA96].

Em 1995, o projeto SPICE realizou avaliações de processos de software por intermédio de coordenações na América, Europa, Ásia e Oceania e, por um período de 2 anos, o modelo SPICE foi utilizado e ajustado com a prática. Destas avaliações resultou o relatório técnico ISO/IEC TR 15504, em 1997, composto de nove partes. A parte 2 e parte 3 são normativas, as demais informativas [WEB99, DIA96].

A essência do SPICE está na parte 2, onde são apresentadas duas dimensões processos e capacidade. A primeira, diz respeito aos processos envolvidos no desenvolvimento de software e a segunda, é na realidade, seu objetivo principal, que se constitui de avaliar a capacitação da organização em cada processo e permitir a sua melhoria. Os processos compõe-se de cinco categorias (Quadro 10): cliente-fornecedor (CUS), engenharia (ENG), suporte (SUP), gerenciamento (MAN) e organizacionais (ORG). Esta estrutura de processos é compatível com a ISO/IEC 12207. Capacidade compõe-se de seis níveis (Quadro 11): nível 0 – processo incompleto, nível 1 – processo realizado, nível 2 – processo gerenciado, nível 3 – processo estabelecido, nível 4 – processo previsível e nível 5 – processo otimizado[WEB99, BAR97, TSU97].

O SPICE pode ser utilizado por organizações envolvidas em planejar, gerenciar, monitorar, controlar e melhorar a aquisição, fornecimento, desenvolvimento, operação, evolução e suporte de software.

Na da visão do SPICE, a avaliação de processos de software tem como propósito [TSU97]:

- entender o estado dos processos de uma organização para a sua melhoria;
- determinar a adequação dos processos de uma organização para um requisito particular ou uma classe de requisitos;
- determinar a adequação dos processos de uma outra organização para um determinado contrato ou para uma classe de contratos.

Quadro 10 - Modelo de referência do processo do SPICE

CATEGORIAS DE PROCESSO	SIGLA	PROCESSOS
Cliente-fornecedor (Customer-Supplier)	CUS.1	Adquirir software
	CUS.2	Gerenciar necessidades do cliente
	CUS.3	Fornecer software
	CUS.4	Operar o software
	CUS.5	Prover serviço ao cliente

CATEGORIAS DE PROCESSO	SIGLA	PROCESSOS
Engenharia (Engineering)	ENG.1	Desenvolver requisitos e projeto do sistema
	ENG.2	Desenvolver requisitos de software
	ENG.3	Desenvolver projeto do software
	ENG.4	Implementar o projeto do software
	ENG.5	Integrar e testar o software
	ENG.6	Integrar e testar o sistema
	ENG.7	Manter o sistema e o software
Apoio (Support)	SUP.1	Desenvolver documentação
	SUP.2	Desempenhar a gerência de configuração
	SUP.3	Executar a garantia da qualidade
	SUP.4	Executar a verificação dos produtos de trabalho
	SUP.5	Executar a validação dos produtos de trabalho
	SUP.6	Executar revisões conjuntas
	SUP.7	Executar auditorias
	SUP.8	Executar resolução do problema
Gerência (Management)	MAN.1	Gerenciar o projeto
	MAN.2	Gerenciar a qualidade
	MAN.3	Gerenciar riscos
	MAN.4	Gerenciar subcontratantes
Organização (Organization)	ORG.1	Construir o negócio
	ORG.2	Definir o processo
	ORG.3	Melhorar o processo
	ORG.4	Prover recursos treinados
	ORG.5	Prover infra-estrutura organizacional

Fonte: [WEB99,BAR97,TSU97]

Quadro 11 – Níveis de Capacitação

Nível	Estágio	Descrição das atividades por estágio
Nível 0	Processo incompleto	O processo não está implementado. O processo falha na tentativa de atingir os seus objetivos.
Nível 1	Processo executado	O processo implementado atinge o seu objetivo definido.
Nível 2	Processo gerenciado	O processo executado entrega produtos de trabalho de definida qualidade dentro de cronogramas e recursos definidos.
Nível 3	Processo estabelecido	O processo gerenciado é executado usando um processo definido baseado em bons princípios de engenharia de software.
Nível 4	Processo previsível	O processo estabelecido é executado consistentemente dentro de limites definidos de controle para atingir seus objetivos.

Nível	Estágio	Descrição das atividades por estágio
Nível 5	Processo otimizado	O processo previsível otimiza o seu desempenho para atender às necessidades de negócio atuais e futuras, e atinge repetibilidade em atender seus objetivos definidos de negócios

Fonte: [WEB99, BAR97, TSU97]

3.2.7 Metodologia Bootstrap

O BOOTSTRAP trata-se de um modelo de avaliação de processo, criado dentro do projeto europeu ESPRIT para resolver crises de desenvolvimento de software. Tem como princípio reduzir custos e melhorar a qualidade prevendo os problemas, e não reagindo a eles [LEM97].

O modelo define o paradigma Organização-Metodologia-Tecnologia que se utiliza no Bootstrap nos níveis de avaliação e agrupamento de resultados. Seu objetivo é melhorar e refinar o CMM, a fim de adaptá-lo a necessidades específicas da indústria de software europeia. Envolve também conceitos da norma de qualidade ISO 9000 e do modelo de processo ESA (European Space Agency – Agência Espacial Europeia) [RUB96].

Segundo o Instituto Bootstrap, antigo ESPRIT, Bootstrap é um método para analisar, redesenhar e melhorar os processos de negócios de desenvolvimento de software. Este se compõe de: um modelo, um processo de avaliação, uma base de dados de suporte, um processo de melhoria e os instrumentos de avaliação. O Bootstrap trabalha com os mesmos cinco níveis de maturidade propostos no CMM. Contudo, entre outras diferenças, preconiza a classificação de uma equipe ou organização em um determinado nível de maturidade por questão do seu questionário. Assim, uma organização pode, por exemplo, estar em nível 1 na questão 2 e nível 3 na questão 10 [HAA94].

O modelo Bootstrap se baseia em 2 atividades principais: processo de avaliação, processo de melhoria. O processo de avaliação se baseia em avaliar UPSs (Unidades de Produção de Software), através de seus projetos, para fazer uma extrapolação em toda a organização. Dentro deste processo, há quatro etapas principais: preparação, execução

da avaliação, determinação do nível de maturidade e capacidade e a apresentação de resultados da avaliação [RUB96].

Na etapa de preparação se realizam as seguintes tarefas: a) um treinamento inicial para ter objetivos claros; b) se selecionam os projetos a serem avaliados para obter a melhor cobertura da UPS; c) se define o pessoal de avaliação para minimizar a subjetividade da avaliação; d) se define o pessoal a ser avaliado para obter a melhor cobertura dos envolvidos nos projetos selecionados e e) se faz um acordo de confidência.

Na execução, as tarefas são: a) uma breve reunião de abertura, para obter um enfoque colaborativo com o pessoal a ser entrevistado; b) o preenchimento dos questionários com características gerais das UPS; c) o preenchimento dos questionários do projeto elegido, incluindo a avaliação de como o processo de produção é aplicado; d) revisão preliminar da avaliação e e) reunião final, com o enfoque de apresentar os resultados da avaliação e obter o consenso para passar para a fase de melhorias.

Na etapa de determinar o nível de maturidade e capacidades, é onde se qualifica cada pergunta com um dos 5 valores possíveis: nulo, insuficiente, regular, extenso (bom) e não-aplicável. Para cada atributo fixado se obtém um nível de maturidade, aplicando um algoritmo numérico, dando como resultado um destes níveis: 1-inicial, 2-repetível, 3-definido, 4-Gerenciado, 5- Otimizado. Estes níveis de maturidade estão subdivididos em quatro, de forma que se obtenha uma qualidade mais exata. Os processos de organização e metodologia se qualificam de 1 a 5, enquanto que a de tecnologia se qualifica somente com dois níveis A ou B.

Como resultado da avaliação, a organização recebe 2 relatórios, um com os resultados da avaliação da UPS e outro com os resultados do projeto avaliado. O correspondente a UPS contém informações como: um resumo executivo, os objetivos da UPS, os pontos fracos e fortes, um plano de ação recomendado, etc. O relatório do projeto contém: comentários do projeto atual detalhando o referente a organização, metodologia e tecnologia, os níveis de maturidade para o projeto, o plano de ação recomendado, etc.

Há vários gráficos que se apresentam para mostrar os aspectos antes mencionados. Estes gráficos podem representar a UPS, o projeto, a organização ou alguma área em particular de acordo com a arquitetura do Bootstrap.

Na definição do processo de melhoria, se encontra um plano de melhoria, onde o primeiro passo é avaliar as necessidades da organização levando em conta as melhoras desejadas e indicadores sobre qualidade do produto e serviço, tempo de desenvolvimento, custos e riscos do produto e do projeto. Em seguida fazer uma revisão e análise de resultados da avaliação, levando em conta a robustez e debilidades detectadas. Depois definir as capacidades a melhorar, considerando um período entre 18 e 24 meses. Definindo após, as prioridades de acordo com uma análise de impactos. Finalmente com base nas atividades definidas, modificar a organização e responsabilidades para iniciar a mudança, estabelecendo um marco de períodos para seu desenvolvimento e avaliação.

Para realizar estas atividades, o Bootstrap conta com uma base de dados de suporte, para realizar suas análises, fundamentando o plano de melhoria. Outro recurso, são os instrumentos que apóiam o processo de avaliação. Dentre eles estão: questionários, a ferramenta para o registro e apresentação de resultados e guias para os acessores.

Os questionários são usados para dirigir as entrevistas, onde os acessores os preenchem com base em discussões e análise do material documentado. A maioria das perguntas estão baseadas em conclusões como: a existência de um procedimento formal, a existência de metodologia, a existência de padrões, a disponibilidade de tecnologia, recomendações de uso da tecnologia, desempenho de tarefas com base em um procedimento, responsabilidades na tomada de decisões, desempenho de uma análise sistemática de resultados, etc.

O software usado como ferramenta para registrar e apresentar os resultados tem os seguintes componentes: *BootCollector*, onde se registram os resultados dos questionários; *BootRetriever*, é para ler ou guardar(manter) dados da base de dados (*BootBase*); *BootCounter* é para calcular os níveis de maturidade; *BootAnalyzer* é para mostrar e imprimir as visões e níveis de maturidade e o *BootManager* que é para administrar a base de dados.

As guias (normas) são para homogeneizar os critérios de qualificação entre os acessores.

3.2.8 ISO/IEC 9126 ou NBR 13596 (Tecnologia de informação – Métricas e características de qualidade de software)

As normas ISO/IEC 9126 listam um conjunto de características que devem ser verificadas em um software para que ele seja considerado um “software de qualidade” [BAR97]. Esta norma não fornece subcaracterísticas e métricas, nem métodos para medição, pontuação ou julgamento. Uma proposta de definição de subcaracterísticas é fornecida no Anexo A, com fins informativos na norma brasileira NBR 13596, que é equivalente à ISO/IEC 9126 [NBR96].

Esta norma define as seguintes seis características e sugere mais vinte e uma subcaracterísticas da qualidade para produtos de software (Quadro12).

Quadro 12 – Características e Subcaracterísticas descritas na norma NBR 13596

Características	Subcaracterísticas	Pergunta chave para a Subcaracterísticas
Funcionalidade (satisfaz as Necessidades?)	Adequação	Propõe-se a fazer o que é apropriado?
	Acurácia	Faz o que foi proposto de forma correta?
	Interoperabilidade	Interage com os sistemas especificados?
	Conformidade	Está de acordo com as normas, leis, etc.?
	Segurança de acesso	Evita acesso não autorizado aos dados?
Confiabilidade (é imune a falhas?)	Maturidade	Com que frequência apresenta falhas?
	Tolerância a falhas	Ocorrendo falhas, como ele reage?
	Recuperabilidade	É capaz de recuperar dados em caso de falha?
Usabilidade (é fácil de usar?)	Inteligibilidade	É fácil entender o conceito e a aplicação?
	Apreensibilidade	É fácil aprender a usar?
	Operacionalidade	É fácil de operar e controlar?

Características	Subcaracterísticas	Pergunta chave para a Subcaracterísticas
Manutenibilidade (é fácil de modificar?)	Modificabilidade	É fácil modificar e adaptar?
	Estabilidade	Há grande risco quando se faz alterações?
	Testabilidade	É fácil testar quando se faz alterações?
Portabilidade (é fácil de usar em outro ambiente?)	Adaptabilidade	É fácil adaptar a outros ambientes?
	Capac. para ser instalado	É fácil instalar em outros ambientes?
	Conformidade	Está de acordo com padrões de portabilidade?
	Capac. para substituir	É fácil usar para substituir outro?

Fonte: [BAR97, NBR96].

Reuniões para elaboração da nova norma indicam as seguintes adaptações nas subcaracterísticas da qualidade [WEB99]:

- na característica eficiência, substituição da subcaracterística comportamento dos recursos por utilização dos recursos;
- na característica portabilidade, inclusão da subcaracterística coexistência;
- inclusão da subcaracterística conformidade em todas as característica.

Pela maneira com que estas características foram definidas, não é possível a sua medição direta, havendo a necessidade da definição de métricas para correlacionar estas características ao software propriamente dito [WEB99].

3.2.9 ISO/IEC 14598 (Tecnologia de informação – Avaliação de produtos de software)

A série de normas ISO/IEC 14598 encontram-se em fase de desenvolvimento, ela visa avaliar a qualidade dos produtos de software e será um complemento à série de normas ISO/IEC 9126. A ISO/IEC 14598, trata-se de uma série de 6 partes, que é constituída por uma visão geral (parte 1), três processos de avaliação do produto focando os processos para desenvolvedores, compradores e avaliadores (respectivamente as partes 3, 4 e 5) e finalmente dois processos de suporte à avaliação (partes 2 e 6), como é descrito em mais detalhes abaixo [RUB96, WEB99]:

a) ISO/IEC 14598-1: Visão Geral

Essa norma, concluída em 1998, apresenta toda a estrutura de funcionamento da série de normas para avaliação da qualidade dos produtos de software além de definir os termos técnicos utilizados nesse modelo. Fornece também os

conceitos e o funcionamento do processo de avaliação da qualidade de qualquer tipo de software, para utilização por equipes de desenvolvimento (incluindo gerentes, analistas de requisitos, projetistas de software, implementadores e equipe de garantia da qualidade). De maneira geral, pode ser utilizada por pessoas envolvidas no desenvolvimento e no uso de tecnologia de avaliação e padronização.

b) ISO/IEC 14598-2: Planejamento e Gerenciamento

Essa norma deverá conter requisitos e guias para suportar funções de avaliação dos produtos de software. O suporte está relacionado ao planejamento e gerenciamento de um processo de avaliação de software e a sua respectiva tecnologia, incluindo: desenvolvimento, aquisição, padronização, controle, transferência e realimentação do uso de tecnologias de avaliação no âmbito da organização.

c) ISO/IEC 14598-3: Processo para a Equipe de Desenvolvimento

Essa norma destina-se ao uso durante o processo de desenvolvimento e manutenção de software, enfocando a seleção e registro de indicadores que possam ser medidos e avaliados nos produtos intermediários obtidos nas fases do desenvolvimento de sistemas, com o objetivo de prever a qualidade do produto final a ser desenvolvido. A norma deverá fornecer em particular:

- critérios para seleção de indicadores de qualidade;
- guia para avaliar dados de medição;
- guia para melhoria do processo de medição.

d) ISO/IEC 14598-4: Processo para Adquirentes

Essa norma deverá estabelecer um processo sistemático para avaliação de produtos de software tipo pacote, ou pré-desenvolvidos, que podem ser obtidos via aquisição ou reuso, visando a decisão quanto à aceitação de um produto, ou ainda para a seleção entre diversas alternativas de produtos, a partir das características da qualidade e do modelo de avaliação de produto de software descrito na norma ISO/IEC 9126-1.

e) ISO/IEC 14598-5: Processos para Avaliadores

Essa norma, concluída em 1998, fornece orientações para a implementação prática de avaliação de produto de software, quando diversas partes necessitam entender, aceitar e confiar em resultados de avaliação. Pode ser utilizada para aplicar os conceitos da Norma ISO/IEC 9126. O processo descrito define as atividades necessárias para analisar os requisitos de avaliação para especificar, projetar e executar as atividades de avaliação e para se obter a conclusão sobre avaliação de qualquer tipo de produto de software.

f) ISO/IEC 14598-6: Documentação de Módulos de Avaliação

Essa norma deverá definir a estrutura e o conteúdo da documentação a ser usada na descrição dos Módulos de Avaliação. Explica como desenvolver módulos de avaliação e como validá-los. Um módulo de Avaliação é um conjunto de instruções e dados usados para avaliação. Ele especifica os métodos de avaliação aplicáveis para avaliar as características de qualidade. Define também os procedimentos elementares de avaliação e o formato do relatório de apresentação dos resultados das medições resultantes das aplicações das técnicas. O uso de módulos de avaliação produzidos e validados, conforme a norma, deve garantir que as avaliações de software possam ser repetidas, reproduzidas e imparciais.

3.2.10 ISO/IEC 12119 ou NBR ISO/IEC 12119 (Tecnologia de informação – Pacotes de software – Teste e requisitos de qualidade)

Esta norma foi 1994 (ISO/IEC 12119) e traduzida em 1998 pela ABNT [NBR98a]. Ela trata da avaliação de pacotes de software, também conhecidos como “software de prateleira”. Sua abrangência refere-se somente ao pacote de software e se resume em estabelecer requisitos de qualidade para este tipo de software e instruções de como testar um pacote de software em relação aos requisitos estabelecidos.

A norma, de maneira geral, pode ser descrita como mostra a Quadro 13 abaixo:

Quadro 13 – Descrição da Norma NBR ISO/IEC 12119

Item	Descrição
1. Escopo	
2. Definições	
3. Requisitos de qualidade	
3.1 Descrição do produto	Um item que estabelece as propriedades do produto, com o propósito de orientar potenciais compradores na avaliação da adequação do produto às suas necessidades, antes de adquiri-lo. Cada declaração deve ser correta e testável. Deve incluir declarações sobre funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. A indisponibilidade deste item é considerado como uma não-conformidade maior.
3.2 Documentação do usuário	A documentação de usuário deve conter as informações necessárias para o uso do produto. E, estas informações, devem ser completas (abrangendo todas as funções do software), corretas (sem ambigüidades ou erros), consistentes (cada termo com significado único, sem conflitos com outras descrições do produto), inteligíveis (fácil de entender) e capaz de fornecer uma visão geral do produto.
3.3 Programas e dados	O conjunto completo de programas de computador e dados fornecido para a aplicação do produto de software e também como parte integral deste produto. Inclui também declarações a serem verificadas sobre funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.
4. Instruções para teste	
4.1 Pré-requisitos de teste	Lista de itens necessários ao teste, incluindo documentos que deverão fazer parte do pacote, componentes do sistema e material de treinamento (caso seja mencionado na descrição do produto).
4.2 Atividades de teste	Instruções detalhadas sobre os procedimentos de teste, inclusive instalação e execução de cada uma das funções descritas.
4.3 Registro de teste	Informações sobre como os testes foram realizados, de tal forma a permitir uma reprodução destes testes. Deve incluir parâmetros utilizados, resultados associados, falhas ocorridas e até a identidade do pessoal envolvido.
4.4 Relatório de teste	Relatório incluindo: identificação do produto, hardware e software utilizado, documentos utilizados, resultados dos testes, lista de não conformidades com os requisitos, lista de não conformidades com as recomendações, datas, etc.

Item	Descrição
4.5 Teste de	Ao retestar um produto (levando-se em consideração o

acompanhamento	teste anterior), todas as parte modificadas e mesmo as inalteradas que podem ser influenciadas pelas partes alteradas, deverão ser retestadas a fim de comprovar a corretude dos dados disponíveis de teste. Todas as outras partes devem ser retestadas, considerando-se uma amostragem dos casos.
----------------	---

Fonte:[BAR97, WEB99, NBR98a]

Um dos grandes méritos desta norma está na profundidade com que são descritos os itens que deverão estar presentes no pacote de software, o que refletiu num nível de compreensão sem grandes dificuldades. Além disso, existe um anexo (Anexo A) de referências cruzadas com outras normas e o principal, e mais importante, é um exemplo (Anexo B) de uma descrição de um produto, um pacote de software simples e fictício, com o objetivo de mostrar as informações que devem estar presentes em toda a descrição do produto, o que não acontece na maioria das normas, senão todas. Normalmente, se limitam a apresentar características gerais sem exemplos aplicáveis o que facilitaria mais o seu entendimento.

Encontra-se ainda, no Anexo C, um rol de bibliografias de normas, nacionais e internacionais, relacionadas às características e subcaracterísticas descritas nesta norma.

3.2.11 Modelo PSP

O Processo Pessoal de Software (PSP) foi desenvolvido pela SEI, para atender a pequenas empresas e até mesmo para um único indivíduo, já que o modelo CMM se aplica mais a grandes empresas de software. Este processo introduz uma disciplina para práticas de engenheiros de software individual, dramaticamente melhorando a qualidade, previsibilidade e tempo de ciclo para sistema de software intensivo. O PSP cria engenheiros cientes do processo que eles utilizam para fazer seus trabalhos e do desempenho desses processos. Eles aprendem a estabelecer metas pessoais para melhoria, medição e análise de seus trabalhos, e adaptar seus processos para satisfazer suas metas. Deste modo, engenheiros de software desenvolvem a habilidade para prever seu desempenho e gerenciar a qualidade do trabalho por eles produzidos. É uma estratégia de auto-desenvolvimento profissional e aumento da produtividade [BAR97, PSP99].

O PSP é introduzido com um treinamento formal. O treinamento fornece aos engenheiros uma oportunidade de aprender e praticar o PSP antes de aplicar em seus projetos de trabalho. O treinamento do PSP é realizado por meio de 10 exercícios de desenvolvimento de programas. Além de, servirem como exemplos de desenvolvimento, os exercícios propostos pelo treinamento são pequenos utilitários que ajudam o indivíduo que está sendo treinado a aplicar o PSP, pois permite medir o número de linhas e objetos nos seus programas, calcular desvio padrão, prever intervalos, etc [BAR97]. Os resultados obtidos com o PSP, com uma amostragem de 104 engenheiros que obtiveram o treinamento prévio completado mostrou respostas animadoras [PSP99]:

- Precisão da estimativa do tempo e tamanho. Dois terços dos engenheiros foram capazes de estimar o tamanho dos programas de software que eles escreveram na classe, e o esforço para desenvolvê-lo, alcançado em +/- 10% do final do curso;
- Melhoras na qualidade de software. Em média, o total de defeitos injetados durante o desenvolvimento foram reduzidos em 60%, e os defeitos encontrados durante a unidade de teste foram reduzidos em 75%;
- A média da melhora da produtividade atingiu 28%.

Assim como o CMM, no modelo PSP, existem diversos níveis com características próprias. O modelo PSP possui os seguintes níveis que encontram-se exibidos na Quadro 14:

Quadro 14 – Níveis do Modelo PSP

Nível	Nome	Atividades
PSP0 PSP0.1	Medição pessoal	Registro de tempo Registro de defeitos Padrão de tipos de defeitos Padrão de codificação Medida de tamanho Proposta de melhoramento do processo
PSP1 PSP1.1	Planejamento pessoal	Estimativa de tamanho Relatório de testes Planejamento de tarefas Cronogramas
PSP2 PSP2.1	Qualidade pessoal	Revisões de código Revisões de projeto Padrões de projeto

Nível	Nome	Atividades
PSP3	Processo cíclico pessoal	Desenvolvimento cíclico

Fonte: [BAR97, HUM95, PSP99]

Inicia-se o projeto no nível de medição pessoal. Neste primeiro passo, engenheiros aprendem como aplicar o PSP (métodos e formulários PSP) em seus trabalhos pessoais. Eles registram então, o tempo gasto em cada etapa do ciclo do desenvolvimento e os defeitos encontrados por meio de formulários adequados. O nível PSP0.1, inclui o uso de um padrão de codificação, de medidas padronizadas e do formulário de proposta de melhoramento do processo (PIP – Processo Improvement Proposal). Os formulários PIP deixam que engenheiros registrem problemas, questões e idéias para usar durante a melhoria de seus processos.

No PSP1 (nível de Planejamento Pessoal), os engenheiros aprendem a planejar suas tarefas. O intuito é de obter a capacidade de estimar quanto tempo levará para realizar uma tarefa baseado nas medições feitas em tarefas anteriores semelhantes. Neste nível aprende-se a assumir compromissos que podem realmente ser cumpridos. O PSP1.1, inclui o planejamento de tarefas e a elaboração de cronogramas.

No nível de Qualidade Pessoal, os engenheiros aprendem a lidar com seus erros, para isso precisam ter conhecimento da quantidade de erros que são cometidos (em média) em cada fase do ciclo de desenvolvimento. O modelo PSP mostra que a forma mais adequada para tratar erros é evitá-los desde sua origem. Você deve utilizar os dados sobre defeitos já coletados para criar uma lista de verificação (checklist) a ser utilizada em suas revisões de projeto e de código. O nível PSP2.1, inclui a criação de padrões de projeto, bem como métodos de análise e prevenção de defeitos.

O nível de Processo Cíclico Pessoal é a última etapa do PSP. Neste nível o PSP trata do desenvolvimento de projetos maiores; deixando o desenvolvimento de pequenos programas; embora ainda esteja em nível pessoal. Para isto, os grandes projetos são divididos em pequenos projetos que possam ser tratados no PSP2. Neste caso, o desenvolvimento acontece em passos incrementais.

3.2.12 Série ISO 9000

ISO 9000, na realidade, não é um norma, e sim uma série de normas internacionais publicadas em 1987, voltadas para a gestão e garantia da qualidade. Estas normas

especificam os requisitos mínimos para que as empresas possam assegurar a qualidade de seus produtos e serviços, não definindo modelos ou impondo sistemas de qualidade a serem implementados nas organizações. As empresas definem seus próprios modelos de gestão da qualidade dependendo do seu tipo de negócio e suas características [TSU97].

Dentre as normas internacionais da série ISO 9000, três tratam de requisitos de sistema da qualidade [WEB99]:

a) ISO 9001:1994 é um modelo para garantia da qualidade em projeto, desenvolvimento, produção, instalação e serviços associados. No Brasil, a NBR ISO 9001/1994 é equivalente à ISO 9001:1994;

b) ISO 9002:1994 é um modelo para garantia da qualidade em produção, instalação e serviços associados. No Brasil, a NBR ISO 9002/1994 é equivalente à ISO 9002:1994;

c) ISO 9003:1994 é um modelo para garantia da qualidade em inspeção e ensaios finais. No Brasil, a NBR ISO 9003/1994 é equivalente à ISO 9003:1994.

Tendo em vista as características próprias do setor de software, originou-se então a ISO 9000-3 [NBR93] que é um guia (diretrizes) de aplicação da ISO 9001 [NBR94b] para o desenvolvimento, fornecimento e manutenção de software.

Como foi dito no capítulo anterior, a ISO 9001 (pode ser encontrada na NBR ISO 9000-3, como NBR 19001, pois esta foi cancelada e substituída pela NBR ISO 9001 em 1994) é aplicável a empresas em geral que atuam em projeto, desenvolvimento, produção, instalação e assistência técnica. Como esses documentos da Série ISO 9000 são genéricos, foi necessária a elaboração de um documento complementar onde fossem abordados alguns aspectos importantes e específicos de software. Em junho de 1993, foi criada a Norma ISO 9000-3 com diretrizes para aplicação da ISO 9001 ao desenvolvimento, fornecimento e manutenção de software. Esta norma se espelha nos itens da ISO 9001, fazendo a necessária adaptação. No entanto, este documento (ISO 9000-3), no que se refere ao formato, não corresponde diretamente à norma NBR ISO 9001, sendo, por essa razão, fornecidos índices de referência cruzada (anexos A e B) para facilitar a consulta àquela norma.

As diretrizes propostas na ISO 9000-3 cobrem questões como o entendimento comum entre as partes (contratante e contratado) de requisitos funcionais e o uso de

metodologias consistentes para o desenvolvimento de software e gerenciamento de projeto como um todo, da concepção até a manutenção. É dividida em três partes principais [TSU97, NBR93]:

Estrutura: descreve aspectos organizacionais, relacionados ao sistema de qualidade. São detalhadas as responsabilidades e ações relacionadas à qualidade que devem ser tomadas tanto pelo fornecedor como pelo comprador. Os pontos abordados são: responsabilidades da administração; sistemas de qualidade; auditorias internas do sistema de qualidade e ação corretiva.

Atividades do ciclo de vida: descreve as atividades de desenvolvimento de software. A Norma define que o desenvolvimento de software deve ser feito segundo um determinado modelo de ciclo de vida, e as atividades relacionadas à qualidade devem ser planejadas e implementadas de acordo com a natureza deste modelo. Independentemente do modelo de ciclo de vida estabelecido pela organização, a norma define que as atividades do ciclo de vida devem ser agrupadas em nove categorias: análise crítica do contrato; especificação dos requisitos do comprador; planejamento do desenvolvimento; planejamento da qualidade; projeto e implementação; ensaios e validação; aceitação; cópia, entrega e instalação e manutenção.

Atividades de suporte: descreve as atividades que apoiam as atividades do ciclo de vida de desenvolvimento. Estão organizadas em nove itens: gestão de configuração; controle de documentos; registros da qualidade; medição; regras; práticas e convenções; ferramentas e técnicas; aquisição; produto de software incluído e treinamento.

A procura pela certificação da série ISO 9000 é um dos grandes motivadores do atual movimento em relação à qualidade em todas as áreas de atividades econômicas, incluindo o software. A certificação ISO 9000 é reconhecida praticamente em todos os países e por todos os setores, não só pelo setor de software. Para uma empresa, conquistar a certificação ISO 9000 significa alcançar padrão internacional de qualidade em seus processos de software [WEB99].

É importante salientar que a ISO 9001 somente indica o controle da não conformidade de um produto e recomenda ações corretivas e preventivas. A melhoria contínua do processo não é abordada por ela tal como é abordada explicitamente no modelo SW-CMM ou em outros [COA94]. Isto explica o por quê, de algumas empresas utilizarem mais de uma norma ou modelo; a intenção é complementar aspectos que

seriam relevantes para atingir os objetivos específicos de determinadas organizações ou projetos.

3.2.13. ISO IEC 12207 ou NBR ISO/IEC 12207 (Tecnologia de informação – Processos de ciclo de vida de software)

Com a atual preocupação em se desenvolver software cada vez mais confiáveis e eficientes a ISO e a IEC, através do Comitê Técnico Conjunto JTC1, aprovaram a Norma Internacional ISO/IEC 12207:1995 (ou NBR ISO/IEC 12207:1998), que define os processos do ciclo de vida do software. Ela estabelece os processos, atividades e tarefas a serem aplicados durante a aquisição, fornecimento, desenvolvimento, operação e manutenção de software [WEB99, NBR98b, TSU97].

Os processos do ciclo de vida de software são apresentados a seguir [WEB99, NBR98b]:

a) Processos Fundamentais:

- a.1) processo de aquisição, que define as atividades do adquirente, isto é, a organização que adquire um sistema, produto ou serviço de software;
- a.2) processo de fornecimento, que define as atividades do fornecedor, isto é, a organização que fornece um sistema, produto ou serviço de software ao adquirente;
- a.3) processo de desenvolvimento, que define as atividades do desenvolvedor, isto é, a organização que define e desenvolve o produto de software;
- a.4) processo de operação, que define as atividades do operador, isto é, a organização que presta serviço de operação de um computador para seus usuários;
- a.5) processo de manutenção, que define as atividades do responsável pela manutenção, isto é, a organização que presta serviços de manutenção de software.

b) Processos de Apoio:

- b.1) processo de documentação, que define as atividades de registro da informação produzida pelos processos do ciclo de vida;

- b.2) processo de gerência da configuração, que define as atividades de gerenciamento das configurações do software;
- b.3) processo de garantia da qualidade, que define as atividades voltadas para assegurar objetivamente que os produtos de software estão em conformidade com os requisitos especificados e aderentes aos planos estabelecidos;
- b.4) processo de verificação, que define as atividades (para o adquirente, o fornecedor ou uma terceira parte independente) para a verificação dos produtos de um projeto de software;
- b.5) processo de validação, que define as atividades (para o adquirente, o fornecedor ou uma terceira parte independente) para a validação dos produtos de um processo de software;
- b.6) processo de revisão conjunta, que define as atividades para avaliar a situação e os produtos de uma atividade em um dos processos de software;
- b.7) processo de auditoria, que define as atividades para determinação da conformidade com requisitos, planos e contratos;
- b.8) processo de resolução de problemas, que define as atividades para análise e remoção de problemas (inclusive de não-conformidades).

c) Processos Organizacionais

- c.1) processo de gerência, que define as atividades de gerenciamento dos processos de software, incluindo a gerência de projetos;
- c.2) processo de infra-estrutura, que define as atividades para o estabelecimento da estrutura de apoio de um processo;
- c.3) processo de melhoria, que define as atividades que uma organização (isto é, o adquirente, o fornecedor, o desenvolvedor, o operador, o responsável pela manutenção ou o gerente de um outro processo) realiza para estabelecer, medir, controlar e melhorar continuamente os seus processos de software;
- c.4) processo de treinamento, que define as atividades para prover o pessoal adequadamente treinado;
- c.5) processo de adaptação, que define as atividades para adaptar a norma para uma organização ou um projeto de software.

Os processos do ciclo de vida de software podem ser usados de diferentes maneiras por diferentes organizações, permitindo assim diversas visões. Cada uma destas visões representam a forma que uma organização realiza estes processos, agrupando-os de acordo com suas necessidades e objetivos. As visões (que podem ser de contrato, de gerência, de operação, de engenharia e de apoio), tem como objetivo organizar melhor a estrutura de uma empresa, ajudando a definir suas gerências e atividades destinadas aos grupos de trabalho [MAC97].

No Brasil, a primeira medição do conhecimento e uso desta norma realizada pela SEPIN/MCT (Secretaria de Política de Informática e Automação do Ministério da Ciência e Tecnologia) junto com o PBQP/SSQP-SW (Subcomitê Setorial da Qualidade e Produtividade em Software, do Programa Brasileiro de Qualidade e Produtividade), revelou dados interessantes, apontando 25% de empresas que a conhecem e mais de 5% que já fazem uso efetivo dela. Esses dados foram coletados em 1997 em 588 empresas participantes da pesquisa (Tabela 3) “Qualidade no Setor de Software Brasileiro” (mais informações podem ser adquiridas em).

Tabela 3 – Conhecimento da Norma ISO/IEC 12207, em 1997

Categorias	Nº	%
Conhece e usa	32	5,4
Conhece, mas não usa	115	19,6
Não conhece	441	75,0
Base	588	100

Fonte: [WEB99]

4 SOFTCHECK: UMA LISTA DE CHECAGEM

4.1 INTRODUÇÃO

Mudanças concebidas nos últimos 25 anos na área de informática, tentam convergir para um software de qualidade e, apesar das grandes facilidades encontradas no mercado mundial, os softwares desenvolvidos deixam a desejar. Deste ponto de vista, duas hipóteses podem ser levantadas: a inexistência de metodologias adequadas ou falta de cultura de software.

Como uma mudança cultural é um elemento que deve ser perseguido, para que, com o passar dos anos verifique-se alguma mudança, o esforço deverá concentrar-se em criar novas ou melhorar metodologias existentes. O ideal é que haja um duplo esforço: continuar com a divulgação dos métodos existentes entre os profissionais e futuros profissionais e dar continuidade na evolução metodológica.

O desenvolvimento de software atualmente conta com várias técnicas, modelos, métricas e padrões. Este último recurso - padronização, que quando da sua criação possuía a finalidade de representar apenas um meio de comunicação entre os seres humanos nos dias atuais mostra-se um meio de uniformização universal, fazendo com que um software desenvolvido em um determinado local possa ser utilizado em qualquer outra parte do mundo e em qualquer máquina, caso não haja restrições predefinidas.

No entanto, a padronização não acontece em muitos softwares e outros produtos oferecidos no mercado. Atualmente, a quantidade de empresas que se utiliza de normas técnicas, assim referidos os processos de padronização por órgãos nacionais e internacionais, são mínimas. O motivo se refere à disposição, ou melhor, apresentação e organização de tais normas que se encontram num formato de contrato, gerando dificuldades no seu entendimento e manipulação, devido à forma subjetiva na sua descrição. Além disso, muitas delas são generalistas, levando o responsável pelo seu uso a interpretá-las e executá-las de acordo com suas experiências em tais atividades, implicando em formas diferentes para se realizar o mesmo processo por pessoas diferentes, rompendo o objetivo original das normas que é a uniformização.

Neste capítulo, apresenta-se uma lista de checagem como meio de minimizar a subjetividade e a generalização, aumentando a utilização de tais técnicas.

Nas sessões 4.2 e 4.3 encontra-se o processo introdutório da lista de checagem, que propõe uma estrutura geral e específica, além de uma notação básica que auxilia o seu desenvolvimento. A sessão seguinte, 4.4, Módulo1, apresenta-se a lista de checagem desenvolvida com base na Norma NBR ISO/IEC 12119. E, finalizando o capítulo, na sessão 4.5, encontra-se o módulo 2 da lista de checagem referente aos utilitários, contendo um glossário de termos técnicos.

4.2 SOFTCHECK

O âmbito deste trabalho gera uma lista de checagem denominada SOFTCHECK (*Software's Checklist* – Lista de checagem ou verificação de *software*). Uma lista de checagem, pode ser definida como um conjunto de questões propostas com o objetivo de verificar se um determinado fim é alcançado ou não, baseando-se ou não em um documento já estabelecido.

O SOFTCHECK é uma lista de questões que propõe a verificação quanto à conformidade existente entre os pacotes de software produzidos e os requisitos contidos na Norma NBR ISO/IEC 12119. Além disso, como contribuição à questão mudança cultural, fornece também ao usuário um rol de recomendações para cada questão analisada e assinalada como não empregada, a fim de mudar a opinião do usuário quanto aos benefícios que trarão o seu emprego.

A estrutura do SOFTCHECK está organizada da seguinte forma:

1. Informações ao usuário do módulo ou primeiro submódulo, definindo as funções/características do mesmo;
2. Questões que abrangem:
 - a) A pergunta em si, ou seja, uma ou mais frases com que se interroga [SOA97], baseada nas respectivas fontes, com numeração identificadora;
 - b) Possíveis respostas pré-determinadas do tipo: sim, não e não aplicável (explicadas posteriormente na sessão de notação). As perguntas são de resposta única.
 - c) Recomendações com argumentação para respostas diferentes de “sim”.
 - d) Em algumas perguntas, caso necessário, haverá um “sinal” de corte e Continuidade, para respostas que não sejam afirmativas, significando que

as questões entre a respondida - origem do corte, até o número da questão equivalente ao sinal de continuidade, são parte do contexto onde a resposta foi marcada como uma afirmação falsa. Assim sendo, não há necessidade de que o usuário passe por todas elas.

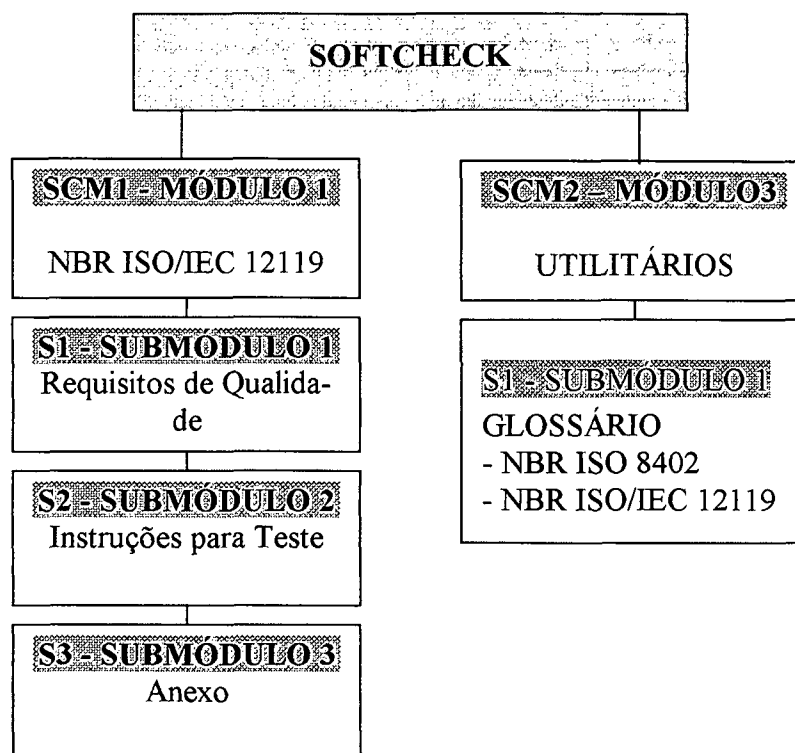
3. Um glossário abrangendo termos técnicos de gestão de qualidade e garantia da qualidade, definidos na Norma NBR ISO 8402, além de outros, especificados na Norma NBR ISO/IEC 12119.

4.2.1 Estrutura Geral do SOFTCHECK

Nesta sessão apresenta-se a estrutura geral da lista de checagem SOFTCHECK. A lista foi organizada de acordo com as Normas originais traduzidas para o português pela ABNT – Associação Brasileira de Normas Técnicas. O MÓDULO1 (SCM1), diz respeito aos processos finais do desenvolvimento de pacotes de software, com características de requisitos de qualidade e testes. A base deste módulo do SOFTCHECK é a Norma de nº 12119, que foi traduzida para o português e validada em 1998. Seu objetivo principal é apresentar condições ao usuário final de adquirir um pacote de software que atinja suas metas no sistema de computação que ele deseja construir ou alterar, isto é, que o produto tenha suporte para o comprador, que o impeça de cometer um equívoco adquirindo um produto que não lhe trará benefícios ou que ele não possa utilizar. Este módulo se divide nos submódulos: requisitos de qualidade, instruções de teste e finaliza com um exemplo fictício de como uma descrição de produto deveria ser para atender às expectativas dos adquirentes.

E, por último, apresenta-se o MÓDULO2 (SCM2), denominado de Utilitários. O primeiro submódulo envolve um glossário de termos utilizados nos módulos anteriores e também os referenciados, como a Norma NBR ISO 8402, que apresenta terminologias de qualidade gerais.

Figura 3– Estrutura Geral do SOFTCHECK



4.2.2 Estrutura Específica dos Módulos

A organização específica dos módulos e submódulos, contida na lista de checagem foi desenvolvida de acordo com as Normas NBR ISO/IEC 12119 e a NBR ISO 8402.

Quadro 15 – Pacotes de Software – Testes e Requisitos de Qualidade

MÓDULO 1 (SCM1)	
Pacotes de Software – Teste e Requisitos de Qualidade	
Submódulo 1 – Requisitos de Qualidade	Submódulo 1.1 – Descrição do produto Submódulo 1.2 – Documentação do usuário Submódulo 1.3 – Programas e dados
Submódulo 2 – Instruções de Teste	Submódulo 2.1 – Pré-requisitos de teste Submódulo 2.2 – Atividades de teste Submódulo 2.3 – Registros de teste Submódulo 2.4 – Relatório de teste Submódulo 2.5 – Teste de acompanhamento
Submódulo 3 – Anexo	Submódulo 3.1 – Exemplo de uma descrição de um produto

Quadro 16 – Utilitários do SOFTCHECK

MÓDULO2 (SCM2) Utilitários
<p style="text-align: center;">Submódulo1 – Glossário</p> <p style="text-align: center;">Submódulo1.1 – Termos e Definições (NBR ISO 8402 – Terminologia de Qualidade e NBR ISO/IEC 12119 – Pacote de software – Teste e requisitos de qualidade).</p>

4.3 NOTAÇÃO DO SOFTCHECK

Foi criada para o SOFTCHECK uma notação própria a fim de reduzir as palavras que seriam redundantes tais como: módulo, submódulo, continuação em tal questão, recomendação para questões não empregadas, recomendações para questões não aplicáveis, entre outras. A notação então, substitui por meio de símbolos, palavras que serão copiosamente utilizadas na lista de checagem.

Quadro 17 – Notação da lista de checagem SOFTCHECK.

Notação	Significado	Exemplo/Descrição
SCM	<u>SoftCheck</u> <u>M</u> ódulo	Exemplo: SCM1 – <u>SoftCheck</u> <u>M</u> ódulo <u>1</u>
S1	<u>S</u> ubmódulo	Exemplo: S1 – <u>S</u> ubmódulo <u>1</u>
S1.1	<u>S</u> ubmódulo	Exemplo: S1.1 – <u>S</u> ubmódulo <u>1.1</u> (vários níveis)
Sim	Palavra “ <u>sim</u> ”	Resposta à questões em conformidade com a pergunta que está sendo respondida
Não	Palavra “ <u>não</u> ”	Respostas às questões onde as atividades e tarefas não são empregadas
Não aplicável	Expressão “ <u>não aplicável</u> ”	Respostas às questões não aplicáveis, isto é, onde seu emprego é impossível devido à estrutura, ao tipo de aplicação, etc.
C	<u>C</u> orte	Conduz o usuário do SOFTCHECK às próximas questões saindo do contexto onde as respostas são “não” ou “não aplicáveis”.
:	⋮	Logo após a notação de “corte”, encontra-se o sinal de dois pontos (:), indicando que há uma continuidade para a resposta efetuada. Este símbolo também é utilizado no começo das questões referenciando a que submódulo elas pertencem.

Notação	Significado	Exemplo/Descrição
CNQ	<u>C</u> ontinuidade do <u>N</u> ão, <u>M</u> ódulo <u>X</u> , <u>S</u> ubmódulo <u>Y</u> na <u>Q</u> uestão <u>Z</u>	Esta notação é complemento da notação de “corte” estabelecida, caso sua resposta à pergunta seja negativa. Exemplo: C2:CNM1 (S1.2 -Q4), significando que o “corte” encontra-se na questão <u>2</u> e o restante, refere-se à “ <u>C</u> ontinuidade do <u>N</u> ão no <u>M</u> ódulo <u>1</u> , no <u>S</u> ubmódulo <u>1.2</u> na <u>Q</u> uestão <u>4</u> ”, mostra a localização da questão de continuidade para o usuário.
CNAQ	<u>C</u> ontinuidade do <u>N</u> ão <u>A</u> plicável na <u>Q</u> uestão <u>Z</u>	Esta notação é complemento da notação estabelecida para o “corte”, caso sua resposta à pergunta seja “não aplicável”. Exemplo: C2:CNAQ4, significando que o “ <u>C</u> orte” encontra-se na questão <u>2</u> e o restante “ <u>C</u> ontinuidade do <u>N</u> ão <u>A</u> plicável na <u>Q</u> uestão <u>4</u> ”, mostra a localização da questão de continuidade para o usuário.
RNQ	<u>R</u> ecomendação do <u>N</u> ão da <u>Q</u> uestão <u>Z</u>	Recomendações para questões que o usuário não emprega ao desenvolver, adquirir, operar... um software. Exemplo: RNQ3, <u>R</u> ecomendação do <u>N</u> ão da <u>Q</u> uestão <u>3</u> , apresenta argumentos para que o usuário comece a utilizar a tarefa, atividade, etc.
RNAQ	<u>R</u> ecomendação do <u>N</u> ão <u>A</u> plicável da <u>Q</u> uestão <u>Z</u>	Recomendações para questões que o usuário não aplica a um software, mas poderia e principalmente deveria aplicar. Exemplo: RNAQ3, <u>R</u> ecomendação do <u>N</u> ão <u>A</u> plicável da <u>Q</u> uestão <u>3</u> , apresenta argumentos para que o usuário comece a utilizar a tarefa, atividade, etc.
&	<u>&</u>	Elo de ligação, utilizado tanto para igualdade de recomendações, quanto para igualdade de continuidades, quando não importa a resposta, isto é, se é “não” ou “não aplicável”. Exemplo1: C1:CN&NAQ3, “ <u>C</u> orte” encontra-se na questão <u>1</u> e o restante “ <u>C</u> ontinuidade do <u>N</u> ão e (&) <u>N</u> ão <u>A</u> plicável na <u>Q</u> uestão <u>3</u> ”. Esta notação é utilizada quando independente da resposta (não ou não aplicável, a localização de continuidade é a mesma). O mesmo acontece com as recomendações, quando não importa a resposta, sendo diferente de “sim”. Por exemplo: RN&NAQ2: <u>R</u> ecomendação do <u>N</u> ão e (&) <u>N</u> ão <u>A</u> plicável da <u>Q</u> uestão <u>2</u> .

O termo definido como SCM1, possui como significado que seu módulo de origem dentro da lista é o primeiro módulo (SOFTCHECK Módulo 1) e assim por diante. Para os submódulos a notação utiliza vários níveis incluindo o inicial que se refere à identificação do Submódulo em questão. Por exemplo, S1 (Submódulo 1 e seus níveis: S1.1, S1.2, etc). Outra notação utilizada diz respeito as respostas para cada questão envolvendo: “sim” – para questões em conformidade com a pergunta que está sendo respondida, “não” – para atividades e tarefas que não são empregadas nos processos que está sendo analisado ou construído e finalmente “não aplicáveis” – correspondendo a questões que seu emprego é impossível devido à estrutura, ao tipo de aplicação, etc. E, finalmente, tem-se a notação de “corte” que foi definida para conduzir o usuário às próximas questões saindo do contexto onde as respostas são: “não aplicável” ou “não”. A utilização desta terminologia é resumida em “cortar ou subtrair” uma ou mais questões pertinentes à não empregabilidade da principal. Por Exemplo, C1:CNM1(S2.2:Q3) (Corte 1 e Continuidade do Não Módulo 1, no Submódulo 2.2 na Questão 3), significa que o corte encontra-se na questão 1, indicando em que posição deu-se o início do corte e o CNQ3 (Continuidade do Não na Questão 3) mostra a localização da questão de continuidade para o usuário, caso sua resposta seja negativa (não). CNAQ3, para questões “não aplicáveis”...

Uma outra notação existente neste modelo está relacionada às recomendações dimensionadas para que o usuário faça uso de determinado submódulo, atividade ou tarefa. Por exemplo RNQ2.2 – significando Recomendação do Não da Questão 2.2, pergunta de origem da recomendação. Após a notação encontra-se um ou mais argumento(s) para que o usuário empregue a atividade ou tarefa. Este argumento apresenta em alguns casos as conseqüências que o não emprego de determinado item acarretará no processo. E, RNAQ2.2, significando, Recomendação Não Aplicável da Questão 2.2, possui as mesmas características acima citadas diferindo apenas na resposta (não aplicável) dada pelo usuário à pergunta.

Em casos que, independente da resposta -negativa ou não aplicável, a continuidade é a mesma, ela será simbolizada com notação de origem de corte, continuidade para respostas negativas, um símbolo de ligação (&) e a continuidade para questões “não aplicáveis”. Por último, a localização da questão de continuidade. Por exemplo,

C1:CN&NAQ3, significando: origem do Corte na questão 1, Continuidade do Não e Não Aplicável na Questão 3.

Além da notação, utilizou-se também uma numeração própria para as questões que fazem parte da lista de checagem SOFTCHECK. A numeração foi organizada de maneira que cada submódulo inicia-se no número 1 (um), modificando o módulo do topo da estrutura. Isto se deve ao fato de que cada módulo é independente e os submódulos são dependentes dos módulos iniciais.

4.4 SCM1 (MÓDULO1) – PACOTES DE SOFTWARE – TESTES E REQUISITOS DE QUALIDADE

A aplicabilidade da Norma 12119, utilizada no desenvolvimento deste Módulo, abrange questões referente a pacotes de software. São exemplos: processadores de texto, planilhas eletrônicas, bancos de dados, softwares gráficos, programas para funções técnicas ou científicas e programas utilitários.

De acordo com o documento NBR ISO/IEC 12119, esta norma estabelece:

- os requisitos para pacotes de software (requisitos de qualidade);
- instruções de como testar um pacote de software com relação aos requisitos estabelecidos (instruções para testes, em particular para teste por terceira parte).

Ressalta-se ainda no documento, que a Norma em questão não trata de processos de produção de software (tampouco atividades e produtos intermediários, por exemplo: especificações); trata somente de pacotes de software na forma como são oferecidos e liberados para uso. O sistema de qualidade do produtor está fora do escopo deste módulo.

S1 – Requisitos de Qualidade

Este primeiro submódulo do SCM1, de acordo com a Norma NBR ISO/IEC 12119, apresenta um documento de descrição do produto, expondo as propriedades de um pacote de software, com o principal objetivo de auxiliar os potenciais compradores na avaliação da adequação do produto antes de sua aquisição.

Esta sessão então, referencia requisitos de qualidade dos pacotes de produtos de software, contendo:

- necessidade de que cada pacote de software tenha uma descrição de produto e documentação de usuário;
- requisitos para a descrição de produto. Em particular, há um requisito que esta descrição deve conter informações específicas e que todas as suas declarações devem ser passíveis de testes e corretas;
- requisitos para a documentação de usuário;
- requisitos para os programas e dados, caso existam, incluídos no pacote.

S1.1 – Descrição do Produto

A descrição do produto é um documento que expõe as propriedades de um pacote de software, com o principal objetivo de auxiliar os potenciais compradores na avaliação da adequação do produto antes de sua aquisição.

S1.1:Q1 Cada pacote de software possui uma descrição de produto?
 sim não não aplicável

RNQ1 - Dentre outros, os principais objetivos da descrição do produto envolvem a ajuda ao usuário ou ao comprador em potencial, quando na avaliação da adequação dos produtos às suas necessidades. Por extensão, ela também fornece informações para a venda, além de servir como base para testes. Sua não empregabilidade acarreta em um desconforto ao usuário, que poderá não adquirir o produto por falta de instruções de uso, de instalação e de adequação à máquina que possui.

C1:CNM1(S1.2 – Q1)

S1.1:Q1.1 Faz parte da descrição, a definição do produto?
 sim não não aplicável

RNQ1.1 - A definição do produto facilita/beneficia a venda do mesmo, em principal, por terceiros, onde o desenvolvedor não está presente.

S1.1:Q1.2 A descrição fornece informações sobre a documentação do usuário, programas e, se existirem, sobre os dados?
 sim não não aplicável

RNQ1.6 - A empregabilidade deste requisito poderá desestimular o usuário pelo não entendimento de vários termos com múltiplos significados no mesmo texto. O mais correto seria definir um glossário de termos utilizado em todo o documento, caso seja necessário.

S1.1:Q1.7 Cada declaração da descrição de produto é correta e passível de teste?

sim não não aplicável

RNQ1.7 - Caso a descrição pertinente ao produto não esteja correta, houve um problema quanto à atualização da descrição do produto. Se as declarações não podem ser testadas, é provável que os programas-fonte, por exemplo, não estejam incluídos no pacote de software.

S1.1:Q2 Há uma única identificação para a descrição do produto, como por exemplo, “Descrição Funcional”, “Informação de Produto” ou “Folha de Produto”.

sim não não aplicável

RNQ2 - Recomenda-se uma única identificação para a descrição do produto, pois, mais de uma identificação poderá confundir o usuário, acarretando em um desvio do usuário no momento da leitura, que imagina ter chegado ao final do item que lhe interessa; ou ainda, não conseguirá identificar do que se trata aquele termo, caso este seja referenciado no meio do contexto atual.

S1.1:Q3 A descrição do produto identifica o produto, contendo no mínimo o nome do produto, versão ou data?

sim não não aplicável

RNQ3 - Seu emprego ajuda o usuário a identificar pelo nome e versão/data, por exemplo, se há uma versão atual para que ele possa adquirir.

S1.1:Q3.1 Caso haja duas ou mais variantes mencionadas na descrição do produto, há para cada variante mencionada na descrição do produto no mínimo o nome do produto, o nome da variante e uma versão ou data?

sim não não aplicável

RNQ3.1 - O não emprego deste item irá confundir o usuário, que não conseguirá compreender de qual produto ou variante as instruções que estão descritas trata.

S1.1:Q4 A descrição do produto contém o nome e o endereço impresso ou carimbado, de no mínimo um fornecedor?

sim não não aplicável

RNQ4 - O contato com o fornecedor é importante, em casos de necessidade de suporte técnico, aquisições de novas versões e aplicações.

S1.1:Q5 A descrição do produto faz referência às tarefas que podem ser executadas com a utilização do produto?

sim não não aplicável

RNQ5 - Este é um dos itens mais importantes, pois envolve a divulgação do que o produto é capaz de fazer pelo adquirente.

S1.1:Q6 A descrição de produto faz referência aos documentos de requisitos com os quais o produto está em conformidade?

sim não não aplicável

RNQ6 - Os requisitos abrangem as características elementares do produto, isto é, por meio deles tem-se todos os detalhes minuciosos pertinentes a origem do produto, a que se destina, entre outros. Para tanto, caso seja um item ativo na aquisição, será de grande relevância se, por exemplo, a intenção do comprador for de integrá-lo/adaptá-lo a outro sistema/programa existente em sua empresa.

S1.1:Q7 São especificados os requisitos de hardware e software necessários para colocar o produto em uso; por exemplo, unidade de processamento incluindo coprocessadores, tamanho de memória, tipos e tamanhos dos periféricos de armazenamento, placas de expansão, equipamentos de entrada e saída, ambientes de rede softwares de sistema, outros softwares e inclusive o nome do fabricante?

sim não não aplicável

RNQ7 - Esta especificação é o mínimo que deveria conter em todos os produtos de software. Sem ela, o adquirente poderá fazer uma compra que não conseguirá se

beneficiar dela, já que seus “dispositivos” não são compatíveis com a configuração que o produto exige.

C7:CNM1(S1.1 – Q8)

S1.1:Q7.1 A declaração “ou qualquer outro..., se compatível” é utilizada somente quando já se tem identificado previamente um produto específico de software ou hardware?

sim não não aplicável

RNQ7.1 - Esta declaração é muito vaga para não ter uma referência anterior, além de pedir complemento quando da sua utilização.

S1.1:Q7.2 A declaração “ou uma versão atualizada, se compatível” é utilizada somente quando já se encontra identificado previamente uma versão de produto?

sim não não aplicável

RNQ7.2 - Se uma versão anterior não for mencionada, a declaração acima não tem fundamento, pois, necessita de uma identificação de origem.

S1.1:Q7.3 A declaração “da versão X à versão Y”, pode aparecer normalmente sem ressalvas. Há este tipo de declaração na descrição do produto?

sim não não aplicável

RNQ7.3 - Este é um tipo de declaração, que poderá ser utilizado sem problemas.

S1.1:Q7.4 A declaração “a partir da versão X”, mostra a imprevisão futura, já que pode acontecer o desenvolvimento de outra versão. Há este tipo de declaração na descrição do produto?

sim não não aplicável

RNQ7.4 - A afirmação acima, pode tornar-se falsa pelo aparecimento de uma versão “x+3”, com a qual o pacote de software falharia ao ser executado. Então, seu não emprego é a opção correta.

S1.1:Q8 As interfaces com outros produtos são identificados na descrição do produto que está sendo documentado?

sim não não aplicável

RNQ8 - A descrição de interfaces com outros produtos é um item importante quando, por exemplo, da manutenção do mesmo, já que características não pertencentes ao produto principal, estão sendo compartilhadas de outro produto.

S1.1:Q9 Dentre os itens a serem entregues, todo componente físico, juntamente com todos os documentos impressos e todos os meios de armazenamento estão identificados?

sim não não aplicável

RNQ9 - A identificação é necessária por determinar o reconhecimento com precisão da natureza, tipo ou categoria de cada parte integrante do produto.

S1.1:Q10 Na descrição do produto é declarado se o próprio usuário pode ou não conduzir a instalação do produto?

sim não não aplicável

RNQ10 - A não declaração da possibilidade ou impossibilidade de condução do processo de instalação pelo usuário final, acarretará em dificuldades durante o processo e possíveis reduções de desempenho do produto.

S1.1:Q11 É declarado na descrição se o suporte para a operação do produto é oferecido ou não?

sim não não aplicável

RNQ11 - A ausência de declaração da existência do suporte para operação do produto, inviabiliza o acesso do usuário em caso de problemas com a operação do mesmo.

S1.1:Q12 São oferecidos serviços de manutenção para o produto?

sim não não aplicável

RNQ12 - Serviços de manutenção são recomendáveis, pois, podem agir como fonte de correção para produtos que foram colocados no mercado e que apresentaram problemas, além de outras aplicações.

C1.12:CNM1(S1.1 – Q13)

S1.1:Q12.1 O oferecimento ou não de serviços de manutenção são mencionados na descrição do produto?

sim não não aplicável

RNQ12.1 - Se a empresa oferece os serviços, e eles não estão declarados na descrição do produto para o comprador, o cliente poderá considerá-los inexistentes, podendo optar, pela aquisição de outros produtos, onde exista menção de tais serviços.

S1.1:Q12.2 É declarado especificamente, na descrição do produto, o que pode ser incluído nos serviços de manutenção?

sim não não aplicável

RNQ12.2 - É indispensável a especificação dos itens de manutenção para estipular a abrangência desta prestação de serviço.

S1.1:Q13 A descrição do produto fornece para o usuário uma visão geral das funções disponíveis e as facilidades oferecidas?

sim não não aplicável

RNQ13 - As declarações sobre funcionalidade do produto são imprescindíveis para denotar a superioridade do produto com relação aos concorrentes.

C13:CNM1(S1.1 – Q14)

S1.1:Q13.1 É declarada de forma nítida para cada função mencionada, se ela é parte: do produto, de uma extensão do produto que está integralmente apresentada ou apenas referida na descrição do produto ou é parte simplesmente de uma suplemento sem garantia?

sim não não aplicável

RNQ13.1 - Apesar de que nem toda função precisa ser mencionada e, nem todos os detalhes de como uma função é chamada necessitam ser descritos, as mencionadas

precisam ser devidamente esclarecidas. Caso alguma das funções mencionadas não faça parte do produto adquirido e sua situação não foi declarada, o desconhecimento do cliente poderá acarretar conflitos com o fornecedor.

S1.1:Q14 São fornecidos os valores-limite que são utilizados no produto, como por exemplo, valores máximos ou mínimos, comprimento de chaves, número máximo de registros em um arquivo, número máximo de critérios de busca, tamanho mínimo de amostra, entre outros?

sim não não aplicável

RNQ14 - Haverá uma má utilização do produto por parte do usuário sem os valores-limite especificados. Além disso, talvez o produto não se encaixe no perfil que o comprador precisa. Por exemplo, caso o produto trabalhe apenas com um tipo de moeda e a empresa possui clientes de várias nacionalidades.

C14:CNM1(S1.1 – Q15)

S1.1:Q14.1 No caso de não ser possível fornecer valores-limite fixos (por exemplo, quando eles dependem do tipo de aplicação ou do tipo de dado de entrada), as limitações são estabelecidas?

sim não não aplicável

RNQ14.1 - A falta de estabelecimento de limites poderá suscitar inconsistências de conseqüências não previsíveis.

S1.1:Q14.2 São fornecidas também as combinações de valores que podem ser feitas?

sim não não aplicável

RNQ14.2 - As combinações possíveis de valores devem ser fornecidas para que o usuário tenha conhecimento da real capacidade de processamento do sistema.

S1.1:Q15 A descrição inclui informações a respeito de maneiras para evitar o acesso não autorizado (acidental ou intencional) a programas e dados, caso este serviço seja fornecido?

sim não não aplicável

RNQ15 - Se este serviço não for mencionado na descrição do produto o comprador poderá não adquiri-lo por precisar de um produto que possua sistema de segurança.

S1.1:Q16 Na descrição do produto incluem-se informações sobre os procedimentos para a preservação dos dados (confiabilidade)?

sim não não aplicável

RNQ16 - Mencionar tais procedimentos na descrição do produto, possibilita ao usuário o conhecimento sobre a meios para manutenção de arquivos importantes.

C16:CNM1(S1.1 – Q17)

S1.1:Q16.1 Propriedades adicionais do produto, por exemplo, verificar se a entrada é aceitável, proteção contra conseqüências danosas decorrentes de erro de usuário, recuperação de erro entre outras, são descritas para assegurar a capacidade funcional do produto?

sim não não aplicável

RNQ16.1 - As descrições de tais propriedades, ajudam a identificar o quão confiável o produto é diante de intervenções do usuário.

S1.1:Q17 A interface com o usuário, por exemplo, linha de comando, menu, janelas, teclas de função e função de auxílio, é especificada na descrição do produto?

sim não não aplicável

RNQ17 - A importância da especificação detalhada da interface com o usuário, é devida, normalmente, ao fornecimento de um atrativo visual e alto nível de acessibilidade.

S1.1:Q17.1 O conhecimento específico requerido para a aplicação do produto, por exemplo, conhecimento de uma área técnica, de um sistema operacional, de outro idioma diferente daquele em que foi escrita a descrição de produto ou que possa ser adquirido via treinamento especial, é declarado na descrição do produto?

sim não não aplicável

RNQ17.1 - Se houvesse, na descrição do produto, menção sobre o conhecimento específico necessário, aquisições por pessoas não qualificadas poderiam ter sido evitadas.

S1.1:Q17.2 São declarados todos os idiomas utilizados na documentação de usuário e na interface com o usuário (incluindo mensagens de erro e dados visíveis), tanto para o pacote de software como os de todos os outros produtos mencionados na descrição de produto?

sim não não aplicável

RNQ17.2 - O usuário pode não estar apto para utilizar o produto, se em plena utilização surgir uma mensagem de inconsistência em inglês ou outra língua. Para tanto, verifica-se a necessidade de conter na descrição do produto os idiomas que serão requeridos do usuário.

S1.1:Q18 Caso o sistema possa ser adaptado pelo usuário (por exemplo: mudar parâmetros, mudar algoritmos e atribuir teclas de função), há informações na descrição do produto sobre as ferramentas para esta adaptação e as condições para o seu uso?

sim não não aplicável

RNQ18 - Esta característica do sistema de se adaptar às necessidades do usuário em determinados pontos, é favorável, mostrando a liberdade de opções existentes. No entanto, o usuário deve ser informado de seus limites, evitando problemas por corromper programas e dados.

S1.1:Q19 Se houver proteção técnica contra infrações a direitos autorais que possa dificultar a usabilidade, tais como: proteção técnica contra cópia, datas programadas de expiração de uso, lembretes interativos para pagamento de cópia, tal proteção é declarada?

sim não não aplicável

RNQ19 - Não = A não declaração de tais funções pode ocasionar problemas ao usuário que, por falta de informações pode ter seu sistema bloqueado.

S1.1:Q20 A descrição inclui dados sobre a eficiência de uso e satisfação de usuário?

sim não não aplicável

RNQ20 - Dados sobre a eficiência de uso e satisfação de usuário podem ser colhidas pelos próprio usuários do produto que já o adquiriram. A opinião de outro usuário é muito importante sob o ponto de vista de confiabilidade no pacote de software.

S1.1:Q21 A descrição de produto inclui dados sobre o comportamento do produto em relação ao tempo, tais como tempo de resposta e taxas de *throughput* para uma dada função sob condições estabelecidas (por exemplo, a configuração do sistema)?

sim não não aplicável

RNQ21 – Dados sobre o comportamento do produto são importantes quando, por exemplo, o adquirente necessita de uma tempo de resposta rápido e o programa possui esta característica, dentre outros comportamentos.

S1.1:Q22 A descrição do produto possui declarações sobre a manutenibilidade, isto é, o conjunto de atributos que evidenciam o esforço necessário para fazer modificações no software?

sim não não aplicável

RNQ22 - Não é uma característica obrigatória, no entanto, quanto menor o nível de dificuldades no momento da manutenção, devido a previsão feita durante a definição dos requisitos, maior as facilidades do adquirente.

S1.1:Q23 A descrição do produto contém declarações de portabilidade, isto é, o conjunto de atributos que evidenciam a capacidade do software de ser transferido de um ambiente para outro?

sim não não aplicável

RNQ23 - A portabilidade é uma característica imprescindível nos dias atuais pela renovação tecnológica voraz que a informática tem revelado.

S1.2 – Documentação do Usuário

A documentação do usuário é um conjunto completo de documentos, disponível na forma impressa ou não, que é fornecido para a utilização de um produto, sendo também uma parte integrante do produto. Este submódulo abrange características da documentação como a: completude, correção, consistência, inteligibilidade e a apresentação e organização.

S1.2:Q1 Existe uma documentação do usuário para o pacote de software que está sendo adquirido?

sim não não aplicável

RNQ1 - A documentação do usuário é sua fonte de informações para qualquer problema de pouca gravidade, isto é, que ele mesmo possa resolver; evitando assim, consultas desnecessárias à assistência técnica.

C1:CNM1(S1.3 – Q1)

S1.2:Q1.1 Na documentação do usuário todas as funções estabelecidas na descrição de usuário e todas as funções de programa que os usuários tenham acesso, estão minuciosamente descritas?

sim não não aplicável

RNQ1.1 – Caso exista uma documentação de usuário, mas que não contemple uma descrição razoável das funções estabelecidas ou aquelas que o usuário possua acesso, tal documentação omitiu seu principal objetivo, que é mostrar o que o programa oferece, delimitando os acessos por parte do usuário para o melhor funcionamento do programa, privando o usuário de obter o máximo desempenho do produto.

S1.2:Q2 Todo valor-limite é declarado na documentação do usuário, mesmo que tenha sido citado na descrição do produto?

sim não não aplicável

RNQ2 - Haverá uma má utilização do produto por parte do usuário sem os valores-limite especificados. Além disso, talvez o produto não se encaixe no perfil que o

comprador precisa. Por exemplo, caso o produto trabalhe apenas com um tipo de moeda e a empresa possui clientes de várias nacionalidades.

S1.2:Q3 Se a instalação puder ser executada pelo usuário, há um manual de instalação disponível contendo todas as informações necessárias?

sim não não aplicável

RNQ3 - Recomenda-se o desenvolvimento de um manual de instalação, pois, poderá evitar possíveis erros por parte do usuário e a instalação será executada corretamente.

C3:CNM1(S1.2 – Q4)

S1.2:Q3.1 No manual de instalação está estabelecido o espaço de armazenamento mínimo e máximo para a instalação do produto?

sim não não aplicável

RNQ3.1 - O não estabelecimento de tais requisitos, pode conduzir o usuário a problemas na instalação se seu espaço disponível não for o suficiente.

S1.2:Q4 Se algum tipo de manutenção puder ser executada pelo usuário, a documentação de usuário inclui um manual de manutenção de programa contendo todas as informações necessárias para essa manutenção?

sim não não aplicável

RNQ4 - Sem o manual, o usuário não conseguirá fazê-la, mesmo que possa. Com o manual de manutenção o usuário terá condições de fazer uma manutenção sem riscos, “assistida” pelos passos previstos pelo desenvolvedor do produto.

S1.2:Q5 Todas as informações contidas na documentação do usuário foram revistas para evitar possíveis erros, informações incorretas e ambíguas?

sim não não aplicável

RNQ5 - Para maior segurança é recomendável que a revisão seja feita por mais de uma pessoa, que não aquela que participou do desenvolvimento da documentação do usuário, pois um produto com documentação incorreta pode causar um impacto tão negativo, quanto uma documentação inexistente.

S1.2:Q6 Realizou-se uma revisão na documentação do usuário verificando se os documentos utilizados não apresentam contradições internas entre si e com a descrição do produto?

sim não não aplicável

RNQ6 - A realização de revisão após desenvolver a documentação do usuário poderá evitar, por exemplo, problemas de inconsistência, que colocará o usuário em dúvidas sobre qual atitude tomar, mostrando desorganização por parte do desenvolvedor/fornecedor do produto.

S1.2:Q7 Cada termo possui um significado único em toda a documentação?

sim não não aplicável

RNQ7 - Na documentação de qualquer produto, há a necessidade de que os termos utilizados possuam o mesmo significado em toda a extensão do documento, mudando ou não de contexto, evitando assim, equívocos provocados pelo não entendimento do significado de determinado termo.

S1.2:Q8 A documentação do usuário é inteligível para a classe-alvo de usuários a que se destina, utilizando, por exemplo, uma seleção apropriada de termos, exibições gráficas, explicações detalhadas e citando fontes úteis de informações?

sim não não aplicável

RNQ8 - Provavelmente se o usuário for leigo, não conseguirá compreender jargões utilizados pelo pessoal especializado da área de informática, dificultando o entendimento de uma fonte de informação que foi desenvolvida para atender suas dúvidas. O ideal é que possua um texto fácil de ser compreendido, com imagens gráficas, citando outras fontes de informações se for o caso, e assim por diante.

S1.2:Q9 A documentação do usuário possui uma boa apresentação e organização, de tal modo que quaisquer relacionamentos sejam facilmente identificados?

sim não não aplicável

RNQ9 - A apresentação e a organização são duas características marcantes em qualquer tipo de texto. Estas características proporcionam ao usuário condições de localizar itens correlatos ou não, mais rapidamente, isto é, se um usuário deseja encontrar todos os

assuntos pertinentes a um termo “X”, não haverá dificuldades em localizá-lo, por exemplo, na existência de um índice remissivo ou outro meio qualquer.

S1.2:Q9.1 A documentação do usuário possui índice remisso e analítico?

sim não não aplicável

RNQ9.1 - O índice é o método mais eficaz de localização em material impresso ou não. O analítico é uma lista detalhada de assuntos, nomes de pessoas, nomes geográficos, acontecimentos, etc; organizados normalmente em ordem alfabética, com indicação (capítulos, títulos e subtítulos) de sua localização na publicação em que aparecem com suas respectivas páginas. O remissivo (índice de assuntos), é organizado por palavras-chave contidas em todo o material e suas respectivas páginas, capítulos e outros, também organizados em ordem alfabética. Sem a ajuda de qualquer tipo de índice, a localização de itens torna-se inviável. Os índices possuem como objetivo facilitar o manuseio do usuário, permitindo a localização ou registro do assunto desejado.

S1.2:Q9.2 Caso a apresentação do documento não esteja numa forma impressa, há informações sobre o procedimento para impressão?

sim não não aplicável

RNQ9.2 - É importante apresentar informações sobre o procedimento de impressão do documento, pois, o material impresso normalmente é mais fácil de ser manuseado do que um material em mídia, ainda mais se for muito extenso.

S1.3 – Programas e Dados

Neste submódulo são identificadas todas as características referentes aos programas que fazem parte, em conjunto, do produto que será adquirido e dos dados que são necessários para que possa ser explorada toda a funcionalidade do produto.

S1.3:Q1 Se o usuário pode realizar a instalação dos programas por meio de um manual de instalação disponível com o produto adquirido, é possível identificar se os programas funcionam, por exemplo, por meio de: guias de testes fornecidos ou por intermédio de auto-testes correspondentes?

RNQ6 - O ideal é que haja um sistema preventivo que exija do produto uma reação previamente programada. Tal sistema preventivo deveria interferir ou simplesmente não aceitar tais operações.

C6:CNM1(S1.3 – Q7)

S1.3:Q6.1 Os requisitos de confiabilidade são cumpridos ainda que: a capacidade seja explorada até os limites especificados, hajam tentativas feitas para explorar a capacidade além dos limites especificados, sejam introduzidos dados incorretos pelo usuário ou por outros programas listados na descrição do produto ou quando instruções explicadas na documentação de usuário sejam violadas?

sim não não aplicável

RNQ6.1 - Cumprir os requisitos de confiabilidade significa que o produto possui uma previsão e um controle ou interrupção de condições de riscos por períodos previamente estabelecidos pelo sistema, mantendo suas funções em perfeito desempenho.

S1.3:Q6.2 Estão excluídas somente as possibilidades de interrupção do hardware e sistema operacional (por exemplo, a tecla ou combinações de teclas para reinicializar o sistema operacional) que não podem ser controladas por nenhum outro programa?

sim não não aplicável

RNQ6.2 - As teclas ou combinações de teclas que reinicializam o sistema operacional não devem ser trocadas ou desabilitadas, pois o sistema operacional é o meio de comunicação entre a parte física (hardware) e lógica (software – programas), é ele quem gerencia todas funções do computador, então suas funções precisam ser respeitadas para que ele seja capaz de administrá-las.

S1.3:Q6.3 O programa é capaz de reconhecer as violações de sintaxe estabelecida para entrada de dados?

sim não não aplicável

RNQ6.3 - O reconhecimento das violações de sintaxe, garante a consistência dos dados, já que, usuários podem vir a digitar acidentalmente palavras com erros de sintaxe.

C6.3:CNM1(S1.3 – Q7)

S1.3:Q6.3.1 No caso do programa reconhecer uma entrada como errônea ou indefinida, ele não deverá permitir seu processamento como entrada permitida. O sistema possui algum dispositivo de segurança para evitar este tipo de violação?

sim não não aplicável

RNQ6.3.1 - Permitir que uma entrada errada passe por uma entrada permitida, poderá comprometer os dados contidos no banco de dados do programa, pois não existe segurança no sistema, gerando inconsistências nos dados armazenados sob controle do programa e também, originando descredibilidade no produto.

S1.3:Q7 Com relação à usabilidade, são utilizadas algumas partes da Norma da série ISO 9241, em particular as partes 10 e 13, caso haja um acordo para tal entre as partes?

sim não não aplicável

RNQ7 - A utilização de tais normas, pode ajudar o desenvolvedor a criar programas que o usuário tenha maior facilidade e conforto em sua utilização, já que, a usabilidade é o conjunto de atributos que evidenciam o esforço necessário para se poder utilizar o software.

S1.3:Q8 Existem mensagens dentro do sistema que interagem com o usuário?

sim não não aplicável

RNQ8 - A mensagem é uma boa maneira de interação com o usuário, caso, por exemplo, ele cometa um erro, e o sistema necessite alertá-lo.

C8:CNM1(S1.3 – Q9)

S1.3:Q8.1 As mensagens dos programas foram projetadas de forma que o usuário possa diferenciá-las facilmente pelo tipo, por exemplo: confirmações, solicitações, advertências, mensagens de erro, etc?

sim não não aplicável

RNQ8.1 - A função visual dos seres humanos é a mais perspicaz, portanto, padronizar as mensagens por tipos, é uma forma de melhorar e agilizar o entendimento do usuário.

S1.3:Q8.2 O sistema utiliza-se de perguntas para interagir com o usuário?

sim não não aplicável

RNQ8.2 - A interação do usuário com o sistema, é um meio eficiente que o desenvolvedor pode utilizar para comunicar-se com o usuário conduzindo-o em determinados passos, por exemplo, configurar o sistema para atender às suas necessidades.

S1.3:Q8.3 As mensagens de erro fornecem informações detalhadas, explicando a sua causa ou forma de correção (por exemplo, por uma referência a um item na documentação de usuário)?

sim não não aplicável

RNQ8.3 - Se as mensagens não fornecerem subsídios suficientes para o usuário compreender e pesquisar a fundo o problema, deveriam ao menos indicar fontes de pesquisa, do contrário o usuário saberá que tem um problema, mas não conseguirá resolvê-lo.

S1.3:Q8.4 As perguntas, as mensagens e os resultados dos programas são inteligíveis, por exemplo, por meio de: uma seleção adequada de termos, representações gráficas, fornecimento de informações básicas que facilitem o entendimento e explicações dadas por uma função de auxílio?

sim não não aplicável

RNQ8.4 - O não entendimento do usuário pode fazê-lo utilizar o produto de forma incorreta, ou não utilizá-lo, ao perder o interesse devido às dificuldades.

S1.3:Q9 Cada meio de armazenamento de dados apresenta identificação do produto, e em caso de mais de um meio, eles estão distinguidos por um número ou texto?

sim não não aplicável

RNQ9 - A identificação da mídia utilizada no armazenamento de dados é fundamental para não ser confundida com outra sem grande importância, que possa ser alterada ou descartada por outra de necessidade momentânea.

S1.3:Q10 É possível para o usuário quando estiver trabalhando com os programas, verificar/descobrir qual função está sendo executada?

sim não não aplicável

RNQ10 - Tal verificação é importante pois, fornece ao usuário meios de averiguar com certeza que a função realmente está sendo executada e não se perdeu em um dado momento.

S1.3:Q11 Os programas fornecem ao usuário informações claramente visíveis e fáceis de serem lidas?

sim não não aplicável

RNQ11 - Uma das maiores preocupações da interface homem-máquina está relacionada ao quão claros e fáceis os sistemas estão do ponto de vista do usuário. Antigamente, os desenvolvedores construíam sistema individualistas (para si próprios), não se importando com quem iria operá-los, na realidade, naquela época, normalmente os próprios desenvolvedores. O antigo mundo individualista dos profissionais da computação foi submetido a uma grande mudança. Hoje, o grande mérito é reservado ao profissional que consegue tornar seus sistemas claros, descomplicados, bem documentados, num ritmo de trabalho multidisciplinar, de companheirismo, a fim de obter o melhor possível.

S1.3:Q11.1 As informações que guiarão o usuário dentro do programa estão codificadas e agrupadas adequadamente, facilitando seu uso?

sim não não aplicável

RNQ11.1 - O elo de ligação entre as informações é imprescindível. Ao buscar uma determinada informação por uma palavra-chave, por exemplo, se imagina que todas as informações referentes àquela palavra estejam ligadas e que sejam exibidos naquele momento, o que não será possível se as informações não foram organizadas para tal comportamento. Nestes casos, provavelmente o usuário não conseguirá ter acesso à todas as informações que foram projetadas para seu uso.

S1.3:Q12 Há sinais de alerta para o usuário?

sim não não aplicável

RNQ12 - O sinal de alerta é uma boa maneira de advertência ou mesmo aviso para o não esquecimento ou não execução de determinada tarefa.

S1.3:Q13 Os formatos de tela de entrada, de relatórios e de outras entradas e saídas são projetados para serem claros e com boa apresentação e organização, como por exemplo, com: campos alfanuméricos alinhados pela esquerda, campos numéricos alinhados pela direita, em tabelas pontos decimais e vírgulas colocados na mesma linha vertical, limites dos campos sejam reconhecíveis, campos obrigatórios sejam reconhecíveis como tal, na detecção de falhas de entrada, as mesmas sejam imediatamente realçadas na tela ou quando ocorrer uma mudança no conteúdo da tela, o usuário seja alertado por um sinal auditivo ou visual?

sim não não aplicável

RNQ13 - A clareza e a boa apresentação e organização de telas de entrada, de relatórios e de outras entradas e saídas, sugere a padronização das informações, além de prevenir inconsistências, por exemplo, ao evitar que um usuário entre com um campo obrigatório e outro usuário não o faça, por falta de advertências.

S1.3:Q14 Funções, cujas execuções, têm conseqüências graves, possuem meios de reversão?

sim não não aplicável

RNQ14 - Usuários são passíveis de erros, assim sendo, reverter suas ações é uma maneira de permitir que erros graves sejam desfeitos, fornecendo mais um dispositivo de segurança ao usuário.

C14:CNM1(S1.3 – Q15)

S1.3:Q14.1 Em particular, o processo de apagar dados ou sobrepô-los, bem como de interromper um processamento demorado, pode ter conseqüências graves. Além de possuir meios de reversão, tais meios possuem métodos que assegurem a integridade dos dados e programas?

sim não não aplicável

RNQ14.1 - O ideal é que tais programas possuam meios de segurança que previnam o não corrompimento dos dados durante o processo de reversão. Um exemplo deste tipo de reação de programa, pode ser alcançado por meio de uma cópia de segurança que só será atualizada como último comando, a ser confirmado, na saída do programa, entre outros.

S1.3:Q15 Caso não hajam meios de reversão, existem advertências claras sobre as conseqüências, com solicitação de confirmação antes da execução do comando?

sim não não aplicável

RNQ15 - Advertências com solicitação de confirmação para a execução da ação, são medidas de segurança que o sistema poderá conter, contra usuários que acidentalmente ou intencionalmente executassem funções que causariam graves erros.

S1.3:Q16 Se um texto de documentação é exibido em um diálogo, o sistema proporciona ao usuário acesso aos subitens do texto de uma maneira direta, por exemplo: pela seleção em uma tabela de conteúdo exibida na tela, ou por uma função de busca baseada em palavras-chave?

sim não não aplicável

RNQ16 - A apresentação de um diálogo deve fornecer ao usuário caminhos, que percorridos, levem-no à compreensão de uma determinada informação, para uma boa tomada de decisão.

S1.3:Q17 Dentro das características de usabilidade as declarações de eficiência, manutenibilidade e portabilidade citadas na descrição do produto, estão em conformidade com tal descrição?

sim não não aplicável

RNQ17 - De acordo com a norma NBR ISO/IEC 12119 utilizada, tais características de usabilidade não são imprescindíveis. No entanto, os produtos devem estar em conformidade com as declarações citadas em sua descrição. Além de que, a ausência destas características pode repercutir em dificuldades para o usuário quando, por exemplo, houver necessidade de transferir o software de um ambiente para outro e o software não proporcionar tal capacidade (portabilidade).

S2 – Instruções para Teste

Este submódulo do SCM2, de acordo com a Norma NBR ISO/IEC 12119, especificará como um produto deve ser testado em relação aos requisitos da qualidade. Elas incluem tanto o teste das propriedades necessárias a todos os produtos de mesmo tipo, quanto o teste das propriedades especificadas na descrição do produto. Também estão incluídos os testes por inspeção dos documentos e o teste caixa-preta.

Estas instruções descrevem somente o teste funcional (teste caixa-preta). O teste estrutural não está incluído porque requer a disponibilidade do código-fonte.

Somente o produto, no seu ambiente de hardware e software, é testado. A avaliação ergonômica do ambiente de uso do sistema computacional não é considerada nesta Norma, e conseqüentemente na lista de checagem.

S2.1 – Pré-requisitos de teste

Os pré-requisitos de teste são atividades e tarefas que devem ser checadas, antes de iniciar os testes propriamente ditos. A presença de: itens do produto adquirido, componentes do sistema tanto de hardware como de software e por final, o treinamento, caso este tenha sido citado na descrição do produto, fazem parte dos pré-requisitos de teste.

S2.1:Q1 Todos os itens a serem entregues, assim como os documentos de requisitos identificados na descrição de produtos, estão presentes para que se dê início aos testes do pacote de software?

sim

não

não aplicável

RNQ1 - A ausência da documentação, em principal, a especificação do produto, inviabiliza o teste mais comum, que é averiguar se o produto está em conformidade com o que ele se destinou a fazer, na fase inicial de levantamento de requisitos.

C1:CNM1(S2.2 – Q1)

S2.1:Q2 As partes constituintes de todo o ambiente de hardware e software, como identificadas na descrição do produto, estão disponíveis para que o pacote de software seja testado?

sim não não aplicável

RNQ2 - Na ausência de uma das partes do ambiente, o resultado final provavelmente, não terá a mesma validade ou falhará, já que todos os itens não foram testados adequadamente.

S2.1:Q3 Se uma atividade de treinamento é mencionada na descrição do produto, o responsável pelo teste tem acesso ao material e ao programa de treinamento?

sim não não aplicável

RNQ3 - O responsável pelo teste deverá verificar também a conformidade entre a atividade de treinamento mencionada na descrição do produto e o material que de fato foi preparado para executá-la, avaliando se o treinamento preparado condiz com o necessário para satisfazer o que foi proposto.

S2.2 – Atividades de Teste

A atividade de teste é uma operação técnica que consiste na constatação de uma ou mais características de um dado produto, processo ou serviço, de acordo com um procedimento especificado, isto é, envolve a verificação do quanto o produto está em conformidade com o especificado quando da sua requisição ou início de desenvolvimento.

S2.2:Q1 A descrição do produto, a documentação de usuário, os programas e quaisquer dados a serem fornecidos como parte do pacote de software são testados com relação à conformidade com os requisitos de qualidade manipulados no submódulo 1 (S1), do módulo 1 (SCM1) da lista de checagem?

sim não não aplicável

RNQ1 - A conformidade dos requisitos de qualidade do submódulo 1 (S1) do módulo de pacotes de software (SCM1), com o produto final, mostra que o produto realiza o que se propôs a fazer. Atenção, esta afirmação não significa que o produto deverá empregar todos os itens de tal submódulo e sim, que os itens mencionados na descrição do

produto, documentação do usuário, e sobre os programas e dados, deverão constar como empregados no submódulo 1, para que a conformidade entre ambos sejam verificadas no momento dos testes.

S2.2:Q2 Os objetivos dos testes são derivados dos requisitos de qualidade (completude, consistência, etc.) explorados no submódulo (S1) do módulo 1 (SCM1)?

sim não não aplicável

RNQ2 - Requisitos para a qualidade, podem ser tidos como uma ou mais condição(ões) necessária(s) a um determinado fim, para a realização de um objetivo. Condicionar os objetivos dos testes a serem derivados dos requisitos de qualidade, é uma maneira de verificar se os objetivos iniciais de desenvolvimento do produto foram cumpridos.

S2.2:Q2.1 Os objetivos dos testes incluem também todos os requisitos do submódulo (S1) do SCM1, tais como: completude, consistência, entre outros apresentados na lista do submódulo em questão?

sim não não aplicável

RNQ2.1 - Caso um dos requisitos não forem incluídos na verificação (teste), o produto não está em conformidade com o especificado, conseqüentemente não atingiu seus objetivos.

S2.2:Q3 Se outros produtos forem mencionados na descrição de produto, eles só precisam ser testados para as exigências feitas na descrição do produto submetido a teste. É realizado algum tipo de teste em tais produtos?

sim não não aplicável

RNQ3 - Para o melhor funcionamento do produto principal, caso ele utilize outros produtos para compor sua estrutura, é conveniente que os testes sejam estendidos a eles também, já que tal produto irá integrar-se ao pacote de software e que, seu funcionamento inadequado irá prejudicar o funcionamento total do produto principal.

S2.2:Q4 Os detalhes na descrição de produto, na documentação de usuário, nas funções ou dados do produto não precisam ser testados se, de acordo com o julgamento do responsável pelo teste, eles tiverem influência desprezível na adequação da tarefa

indicada ou se o custos dos testes não são justificáveis. Verifica-se a existência de tais casos no produto?

sim não não aplicável

RNQ4 - É melhor dispensar tempo e dinheiro com testes que realmente são necessários, no entanto, deve-se ter muito cuidado ao dispensar um item das atividades de teste.

S2.2:Q4.1 Na existência de detalhes não testados, estes são mencionados nos registros e relatórios de teste?

sim não não aplicável

RNQ4.1 - A ausência de descrição dos detalhes não testados, nos registros e relatórios de testes, pode levar o usuário a concluir que todos os dados, funções e outros, foram submetidos a testes, comprometendo o responsável pelo teste.

S2.2:Q5 Considera-se o cumprimento dos requisitos da qualidade, explorados pela lista de checagem do módulo 1 (SCM1) no S1 (por exemplo: descrição do produto, documentação de usuário e características de programas e dados), no momento do teste?

sim não não aplicável

RNQ5 - Quanto maior a quantidade de requisitos empregados durante a checagem do submódulo 1, mais próximo o produto estará da certificação de conformidade com parte da Norma NBR ISO/IEC 12119, então elaborada por comissões de estudo (CE) nacionais e internacionais, para a padronização de tais processos.

C5:CNM1(S2.2 – Q6)

S2.2:Q5.1 Os programas são testados em todos os ambientes de hardware e software especificados na descrição do produto?

sim não não aplicável

RNQ5.1 - A ausência de testes nos ambientes mencionados na descrição do produto, poderá levar o usuário a adquirir um produto que não poderá executar por questão de: sistema operacional incompatível, quantidade de memória principal insuficiente, dentre outros.

S2.2:Q6 Caso existam variantes do programa todas são testadas?

sim não não aplicável

RNQ6 - As variantes do programa em questão, a rigor, deveriam normalmente, possuir as mesmas funções do programa principal, apenas adequadas a tipos de ambientes diferenciados. No entanto, tal adequação, leva o programa variante a possuir comportamentos diferentes em determinadas situações, por exemplo, na mudança de plataforma de trabalho, o que pode causar impactos, denegrindo a confiabilidade do produto.

C6:CNM1(S2.2 – Q7)

S2.2:Q6.1 As funções que, de acordo com a descrição de produto e documentação de usuário, são idênticas para um grupo de variantes, são testadas cada função em uma variante?

sim não não aplicável

RNQ6.1 - A ausência de testes em funções iguais, independente da variante pode comprometer a funcionalidade do produto, já que cada variante possui um comportamento próprio, devido às diferenças contidas em cada variante.

S2.2:Q7 O guia de teste é uma instrução documentada para o responsável pelo teste, que especifica como deve ou convém que seja testada uma função ou uma combinação de funções. O responsável pelos testes utiliza um guia de teste construído com base na descrição de produto e na documentação do usuário?

sim não não aplicável

RNQ7 - Os guias de teste servem para orientar/conduzir o responsável pelos testes a verificar conformidades com determinados itens, evitar que funções importantes passem por despercebidas, entre outros; visando um bom funcionamento/desempenho do produto.

C7:CNM1(S2.2 – Q8)

S2.2:Q7.1 As instruções do guia de teste também contem informações detalhadas sobre: o objetivo do teste, funções a serem testadas, ambiente de testes e outras condições (detalhes de configuração e trabalho preparatório), dados de testes, procedimentos e comportamento esperado do sistema?

sim não não aplicável

RNQ7.1 - Os guias de testes devem conter além de informações sobre os requisitos para o sistema, também sobre por exemplo, as funções que o desenvolvedor, pelo nível de dificuldades encontradas no desenvolvimento, julga-as como críticas, onde os testes devem ser mais rigorosos.

S2.2:Q7.2 Os guias de testes são construídos metódica e sistematicamente ?

sim não não aplicável

RNQ7.2 - São necessários métodos para construir um guia e, estes métodos têm que ser melhorados sistematicamente. A não atualização dos guias de testes, por exemplo, com relação à mudanças em requisitos, implicará em problemas de inconsistências tanto dos guias como do produto, pois será revisado por meio deles.

S2.2:Q7.3 Se forem dados exemplos na documentação de usuário, eles são utilizados para a criação dos guias de testes, apesar da restrição de que tal guia não deve conter apenas estes exemplos?

sim não não aplicável

RNQ7.3 - Tais exemplos são muitos importantes, para o desenvolvedor novato, por exemplo, até que ele monte sua própria estrutura de testes, poderá partir no início destes meios. Além disso, exemplos de funcionamento de funções, desenvolvidos pelo desenvolvedor do produto, pode mostrar detalhes que o responsável pelos testes não preveria, por não ter participado ativamente do projeto de desenvolvimento.

S2.2:Q7.4 Materiais adicionais (por exemplo, programas-fonte) não precisam ser considerados, a menos que as declarações na descrição de produto e na documentação de usuário não contenham as informações suficientes. Tais informações são insuficientes, havendo necessidade de considerar materiais adicionais?

sim não não aplicável

RNQ7.4 - O ideal é que no momento de teste estes materiais já tenham sido referenciados claramente, do contrário o responsável pelo teste poderá requerer uma revisão do material, caso ele mesmo não o reorganize.

S2.2:Q7.5 Os guias de testes disponibilizados pelo fornecedor do pacote de software podem ser usados, no entanto, as atividades de testes não devem se restringir a eles. Existem outras atividades paralelas?

sim não não aplicável

RNQ7.5 - O responsável pelos testes deve utilizar também outras atividades que não aquelas exigidas/informadas no guia de teste, evitando assim que problemas que passaram despercebidos pelo fornecedor do pacote sejam revelados no momento de teste.

S2.2:Q8 Se, de acordo com a descrição do produto, a instalação puder ser executada pelo usuário, há meios de verificar se a instalação foi bem sucedida, conforme descrito no seu manual de instalação?

sim não não aplicável

RNQ8 - O manual de instalação deve conter os passos corretos para que o usuário possa executar a instalação sem problemas, pois, se a intenção era de não ter custos com suporte para instalação, caso o usuário não tenha êxito, provavelmente ele entrará em contato com o pessoal de suporte, podendo envolver custos ainda maiores se, por exemplo, uma equipe de suporte tiver que ser deslocada para outra cidade.

S2.2:Q9 O usuário é advertido para adequar seu ambiente de hardware e software onde os programas serão instalados, ao sistema computacional estabelecido na descrição do produto, sendo igual ou superior?

sim não não aplicável

RNQ9 - Um ambiente inadequado poderá diminuir o desempenho do pacote de software e até mesmo não conseguir executá-lo. Inadequado significa: com um hardware inferior ao pedido na descrição do produto, software básico incompatível com a plataforma do pacote de software, entre outros.

S2.2:Q10 Os guias de teste consideram as combinações de funções que são representativas para a tarefa?

sim não não aplicável

RNQ10 - As combinações de funções, por exemplo, funções que dependem de outras funções para serem executadas ou que farão parte de outras, devem ser testadas cuidadosamente, pois o não funcionamento de uma, poderia ocasionar uma reação em cadeia, com impacto em outras e dependendo da gravidade, no sistema como um todo.

S2.2:Q11 Os programas são testados para todos os valores-limite, tanto de hardware, quanto de software, aos quais estes valores se aplicam?

sim não não aplicável

RNQ11 - Testar valores-limite, implica em garantir a total e irrestrita operacionalidade dos programas, descritos na especificação do produto.

S2.2:Q12 São usadas nos testes, entradas ou seqüências de comandos que estão explicitamente desaprovadas ou declaradas como proibidas na documentação do usuário?

sim não não aplicável

RNQ12 - Na ausência de testes com entradas proibidas ou seqüência de comandos inadequada, não será verificado o funcionamento do sistema de reconhecimento de entradas inválidas.

S2.3 – Registros de Teste

Os registros de teste contem informações sobre todas as atividades que foram desenvolvidas durante os testes incluindo especificações sobre os testes executados, os resultados e a equipe que os realizou.

S2.3:Q1 Faz parte do pacote de software os registros de testes realizados pelo desenvolvedor?

sim não não aplicável

RNQ1 – Os registros de testes identificam os processos pelo qual os softwares passaram para que fossem caracterizados como produtos que atendem as especificações de seus requisitos. Sua ausência aparenta um não-comprometimento com o adquirente, já que o mesmo não conseguirá averiguar se tais atividades foram ou não realizadas e como o foram.

C1:CNM1(S2.4 – Q1)

S2.3:Q2 Os registros de testes contém informações suficientes para permitir a repetição do teste?

sim não não aplicável

RNQ2 - A impossibilidade da repetição dos testes, negará ao adquirente a possibilidade de averiguar o desempenho do produto em condições não previstas nos requisitos especificados, como por exemplo, se há necessidade de integração da aquisição a um conjunto de softwares.

S2.3:Q2.1 Nos registros, incluem-se um plano ou especificação de teste contendo guias de teste?

sim não não aplicável

RNQ2.1 - A ausência de um plano ou uma especificação de testes contendo guias de testes, resulta em maior dificuldade no momento da repetição dos testes se necessário, em principal se forem executados por outras pessoas que não aquelas que participaram do projeto.

S2.3:Q2.2 Todos os resultados associados com os guias de teste, incluindo as falhas que ocorreram durante os testes, estão descritos no registros de teste?

sim não não aplicável

RNQ2.2 - O registro de resultados de sucesso e também as falhas ocorridas e provavelmente corrigidas, deixará o adquirente mais confiante no produto que está adquirindo, tendo consciência de que o produto foi testado e possíveis falhas foram corrigidas. As falhas anotadas também poderão representar um foco de novos testes por parte do adquirente que verificará após a repetição de todas as atividades de teste que o produto realmente foi corrigido e é confiável.

S2.3:Q2.3 Nos registros de teste, encontra-se a identificação do pessoal envolvido no teste?

sim não não aplicável

RNQ3 - As declarações ou observações são importantes tanto para a equipe de teste, quanto para o adquirente que será capaz de averiguar todos os passos pelos quais seu produto passou desde a descrição do produto até as atividades de teste sob o ponto de vista das pessoas envolvidas diretamente em cada parte do sistema, que anotando possíveis decisões importantes num momento de uma integração com outro sistema.

S2.4:Q4 Em cada página do relatório de teste contem a identificação do relatório (laboratório de teste, identificação do produto, data do relatório) e o número total de páginas?

sim não não aplicável

RNQ4 - As informações mencionadas acima são características de uma boa apresentação e organização. A não identificação das páginas revela dificuldades para o usuário entrar em contato com o fornecedor ou identificar o produto.

S2.4:Q5 O relatório de teste inclui uma declaração de que os resultados de teste se referem somente aos itens testados?

sim não não aplicável

RNQ5 - É um meio que o responsável pelas atividades de teste possui de não se responsabilizar pelas funções ou produtos compartilhados que não foram testados.

S2.4:Q6 O relatório de teste também menciona os direitos autorais do laboratório, sobre os resultados de teste, não podendo aparecer parcialmente reproduzido, a não ser que haja aprovação por escrito do laboratório?

sim não não aplicável

RNQ6 - O laboratório, normalmente, não aprova a divulgação parcial dos resultados, com intuito de precaver-se de fraudes, a não ser que haja aprovação por escrito do laboratório.

S2.5 – Teste de Acompanhamento

Testes de acompanhamento consiste em testar novamente um produto que já foi testado. Tais testes são realizados quando há alteração de determinada função ou programa devido a não conformidade aos requisitos, falha na função, etc.

S2.5:Q1 Os testes de acompanhamento fazem parte dos documentos que acompanham o pacote de software?

sim não não aplicável

RNQ1 – Quando há alterações, ainda que pequenas, há necessidade de reteste, pois, as modificações podem influir nas outras partes do sistema, alterando seu desempenho, confiabilidade e até fazendo com que ele não funcione sob condições de extrapolação de seus limites e outros.

C1:CNM1(S2.6 – Q1)

S2.5:Q2 Quando um produto é retestado leva-se em consideração os testes anteriores?

sim não não aplicável

RNQ2 - Os testes anteriores devem ser considerados para verificar em quais itens de teste o programa apresentou problemas.

S2.5:Q3 Todas as partes modificadas nos documentos, funções e dados são testadas como se fossem um novo produto?

sim não não aplicável

RNQ3 - Os testes de correções, têm que ser tão rigorosos quanto os primeiros, já que, nos primeiros testes foram verificadas falhas ou desconformidades iniciais.

S2.5:Q4 Todas as partes inalteradas com possibilidade de serem influenciadas pelas partes alteradas ou por modificações nos requisitos de hardware e software (de acordo com o conhecimento especializado do responsável pelo teste) são testadas como se fossem um novo produto?

sim não não aplicável

RNQ4 - Alterações ou modificações de hardware e software necessitam ser testadas com todos os cuidados tomados com os testes iniciais de produto novo, pois, após sua mudança obtem-se um novo produto a partir das novas funções e/ou especificações.

S2.5:Q5 Se o responsável pelos testes não verifica necessidade de novos testes em muitas funções, ainda assim como medida preventiva, todas as outras partes são testadas, considerando-se, pelo menos, casos de testes com seleção por amostragem?

sim não não aplicável

RNQ5 - Mudanças em sistemas é uma atividade sempre de muito risco, seu impacto muitas vezes levam o desenvolvedor a deixar de lado aquele programa ou parte dele e recomeçar. Assim sendo, mesmo que algumas funções não estejam diretamente ligadas à função alterada, uma maneira de diminuir riscos e não aumentar em demasia os custos é escolher um rol de casos de testes já realizados, ou seja, uma amostragem para a repetição.

S3 Exemplo de uma Descrição de um Produto

A Norma NBR ISO/IEC 12119, traz em um de seus anexos, um exemplo de uma descrição de produto em conformidade com os padrões por ela estabelecidos. O exemplo descreve um pacote de software simples e fictício, com o objetivo de mostrar as informações que devem estar presentes em toda descrição de produto.

Formulário de descrição do produto:

FIREatWORK Versão 2.6

FIREatWORK – Proteção de telas (screen saver) com utilização de senha.

O programa FIREatWORK protegerá sua tela através de fantástica exibição – em monitores coloridos – de fogos de artifício, enquanto você estiver trabalhando com o seu computador. Se você registrar uma senha você será alertado se alguém tentar utilizar o seu computador em sua ausência.

O FIREatWORK instala-se em memória. A proteção da tela será iniciada quando você deixar de pressionar alguma tecla e de mover o mouse por um período de tempo passível de ajuste. Ela será desativada tão logo você pressione alguma tecla ou mova o mouse. Contudo, se você tiver definido uma senha, o FIREatWORK esperará que ela seja digitada.

Você pode definir sua configuração favorita quanto ao:

- tempo que o FIREatWORK esperará para acionar a proteção (1 a 999 minutos, ou não aciona);
- número de fogos de artifício que explodirão em conjunto (1 a 19).

Para tanto, o FIREatWORK usará linha de comando, ou janela, da mesma forma que seu sistema operacional faz para mudança de data e hora do sistema.

Da mesma maneira você pode definir uma senha contendo de 6 a 45 caracteres. Feito isso, se o FIREatWORK parar após a digitação de um caractere qualquer, ou não parar ao ser digitada a sua senha, isto significa que algo interrompeu o FIREatWORK (por exemplo, pelo desligamento da energia) e alguém o reiniciou sem a utilização de senha, ou com outra diferente.

Você pode conduzir cópias de segurança do programa e da configuração, através do seu sistema operacional. A senha não será salva.

Alguns detalhes técnicos:

- O FIREatWORK pode ser executado em um computador pessoal Quince Hardcore 119xi e em computadores compatíveis, com no mínimo 1MB de memória principal e uma unidade de disquete de 3 ½ ou 5 ¼ polegadas, com capacidade mínima de 720 kb. Ele não necessita de unidade de disco rígido e suporta mouse Mini-Rat serial ou paralelo, ou qualquer outro compatível, mas o mouse não é requerido.

- O FIREatWORK necessita de placa gráfica Hercules Deluxo ou PowerEGA 16+, ou qualquer outra compatível com esses dois modelos.

- O FIREatWORK executa sob o sistema operacional B.I.T.S 1.01 ou Gnome 3.0, ou qualquer outro compatível com um desses dois.

- Quando encomendar o FIREatWORK, favor informar:

- se você deseja a variante para o sistema B.I.T.S ou para o Gnome;
- se você deseja um FIREatWORK em disquete de 3 ½ ou 5 ¼ polegadas.

O pacote consiste no programa (módulo de carga) em um disquete e um livrete de documentação, o qual inclui o guia de instalação.

É importante saber que:

- não é necessário qualquer conhecimento especial para instalar ou usar o FIREatWORK;
- as mensagens do programa e a documentação estão descritas em português;

- o FIREatWORK está totalmente em conformidade com a NBR ISO/IEC 12119 – Tecnologia de informação – Pacote de software – Teste e requisitos de qualidade.

Serviços de manutenção e de suporte para operação do produto não são fornecidos.

O FIREatWORK pode ser obtido através de:

PyroManiac Klaus P Schmidt Ltdd

33 Bell Street

Telefone (022) 845 3902

Borgheim, SU 53844

4.5 SCM2 (SOFTCHECK MÓDULO2): UTILITÁRIOS

O módulo2 (SCM2) foi construído para conter funções auxiliares de interesse geral. É constituído por um glossário de termos que foram direta ou indiretamente utilizados na lista de checagem. No entanto, compõem também o glossário outros termos relativos à qualidade de software, tornando-se assim, fonte de pesquisa de termos técnicos, não somente para o estágio atual do SOFTCHECK, mas também para estágios futuros.

S1 Glossário

O glossário desenvolvido no submódulo 1 do módulo 2 (SCM2), possui como fonte de informação o Anexo A (Definições de outras normas) contido na norma NBR ISO/IEC 12119 (Tecnologia de informação – Pacotes de software – Testes e requisitos de qualidade) e também a Norma NBR ISO 8402 (Gestão da qualidade e garantia da qualidade – Terminologia).

A Norma NBR ISO 8402, possui além da descrição dos termos técnicos, situações e/ou exemplos de empregabilidade dos termos. No submódulo em questão, serão encontrados apenas os significados de cada termo. Além disso, algumas alterações foram necessárias ao descartar termos em duplicidade, haja vista, que os mesmos também foram referenciados na fonte citada acima da Norma NBR ISO/IEC 12119.

O glossário foi definido sem agrupamentos referentes a qualquer tipo de característica, módulo, submódulo ou quanto à sua empregabilidade. Os termos

encontram-se relacionados simplisticamente em ordem alfabética, possuindo apenas numeração própria com uma notação de identificação antes do número e termo, determinando que o conteúdo sucedido é referente a, por exemplo, G1 - o primeiro termo do glossário.

S1.2 Termos e Definições

De acordo com a Norma NBR ISO 8402, no campo da qualidade muitos termos de uso corrente são empregados com significado específico ou mais restrito se comparado com as definições constatadas nos dicionários. Com o mesmo intuito da Norma acima citada, o glossário do SOFTCHECK possui como objetivo esclarecer e padronizar termos utilizados na gestão da qualidade.

O glossário então é composto de termos e definições relativos aos conceitos da qualidade, aplicáveis a todas as áreas, para elaboração e utilização de normas sobre qualidade e para entendimento mútuo nas comunicações nacionais e internacionais.

O glossário do SOFTCHECK está preparado para amparar uma série de outras normas da qualidade, além da NBR ISO/IEC 12119, já utilizada no desenvolvimento da lista em questão.

Quadro 18 – Glossário da lista de checagem

Identificador	Definição
G1	Ação corretiva: Ação implementada para eliminar as causas de uma possível não-conformidade, de um defeito ou de outra situação indesejável existente, a fim de prevenir sua repetição.
G2	Ação preventiva: Ação implementada para eliminar as causas de uma possível não-conformidade, de um defeito ou de outra situação indesejável, a fim de prevenir sua ocorrência.
G3	Análise crítica pela Administração: Avaliação formal, pela Alta Administração, do estado e da adequação do sistema da qualidade, em relação à política da qualidade e seus objetivos.
G4	Análise crítica do contrato: Atividades sistemáticas executadas pelo fornecedor, antes da assinatura do contrato para garantir que os requisitos para a qualidade estão adequadamente definidos, sem ambigüidade e documentados, e que eles podem ser atendidos pelo fornecedor.
G5	Análise crítica do projeto: Exame documentado completo e sistemático de um projeto para avaliar sua capacidade de atender os requisitos para a qualidade, identificar problemas, se houver, e propor o desenvolvimento de soluções.

Identificador	Definição
G6	Auditoria da qualidade: Exame sistemático e independente, para determinar se as atividades da qualidade e seus resultados estão de acordo com as disposições planejadas, se estas foram implementadas com eficácia e se são adequadas à consecução dos objetivos. Frequentemente chamadas de auditoria da qualidade, auditoria da qualidade do produto, auditoria da qualidade do processo, auditoria da qualidade do serviço, auditoria do sistema da qualidade . As auditorias da qualidade são executadas por pessoas que não têm responsabilidade direta nas áreas a serem auditadas, mas que, de preferência, trabalhem em cooperação com o pessoal destas áreas.
G7	Auditado: Organização que está sendo auditada.
G8	Auditor líder: Um auditor da qualidade designado para conduzir uma auditoria da qualidade .
G9	Auditor da qualidade: Pessoa qualificada para efetuar auditorias da qualidade .
G10	Auto-inspeção: Inspeção , realizada pelo próprio executante, do seu próprio trabalho, de acordo com regras específicas.
G11	Avaliação da qualidade: Exame sistemático para determinar até que ponto uma entidade é capaz de atender os requisitos especificados. Um qualificativo pode ser adicionado ao termo avaliação da qualidade , dependendo do escopo (por exemplo, processo, pessoal, sistema) e do momento em que ocorre a avaliação da qualidade (por exemplo, antes do contrato), tal como: “ avaliação da qualidade do processo, antes do contrato ”.
G12	Ciclo da qualidade: Modelo conceitual de atividades interdependentes que influenciam a qualidade , nas diferentes fases, variando desde a identificação das necessidades até a avaliação de atendimento dessas necessidades. A “ espiral da qualidade ” é um conceito similar.
G13	Classe: Categoria ou classificação atribuída às entidades que têm a mesma função, mas que diferem entre si quanto aos requisitos para a qualidade .
G14	Cliente: Destinatário de um produto provido pelo fornecedor .
G15	Compatibilidade: Capacidade de entidades serem usadas em conjunto, sob condições específicas, para atender requisitos pertinentes.
G16	Comprador: Cliente em uma situação contratual. O comprador também é chamado de “ segunda parte ”.
G17	Concessão pós-produção: Autorização escrita para usar ou liberar um produto não-conforme em relação aos requisitos especificados.
G18	Confiabilidade, característica de um produto: Conjunto de atributos que evidenciam a capacidade do software de manter seu nível de desempenho sob condições estabelecidas durante um período de tempo estabelecido.

Identificador	Definição
G19	Configuração: A maneira na qual o hardware e o software de um sistema de processamento de informação são organizados e interconectados.
G20	Conformidade: Atendimento a requisitos especificados.
G21	Contrato, análise crítica: Atividades sistemáticas executadas pelo fornecedor, antes da assinatura do contrato, para garantir que os requisitos para a qualidade estão adequadamente definidos
G22	Contratado: Fornecedor em uma situação contratual. O contrato também é chamado de “ primeira parte ”.
G23	Controle da qualidade: Técnicas e atividades operacionais usadas para atender os requisitos para a qualidade .
G24	Contexto de trabalho: Combinação de pessoa e equipamento, atuando juntos no processo produtivo, para executar uma tarefa, no espaço e no ambiente de trabalho, sob condições determinadas pela tarefa.
G25	Correção: Se refere a um reparo , um retrabalho ou um ajuste, e está relacionada a disposição de uma não-conformidade existente.
G26	Corretiva, ação: Ação implementada para eliminar as causas de um não-conformidade , defeito ou de outra situação indesejável existente, a fim de prevenir sua repetição. Está relacionada à eliminação das causas de uma não-conformidade .
G27	Custos relativos à qualidade: Custos incorridos para garantir e assegurar a qualidade , bem como aqueles decorrentes das perdas, quando essa qualidade não é obtida.
G28	Dados: Representação reinterpretabil da informação de maneira formalizada, adequada para comunicação, interpretação ou processamento.
G29	Defeito: Não-atendimento de um requisito de uso pretendido ou de uma expectativa razoável, inclusive quanto à segurança .
G30	Demonstração, grau de: Extensão em que a evidência é gerada com o objetivo de prover a confiança de que os requisitos especificados são atendidos.
G31	Dependabilidade: Termo coletivo usado para descrever o desempenho quanto à disponibilidade e seus fatores de influência: confiabilidade, manutenibilidade e logística de manutenção.
G32	Desvio, pré-produção; permissão de: Autorização escrita que permite o desvio dos requisitos especificados originalmente para um produto , antes da sua produção.
G33	Disposição de não-conformidade: Ação a ser implementada na entidade não-conforme , de modo a resolver a não-conformidade .
G34	Eficiência, característica de um produto: Conjunto de atributos que evidenciam o relacionamento entre o nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas.

Identificador	Definição
G35	Entidade; item: Todo elemento que pode ser considerado e descrito individualmente. Uma entidade pode ser, por exemplo: uma atividade ou um processo ; um produto , uma organização , um sistema ou uma pessoa ou uma combinação dos elementos acima.
G36	Especificação: Documento que define requisitos. Uma especificação deve referir-se, ou incluir desenhos, a modelos ou outros documentos apropriados, bem como indicar os meios e os critérios segundo os quais a conformidade pode ser verificada.
G37	Estrutura organizacional: Responsabilidades, vinculações hierárquicas e relacionamentos, configurados segundo um modelo, através do qual uma organização executa suas funções.
G38	Evidência objetiva: Informações cuja veracidade pode ser comprovada com base em fatos obtidos através de observação, medição, ensaio ou outros meios.
G39	Fornecedor: Organização que fornece um produto ao cliente . Numa situação contratual o “ fornecedor ” pode ser chamado de “ contratado ”.
G40	Funcionalidade, característica de um produto: Conjunto de atributos que evidenciam a existência de um conjunto de funções e suas propriedades especificadas. As funções são as que satisfazem as necessidades explícitas ou implícitas.
G41	Garantia da qualidade: Conjunto de atividades planejadas e sistemáticas, implementadas no sistema da qualidade e demonstradas como necessárias, para prover confiança adequada de que uma entidade atenderá os requisitos para a qualidade .
G42	Gestão da qualidade: Todas as atividades da função gerencial que determinam a política da qualidade , os objetivos e as responsabilidades, e os implementam por meio como planejamento da qualidade, controle da qualidade, garantia da qualidade e melhoria da qualidade dentro do sistema da qualidade .
G43	Gestão da qualidade total: Modo de gestão de uma organização , centrado na qualidade , baseado na participação de todos os seus membros, visando ao sucesso a longo prazo, através da satisfação do cliente e dos benefícios para todos os membros da organização e para a sociedade.
G44	Grau de demonstração: Extensão em que a evidência é gerada com o objetivo de prover a confiança de que os requisitos especificados são atendidos.
G45	Inspeção: Atividades tais como: medição, exame, ensaio, verificação com calibres ou padrões, de uma ou mais características de uma entidade , e a comparação dos resultados com requisitos especificados, a fim de determinar se a conformidade para cada uma dessas características é obtida.

Identificador	Definição
G46	Intercambialidade: Capacidade de uma entidade ser usada no lugar de outra, sem modificação, para atender os mesmos requisitos.
G47	Interface: Limite compartilhado entre duas unidades funcionais, definido por características funcionais, características de interconexão física comuns, características de sinal, e outras características, quanto apropriado.
G48	Interface com o usuário: Uma interface que permite que as informações que as informações sejam passadas entre o usuário e componentes de hardware ou de software de um sistema computacional.
G49	Líder, auditor: Pessoa qualificada para efetuar auditorias da qualidade . Um auditor da qualidade designado para conduzir uma auditoria da qualidade é chamado de “ auditor-líder ”.
G50	Manual da qualidade: Documento que declara a política da qualidade e descreve o sistema da qualidade de uma organização . Um manual da qualidade , para se adaptar às necessidades de uma organização , pode variar de detalhe e formato. Ele pode compreender mais de um documento. Dependendo do escopo do manual, um qualificativo pode ser usado, como, por exemplo, manual de garantia da qualidade , manual de gestão da qualidade .
G51	Manual de manutenção de programa: Documento que fornece toda a informação necessária para a manutenção de um programa.
G52	Manutenção de sistema: Alteração de um sistema para corrigir defeitos, para melhorar o desempenho, ou para adaptar o sistema às mudanças de ambiente ou de requisitos.
G53	Manutenibilidade, característica de um produto: Conjunto de atributos que evidenciam o esforço necessário para fazer modificações especificadas no software.
G54	Meio de dados: Material no qual os dados podem ser armazenados e do qual podem ser recuperados.
G55	Melhoria da qualidade: Ações implementadas em toda a organização a fim de aumentar a eficácia e a eficiência das atividades e dos processos , para proporcionar benefícios adicionais tanto à organização quanto aos clientes .
G56	Modelo para garantia da qualidade: Conjunto de requisitos padronizados ou selecionados do sistema da qualidade , combinados para atender as necessidades de garantia da qualidade em dada situação.
G57	Não-conformidade: Não-atendimento de um requisito especificado. A definição abrange afastamento ou ausência de uma ou mais características da qualidade (incluindo características de dependabilidade) ou elementos do sistema da qualidade em relação a requisitos especificados.
G58	Não-conformidade, disposição de: Ação a ser implementada na entidade não-conforme, de modo a resolver a não-conformidade .

Identificador	Definição
G59	Observação da auditoria da qualidade: Constatação do fato, feita durante uma auditoria da qualidade e consubstanciada por evidência objetiva .
G60	Organização: Companhia, corporação, firma, empresa ou instituição, ou parte destas, pública ou privada, sociedade anônima, limitada ou com outra forma estatutária que tem funções e estrutura administrativa próprias.
G61	Pacote de software: Conjunto completo e documentado de programas fornecidos a diversos usuários para uma aplicação ou função genérica.
G62	Perdas da qualidade: Perdas ocasionadas pela não-utilização do potencial de recursos nos processos e atividades.
G63	Plano da qualidade: Documento que estabelece as práticas, os recursos e a seqüência de atividades relativa à qualidade de um determinado produto , projeto ou contrato. Dependendo do escopo do plano, um qualificativo pode ser usado, como, por exemplo, “ plano da garantia da qualidade, plano de gestão da qualidade ”.
G64	Planejamento da qualidade: Atividades que determinam os objetivos e os requisitos para a qualidade , assim como os requisitos para a aplicação dos elementos que compõem o sistema da qualidade .
G65	Ponto de parada: Ponto, definido em documento apropriado, além do qual uma atividade não pode prosseguir sem a aprovação de uma organização ou autoridade designadas.
G66	Política da qualidade: Intenções e diretrizes globais de uma organização relativas à qualidade , normalmente expressas pela Alta Administração.
G67	Portabilidade, característica de um produto: Conjunto de atributos que evidenciam a capacidade do software ser transferido de um ambiente para outro.
G68	Prestação de serviço: Atividades do fornecedor necessárias ao provimento do serviço .
G69	Procedimento: Forma especificada de executar uma atividade.
G70	Processo: Conjunto de recursos e atividades inter-relacionados que transformam insumos (entradas) em produtos (saídas).
G71	Processo de qualificação: Processo que demonstra se uma entidade é capaz de atender os requisitos especificados. O termo “ qualificação ” é usado às vezes para designar este processo .
G72	Produto: Resultado de atividades ou processos . O termo produto pode incluir serviço , materiais e equipamentos, materiais processados, informações, ou uma combinação destes.
G73	Produto, responsabilidade civil pelo fato do: Termo genérico usado para descrever os ônus que recaem sobre um fornecedor , ou outros, para fazer compensações por perdas relativas a danos pessoais, materiais ou outros, causados por um produto .

Identificador	Definição
G74	Programa (de computador): Unidade sintática que está em conformidade com as regras de uma linguagem de programação particular, composta de declarações, comandos ou instruções necessárias para executar uma certa função ou tarefa, ou ainda solucionar um problema.
G75	Programa utilitário, rotina utilitária: Rotina (um programa computacional) que fornece serviços gerais, freqüentemente necessários. Exemplos: uma rotina de entrada, um programa de diagnóstico, um programa de classificação.
G76	Qualidade: Totalidade de características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas, de acordo com a norma em vigor para elaboração deste glossário.
G77	Qualificado: Designação do estado de uma entidade que demonstra a capacidade de atender os requisitos especificados.
G78	Rastreabilidade: Capacidade de recuperação do histórico, da aplicação ou da localização de uma entidade , por meio de identificações registradas.
G79	Registro: Documento que fornece evidência objetiva de atividades realizadas ou resultados obtidos. Um registro da qualidade fornece evidências objetivas sobre a extensão do atendimento dos requisitos para a qualidade por exemplo: registro da qualidade de um produto , ou eficácia de funcionamento de um elemento do sistema da qualidade (por exemplo: um registro do sistema da qualidade).
G80	Reparo: Ação implementada sobre um produto não-conforme de modo que este passe a satisfazer os requisitos de uso previsto, embora possa não atender aos requisitos originalmente especificados.
G81	Requisitos para a qualidade: Expressão das necessidades ou sua tradução num conjunto de requisitos, explicitados em termos quantitativos ou qualitativos, objetivando definir as características de uma entidade a fim de permitir sua realização e seu exame.
G82	Requisitos da sociedade: Obrigações resultantes de leis, regulamentos, regras, códigos, estatutos e outras considerações.
G83	Retrabalho: Ação implementada sobre um produto não-conforme de modo que ele atenda os requisitos especificados.
G84	Segurança: Estado no qual o risco de danos pessoais ou materiais está limitado a um nível aceitável.
G85	Serviço: Resultado gerado por atividades na interface fornecedor e cliente , e por atividades internas do fornecedor para atender às necessidades do cliente .
G86	Sistema da qualidade: Estrutura organizacional, procedimentos, processos e recursos necessários para implementar a gestão da qualidade .

Identificador	Definição
G87	Software: Programas, procedimentos, regras e qualquer documentação associada, pertinente à operação de um sistema computacional.
G88	Software de sistema: Software independente da aplicação que suporta a execução de um software aplicativo.
G89	Subcontratado; Subfornecedor: Organização que fornece um produto ao fornecedor . O subcontratado pode ser chamado também de “ subfornecedor ”.
G90	Supervisão da qualidade: Acompanhamento e verificação contínuos do estado de uma entidade e análise de registros , para garantir que os requisitos especificados estão sendo atendidos.
G91	Tarefa: Resultado esperado em um contexto de trabalho.
G92	Teste: Operação técnica que consiste na constatação de uma ou mais características de um dado produto, processo ou serviço, de acordo com um procedimento especificado.
G93	Teste, dados de: Dados utilizados em um problema de teste.
G94	Teste, método de: Procedimento técnico especificado para executar um teste.
G95	Teste, plano de (ou plano de avaliação e teste de sistema): Plano que estabelece requisitos detalhados, critérios, metodologia geral, responsabilidades e planejamento geral para testes e avaliação de um sistema.
G96	Teste, problema de: Problema com uma solução conhecida usada para determinar se uma unidade funcional está operando corretamente.
G97	Teste, relatório: Documento que apresenta resultados de testes e outras informações relevantes para um teste.
G98	Unidade funcional: Entidade de hardware ou de software ou de ambos, capaz de executar um propósito específico.
G99	Usabilidade, característica de um produto: Conjunto de atributos que evidenciam o esforço necessário para se poder utilizar o software, bem como o julgamento individual desse uso, por um conjunto explícito ou implícito de usuários.
G100	Validação: Confirmação, por exame e fornecimento de evidência objetiva , de que os requisitos especificados, para um determinado uso pretendido, são atendidos. O termo “ validado ” é usado para designar o estado após a validação . No projeto e desenvolvimento, a validação se refere ao processo de examinar um produto para determinar sua conformidade com as necessidades do usuário.
G101	Variantes: Aqui tratadas, como as variações que possa existir do mesmo produto para ambientes (softwares e hardwares) diferentes. Por exemplo, as variantes do sistema operacional Windows, existindo o Windows para desktop, laptop e NT, para redes.

Identificador	Definição
G102	Verificação: Confirmação, por exame e fornecimento de evidência objetiva , do atendimento aos requisitos especificados. O termo “ verificado ” é usado para designar o estado após a verificação . No projeto e desenvolvimento, a verificação refere-se ao processo de examinar o resultado de dada atividade para determinar sua conformidade com os requisitos estabelecidos para a mesma atividade.

4.6 COMPARAÇÃO ENTRE MODELOS APRESENTADOS E O SOFTCHECK

Apresenta-se abaixo um quadro comparativo entre algumas das normas relacionadas aos processos e produtos de software e a lista de checagem descrita neste estudo, assinalando os principais aspectos numa abordagem geral.

Em tal averiguação, foram utilizados como base inicial dois artigos de Tsukumo et. al., mobilizando aspectos e alguns dados abordados no quadro comparativo de número dois [Tsukumo, 1996, 219-234]. Os dados foram complementados e outros aspectos foram adicionados devido à relevância dos mesmos para a apresentação da lista de checagem, compondo assim o Quadro 19.

Os nomes de aspectos que não são auto-explicativos encontram-se descritos abaixo:

- Abordagem: caracteriza resumidamente como e sobre o que cada modelo atua para atingir seu objetivo;
- Público Alvo: caracteriza os tipos de usuários a que cada modelo se restringe;
- Organizações Alvo: caracteriza a que tipo/porte de organização melhor se aplica o modelo;
- Definição de Processos: quantifica o desmembramento e a classificação de processos;
- Flexibilidade: indica a possibilidade de adaptação dos aspectos definidos pelo método;
- Fase de atuação: identifica o estágio em que o modelo pode ser utilizado (análise de requisitos, projeto, desenvolvimento, etc.);
- Nível de dificuldade de interpretação: define o grau de dificuldade para o entendimento textual por parte do usuário;
- Tempo: indica o período que será gasto no uso das normas e da lista;

- **Objetividade:** capacidade do usuário/intérprete de atingir os propósitos expostos nas normas ou listas, possuindo margem mínima ou inexistente de ambigüidade relativa às interpretações, pois do contrário, haverá uma violação no significado da palavra norma e/ou padrão;

- **Condução:** é definida como a capacidade conduzir o usuário, fazer com que desperte nele interesse em utilizar tal técnica;

- **Facilidade de uso:** pessoas preferem muito mais responder questões objetivas a tentar ler e interpretar questões subjetivas.

Quadro 19 - Comparação entre os modelos e o SoftCheck.

Aspectos abordados	Normas			SoftCheck
	NBR ISSO 9000-3	NBR ISSO/IEC 12207	NBR ISSO/IEC 12119	
Objetivo	Certificar a organização de acordo com padrões estabelecidos em situações de contrato de fornecimento de software.	Estabelecer uma terminologia e um entendimento comum entre todos os envolvidos com software.	Estabelecer instruções para processos, determinando características baseadas nos requisitos de qualidade e instruções de testes.	Estabelecer instruções para processos, determinando características baseadas nos requisitos de qualidade e instruções de testes.
Abordagem	Verificação de conformidade de processos a padrões documentados.	Definição dos processos para aquisição, fornecimento, desenvolvimento, operação e manutenção de software.	Checar os requisitos de qualidade e auxiliar nos testes dos pacotes de software.	Checar os requisitos de qualidade e auxiliar nos testes dos pacotes de software, de forma clara e concisa, estabelecendo uma ligação estreita com o usuário, por meio de recomendações.

Aspectos abordados	Normas			SoftCheck
	NBR ISO 9000-3	NBR ISO/IEC 12207	NBR ISO/IEC 12119	
Público Alvo	Compradores e fornecedores ao desenvolverem, e fornecerem e manterem produtos de software, quando em um contrato entre duas partes exigir demonstração de capacidade.	Adquirentes de sistemas, produtos e serviços de software, e para fornecedores, desenvolvedores, operadores, mantenedores, gerentes, gerentes de garantia da qualidade e usuários dos produtos de software.	Fornecedores que estejam: especificando requisitos, projetando um modelo, julgando seus próprios produtos, submetendo seus produtos à certificação ou à obtenção de marcas de conformidade. Entidades de certificação, de credenciamento laboratórios de testes, auditores que julgam a competência dos laboratórios de testes. Compradores que pretendam: comparar seus próprios requisitos com os aqui descritos, procurar por produtos certificados, verificar se os requisitos foram atendidos.	atividades de análise foram abruptamente negligenciadas, e, quando levadas em consideração, suas características de qualidade e/ou de conformidade deverão ser analisadas, por meio de reconhecimento de requisitos mínimos e testes.
Organizações Alvo	Organizações que necessitam de uma certificação.	Organizações em geral.	Organizações que desenvolvem, fornecem, certificam requisitos de qualidade de pacotes de software.	Organizações que desenvolvem, fornecem, certificam requisitos de qualidade de pacotes de software. Organizações educacionais, que necessitam de um meio para sugerir seus discentes ao uso de tais métodos.

Aspectos abordados	Normas			SoftCheck
	NBR ISO 9000-3	NBR ISO/IEC 12207	NBR ISO/IEC 12119	
Definições de Processos	Não estabelece processos, estabelece atividades a serem cumpridas, com visão de estrutura, ciclo de vida e suporte.	Estabelece dezessete processos, organizados em três categorias. No entanto, não define um paradigma sustentável ou passível de comparação.	Estabelece oito processos organizados em duas atividades correlatas.	Estabelece oito processos organizados em duas atividades correlatas.
Flexibilidade	Não admite adaptação nos aspectos abordados.	Classificação dos processos pode ser utilizada conforme os objetivos da organização.	Os processos podem ser utilizados conforme os objetivos do usuário.	Não admite adaptação nos aspectos abordados, mas seus processos podem ser utilizados conforme os objetivos do usuário.
Nível de dificuldade de interpretação	A interpretação depende única e exclusivamente do usuário, possuindo, então, grande nível de dificuldade, já que não se verifica subsídios para aplicação das atividades a serem cumpridas.	A interpretação depende única e exclusivamente do usuário, possuindo, grande nível de dificuldade, já que não se verifica subsídios para aplicação dos processos a serem cumpridos.	A interpretação depende única e exclusivamente do usuário, possuindo nível de dificuldade intermediário, já que se verifica subsídios e paradigmas (modelos) para aplicação dos processos a serem cumpridos.	A interpretação não mais depende única e exclusivamente do usuário, todavia, possui nível de dificuldade menor já que se verifica subsídios e paradigmas(modelos) para aplicação dos processos a serem cumpridos. Além disso, a questão da subjetividade é trabalhada de forma objetiva.
Tempo	Grande subjetividade, tempo maior para interpretação do contexto.	Grande subjetividade, tempo maior para interpretação do contexto.	Grande subjetividade, tempo intermediário para interpretação do contexto, devido a existência de exemplo de execução.	Tempo menor para interpretação do contexto, devido a existência de questões objetivas e recomendações, que possibilitam ao usuário internalizar mais rapidamente os benefícios que poderiam ser verificados com o emprego de um determinado item.

Aspectos abordados	Normas			SoftCheck
	NBR ISO 9000-3	NBR ISO/IEC 12207	NBR ISO/IEC 12119	
Objetividade	Grande chance do usuário se contradizer devido à subjetividade.	Grande chance do usuário se contradizer devido à subjetividade.	Chance intermediária do usuário se contradizer devido à subjetividade, pois a presença de exemplos torna a compreensão mais flexível à assimilação.	Chance pequena do usuário se contradizer devido à objetividade das questões de resposta única.
Facilidade de uso	Subjetiva, formato de contrato.	Subjetiva, formato de contrato.	Subjetiva, formato de contrato.	Objetiva, formato lista de checagem.
Condução	Nível de condução ao emprego do item pelo usuário inexistente, isto é, não existe incentivo por parte do sistema, depende das metas do usuário a aplicação ou não dos recursos da norma.	Nível de condução ao emprego do item pelo usuário inexistente, isto é, não existe incentivo por parte do sistema, depende das metas do usuário a aplicação ou não dos recursos da norma.	Nível de condução ao emprego do item pelo usuário inexistente, isto é, não existe incentivo por parte do sistema, depende das metas do usuário a aplicação ou não dos recursos da norma.	Nível de condução ao emprego do item pelo usuário existente, as recomendações fornecidas ao usuário da lista de checagem promovem tal encaminhamento.
Limitações	Numa situação contratual, limita-se às diretrizes para aplicação da série NBR ISO 9000 para o fornecimento, desenvolvimento e manutenção de produtos.	Participa de todo o processo de ciclo de vida de software.	Participa do processo inicial, definição de requisitos e final, representado por testes para averiguações de qualidade e/ou conformidade.	Participa do processo inicial, definição de requisitos e final, representado por testes para averiguações de qualidade e/ou conformidade.
Aspectos positivos	Norma internacional; Difusão extensa; Reconhecimento do valor da certificação.	Norma internacional; Definição de uma taxonomia para processos; útil para qualquer organização.	Norma internacional; Paradigmas para melhorar a internalização dos benefícios da utilização da norma.	Extraído de norma internacional; Formato de questões objetivas; Paradigmas e recomendações para melhorar a internalização dos benefícios da utilização de itens;

4.7 DISCUSSÕES

Na sessão anterior foi apresentado um quadro que reflete de forma geral e resumida os principais aspectos, incluindo vantagens e desvantagens, de algumas normas descritas neste trabalho, traçando um comparativo entre tais normas e o SoftCheck, que aqui serão comentados.

Com relação aos aspectos e abordagem observa-se que, cada norma está voltada para um tipo de aplicação, até mesmo a lista de checagem que foi baseada na norma NBR ISO/IEC 12119, possui características próprias, tais como: estabelecer um *feedback* quando a ação do usuário indicar a não utilização de um determinado item e manter um formato de apresentação objetivo. A NBR ISO/IEC 9000-3, indica apenas como certificar se um processo de uma organização possui um nível de qualidade adequado, não tratando a melhoria contínua dos processos de qualidade, como é o caso da NBR ISO/IEC 12207, que provê tal processo, além de fornecer definição para os processos de todo o ciclo de vida do software.

Todos os modelos atendem a tipos variados de usuários incluindo-se: compradores, fornecedores, desenvolvedores em diversas situações.

De maneira geral, os modelos podem ser utilizados em qualquer tipo de organização, ressalta-se apenas que o SoftCheck também possui como um de seus objetivos atingir veemente as organizações educacionais.

À exceção da NBR ISO 9000-3, que determina apenas atividades a serem cumpridas para a obtenção da certificação, as demais apresentam processos agrupados em categorias ou organizados em atividades correlatas. Ressalva-se que o processo de melhoria contínua é adotado apenas na norma NBR ISO/IEC 12207.

Todos os modelos podem ser adaptados em relação aos aspectos abordados, excluindo-se a NBR ISO 9000-3, que, como visa apenas a certificação, todos os itens deverão estar em conformidade com o produto a ser certificado, e a lista de checagem, que pela sua forma; questões objetivas; não estão abertas à mudanças. No entanto, tal lista pode ser usada parcial ou totalmente de acordo com os objetivos dos usuários, adaptando-se às suas necessidades, dentro dos processos especificados anteriormente.

Com relação aos aspectos: nível de dificuldade de interpretação, tempo, objetividade e facilidade de uso o comportamento dos modelos são sempre os mesmos,

diferem apenas em algumas características que, contemplam ou não exemplos de aplicação, conduzem ou não o usuário e na forma de apresentação. Sob estes aspectos, as normas NBR ISO/IEC 12207 e NBR ISO 9000-3 estão organizadas de modo que a subjetividade mantenha alto o nível de dificuldade de interpretação, diminuindo a objetividade e aumentando o tempo necessário para sua utilização, dificultando sua disseminação. A NBR ISO/IEC 12119 apresenta uma vantagem sobre as duas anteriores, o texto é mais favorável no que diz respeito ao contexto e ainda traz um exemplo de aplicação em um de seus anexos.

No entanto, em tais aspectos o modelo mais vantajoso é a lista de checagem, que ameniza o problema de interpretação quando faz uso de questões objetivas, e conseqüentemente, reduzindo o tempo para a internalização dos benefícios propostos pelo modelo, aumenta a objetividade e o índice de utilização, além de proporcionar ao usuário uma aplicação mais perto possível de uma padronização para softwares, tendo em vista todas as dificuldades encontradas em se definir como deveria ser um software de qualidade.

A NBR ISO/IEC 12119 e a lista de checagem, limitam-se a atuar apenas na especificação dos requisitos e na fase final de testes, enquanto as demais participam de todas as fases de desenvolvimento do software.

O SoftCheck é o único modelo que se preocupa com a condução do usuário, isto é, possui características que visam levar o usuário a empregar determinados itens que normalmente não o faria por desconhecimento de seu benefício ou dos problemas que poderão afetar o sistema. O elemento na lista de checagem que trata da condução são as recomendações, que enfatizam os problemas que poderiam ser causados no sistema pela negligência a determinadas atividades.

Finalmente, nos aspectos positivos pode-se observar o seguinte:

A NBR ISO 9000-3, apesar de ser muito difundida, promovendo o reconhecimento do valor da certificação, no entanto, tende a encaminhar a certificação como objetivo principal da aplicação da Norma.

A NBR ISO/IEC 12207, propicia uma grande classificação e ordenação de processos, no entanto, não se aprofunda em estratégias práticas, nem mesmo menciona exemplos para sua aplicação.

A NBR ISO/IEC 12119, apresenta um benefício importante ao usuário que foi a definição de um modelo/exemplo de aplicação, além de possuir um texto mais objetivo do que as anteriores.

E, o SoftCheck, evidencia além de vários dos aspectos relacionados nas demais normas, envolve características como a objetividade, pela forma de apresentação da lista de verificação e a condução dos usuários por meio das recomendações, agrupando às normas novos recursos .

5 CONCLUSÃO

O estudo realizado teve por objetivo principal sanar os problemas decorrentes de softwares desenvolvidos sem a utilização de métodos ou normas existentes, devido à complexidade de interpretação, adequação ou realização do processo, não acessibilidade e a subjetividade dos métodos existentes.

A viabilização do trabalho iniciou-se com um referencial bibliográfico concentrado no terceiro capítulo abrangendo:

a) Fatores e atividades de qualidade e garantia de qualidade de software, apresentando ainda qualidade do produto versus processo, além de, agregar a esta sessão os meios para a obtenção de certificação;

b) Normas e Modelos, de características gerais e específicas que são exploradas com o intuito de tornar o trabalho mais completo para futuras pesquisas. Geral, ao incluir características que podem ser utilizadas em todas as áreas, produtos e serviços; o contrário das específicas, aqui descritas, como aquelas que são utilizadas exclusivamente para os processos, produtos e serviços de software. Tais normas são desenvolvidas por meio de comissões, formadas por: produtores, consumidores e neutros (universidades, laboratórios e outros). Enquanto os modelos, podem ser desenvolvidos por uma comunidade específica, para um fim específico.

A proposta do trabalho foi idealizada e descrita no quarto capítulo, definido como SOFTCHECK: uma lista de checagem. A lista desenvolveu-se com o objetivo de atender aos problemas de baixos índices de utilização de modelos, técnicas e normas no desenvolvimento (neste momento, “desenvolvimento” são todas as fases do ciclo de vida do software), originando produtos ou prestação de serviços incoerentes com a tecnologia existente.

Fizeram parte da fundamentação da lista:

a) NBR ISO/IEC 12119 – Tecnologia de Informação – Pacotes de Software – Teste e requisitos de qualidade;

b) NBR ISO 8402 – Gestão da qualidade e garantia da qualidade – Terminologia;

A primeira norma é utilizada para a criação do módulo1 do SOFTCHECK e parte do módulo 2, que por meio do anexo A (informativo), também define alguns dos termos usados na lista de verificação, originando um glossário.

A segunda norma - NBR ISO 8402, complementa o glossário com outros termos além daqueles necessários para o estágio atual do SOFTCHECK, indicando que a ferramenta está preparada para que outras normas sejam integradas a ela.

Concluindo este estudo, tem-se o quinto capítulo, onde se verifica uma explanação sobre o objetivo do trabalho, o que foi realizado em termos de contribuições e sugestões para estudos futuros, e o sexto capítulo finaliza o trabalho, citando as bibliografias e bibliografias referenciadas, utilizadas para o desenvolvimento do projeto.

5.1 CONTRIBUIÇÕES

Este trabalho, contou com um vasto referencial quantitativo para trabalhos futuros ao descrever boa parte das normas e modelos relacionados ou não ao software, apesar de limitar-se à definições não exaustivas. Foram descritas ao todo, 12 (doze) normas e modelos, das quais 6 (seis) são modelos, divididos em 3 (três) modelos de características gerais (TQM, PDCA e GQM) e 3 (três) específicos para os processos ou produtos de software (CMM, BOOTSTRAP e PSP). Constam também 6 (seis) normas, no qual apenas uma generalista, a série NBR ISO 9000, e 6 (seis) específicas (NBR ISO/IEC 9126, ISO/IEC14598, NBR ISO/IEC 12119, NBR ISO/IEC 12207, ISO/IEC TR 15504(SPICE) e NBR ISO 9000-3).

No entanto, a maior contribuição se resume em uma lista de checagem desenvolvida como descrita nas sessões 2.1 e 2.2 do segundo capítulo. Estruturalmente, a lista foi baseada na norma técnica NBR ISO/IEC 12119, preestabelecendo atividades de levantamento de requisitos e finais do ciclo de vida de software, no que diz respeito a embalagens com informações relacionadas aos recursos necessários para a instalação do produto, testes, documentações, entre outros.

Além disso, devido à preocupação com a disseminação do conhecimento entre profissionais novatos ou não, fazem parte da lista de checagem recomendações de uso para respostas que foram assinaladas como negativas (não), no momento da utilização da lista. O SoftCheck é o único modelo que se preocupa com a condução do usuário, isto é, possui características que visam levar o usuário a empregar determinados itens

que normalmente não o faria por desconhecimento de seu benefício ou dos problemas que porventura poderão afetar o sistema.

Ainda como contribuição, se tem uma notação própria, construída para abster da lista de checagem palavras-chave que seriam escritas copiosamente. A notação está preparada para atender a outras normas, possuindo, por exemplo, recomendações para respostas não aplicáveis e derivações, que neste documento não serão utilizadas, pois as questões do módulo 1, não se restringem a tipos de produtos. Na realidade, qualquer produto pode ser verificado por meio dos itens contidos no SOFTCHECK módulo 1.

A lista de checagem possui também como uma de suas características mais marcantes a estrutura objetiva, contribuindo para a interpretação das fontes, de maneira adequada, isto é, sem dar margens à ambigüidades de interpretação entre dois usuários diferentes, diminuindo o tempo necessário para sua implantação ou utilização, enfim, reduzindo os problemas causados pela subjetividade das normas até então verificados.

Concluindo, verifica-se claramente que os objetivos da lista de checagem agem concisa e firmemente auxiliando os problemas decorrentes de processos ou atividades de levantamento de requisitos e testes, que buscam softwares de qualidade por meio da utilização de normas, concretizando um objetivo maior: a utilização da lista por parte de profissionais e novatos, difundindo assim, por meio deste trabalho, o conhecimento e o emprego de tais técnicas.

5.2 SUGESTÃO PARA TRABALHOS FUTUROS

A organização e concepção realizada neste estudo permitem que seja sugerido o desenvolvimento de trabalhos futuros associados ao tema, abordando os seguintes tópicos:

1) Desenvolver outros módulos, agrupando maior número de normas, proporcionando assim, um rol de escolha para o usuário, que poderá experimentar um método, modelo ou norma e utilizar a que melhor se adaptar.

2) Dar continuidade em estudos teóricos com relação à terminologia utilizada nas normas e aos conceitos vigentes, pois, sem eles não é possível a objetividade das recomendações. Motivo pelo qual impossibilitou o desenvolvimento do módulo referente a norma NBR ISO/IEC 12207 – Tecnologia de informação – Processos de Ciclo de Vida de Software, em tempo hábil para este trabalho;

- 3) Atualização das questões aqui elaboradas, caso haja no futuro, versão da norma mais recente ou outra que venha substituí-la;
- 4) Implementação da ferramenta como serviço via Internet, disponibilizando-a para todo e qualquer tipo de usuário. Além disso, colocar também à disposição do usuário contato via e-mail, ou outro meio, para que ele possa informar possíveis inconsistências encontradas ou participar novas idéias;
- 5) Agrupamento das questões aqui sugeridas por tipo de usuário a utilizá-las;
- 6) Quantificação das questões, quanto às decisões a serem tomadas dependente da quantidade de perguntas com respostas iguais. Por exemplo, até uma quantidade “x” de questões não empregadas é considerado normal e anormal acima de “x”; ou outro sistema de quantificação;
- 7) Além de quantificar, qualificar o usuário. Tal tarefa deve ser executada por um representante mais graduado e de maior experiência, que classificará os demais usuários de acordo com suas capacidades de discernimento, “graduando-os” com respectivos pesos dentro de suas avaliações.

BIBLIOGRAFIA

- [BAR97] Barreto, J. Jr. <http://www.barreto.com.br/qualidade>: 1997.
- [CAR97] Carvalho, M. B. et. al. Certificação x Melhoria. IV Workshop da Qualidade de Software, Fortaleza: 1997.
- [COL99] Colanzi, T. E. et. al. Teste de Software OO: Uma abordagem para a organização e o Planejamento da Atividade de Teste. VI Workshop da Qualidade de Software, Florianópolis-SC: 1999.
- [CYB99] Cybis, Walter A.; Pereira, Danilo; Silva, Sâmela S. S. Avaliação de Software hipermídia pela ferramenta “Ergolist”: Uma estratégia válida. ABERGO'99. Salvador: 1999.
- [DIA96] Diaz, Arnaldo. Un Estándar Internacional para la Evaluación del Proceso de Desarrollo de Software ISO/SPICE. Revista Soluciones Avanzadas, nº 31: 1996.
- [FIO94] Fiorini, S. T.; Leite, C. J. Qualidade na Definição de Requisitos. I Workshop da Qualidade de Software, Curitiba-PR: 1994.
- [FRI94] Frizanco, O. et. al. Software com Garantia de Qualidade em Parceria Cliente e Fornecedor. I Workshop da Qualidade de Software, Curitiba: 1994.
- [GQM--] GQM – Goal Question Metrics. <http://jacques.ic.cti.br/ic/pqps/fag-01.htm>.
- [HAZ99] Hazan, Claudia. Introdução da Gerência pela Qualidade Total em Organizações de Desenvolvimento de Software. VI Workshop da Qualidade de Software, Florianópolis: 1999.
- [IAV97] Iavarone, G. M. e Sanches, R. Um instrumento de coleta de dados para verificar a influência do processo de desenvolvimento da interface na usabilidade do produto. IV Workshop da Qualidade de Software, Fortaleza: 1997.
- [LAR92] Cultural, Larousse. Dicionário da Língua Portuguesa. Editora Nova Cultura: 1992.
- [LEM97] Lemes, M. J. R.; Fernandes C. T. Uma Taxionomia para métricas de Software. IV Workshop da Qualidade de Software, Fortaleza: 1997.

- [MAC97] Machado, C. A. F. et. al. Procesos de Ciclo de Vida de Software. I Simpósio Latino Americano de Calidad e Productividad en Desarrollo de Software, Santiago, Chile: 1997.
- [NBR93] NBR ISO 9000-3. Normas de gestão da qualidade e garantia da qualidade. Parte 3: Diretrizes para a aplicação da NBR 9001 ao desenvolvimento, fornecimento e manutenção de "software". ABNT – Associação Brasileira de Normas Técnicas: 1993.
- [NBR94a] NBR ISO 8402. Gestão da qualidade e garantia da qualidade – Terminologia. ABNT – Associação Brasileira de Normas Técnicas: 1994.
- [NBR94b] NBR ISO 9001. Sistemas da qualidade – Modelo para garantia da qualidade em projeto, desenvolvimento, produção, instalação e serviços associados. ABNT – Associação Brasileira de Normas Técnicas: 1994.
- [NBR96] NBR 13596. Tecnologia de informação – Avaliação de produto de software – Características de qualidade e diretrizes para seu uso. ABNT – Associação Brasileira de Normas Técnicas: 1996.
- [NBR98a] NBR ISO/IEC 12119. Tecnologia de informação – Pacotes de software – Teste e requisitos de qualidade. ABNT – Associação Brasileira de Normas Técnicas: 1998.
- [NBR98b] NBR ISO/IEC 12207. Tecnologia de informação – Processos de ciclo de vida de software. ABNT – Associação Brasileira de Normas Técnicas: 1998.
- [PAU95] Paulk, M. C. How ISO 9001 compares with the CMM. IEEE Software, p. 15-24: 1990.
- [PDC94] PDCA Model. The Memory Jogger II: A Pocket Guide of Tools for Continuous Improvement and Effective Planning. 1994.
- [PRE95] Pressman, R. S. Engenharia de Software. Makron Books, São Paulo: 1995.
- [PSP99] Personal Software Process. <http://www.sei.cmu/psp/whatPSP.htm>, Carnegie Mellon University: 1999.
- [RUB96] Rubio, S. E. D. Bootstrap el Estándar Europeo para Evaluación y Mejoras de Procesos de Desarrollo de Software. Revista Soluciones Avanzadas, n° 38: 1996.

- [SOM96] Sommerville, Ian. Software Engineering. Fifth edition, Addison-Wesley: 1996.
- [TQM96] TQM – Total Quality Management. <http://www.qualidade.com/faq-01.htm#tqm>: 1996.
- [TSU97] Tsukumo, A. N. et. al. Qualidade de software: visões de produto e processo de software. II Escola Regional de Informática da Sociedade Brasileira de Computação Regional de São Paulo, Piracicaba, p. 173-189: 1997.
- [WEB94] Weber, K. C. et. al. Dois Modelos para a Gestão da Qualidade. I Workshop da Qualidade de Software, Curitiba-PR: 1994.
- [WEB97] Weber, K. C. Medições em Empresas de Software. IV Workshop da Qualidade de Software, Fortaleza: 1997.
- [WEB99] Weber, K. C. et. al. Qualidade e Produtividade em Software. Makron Books, São Paulo: 1999.
- [WEB99b] Weber, K. C. et. al. Evolução para a Maturidade com Base na ISO 9000. VI Workshop da Qualidade de Software, Florianópolis-SC: 1999.
- [YOU90] Yourdon, Edward. Análise Estruturada Moderna. 3ª ed., Ed. Campus: 1990.

BIBLIOGRAFIA REFERENCIADA

- [ABE95] M., Abel; Price, R. T. Programa da Qualidade Total na Informática. II Escola Regional de Informática – SBC-RS, Caxias do Sul: 1995.
- [ALB79] Albrecht, A. J. Measuring application development productivity. In Proc. SHRE/GUIDE IBM Application Development Symposium, 83-92: 1979.
- [ALB83] Albrecht, A. J and Gaffney, J. E. Software function, lines of code and developemtn effort prediction: a software science validation. IEEE Trans. on Software Engineering, SE-9 (6), 639-47: 1983.
- [ARO74] Aron, J. D. The Program Development Process. Reading MA: Addison-Wesley: 1974.
- [BAK72] Baker, F. T. Chief programmer team management of production programming. IBM Systems J., 11 (1), 56-73: 1972.
- [BAS63] Bass, B. M. and Dunteman, G. Behaviour in groups as a function of self, interations and task orientation. J. Adnorm. Soc. Psychology, 66 (4), 19-28: 1963.
- [BAS78] Basili, V. and Zelkowitz, M. Analysing Medium Scale Software Development. Proc 3rd Intl. Conf. Software Engineering, IEEE, pp. 116-123: 1978.
- [BAS90] Basili, V. R. Viewing Maintenance as Reuse – Oriented Software Development. IEEE Software: 1990.
- [BAS96] Basili, V. R. Experimentation in Sofwtare Engineering. IEEE Transctions on Software Engineering, Vol. SE-12, n° 7: 1986.
- [BRO75] Brooks, F. P. The Mythical Man Month. Reading MA: Addison-Wesley [46, 583]: 1975.
- [CAV78] Cavano, J. P.; MacCall, J. A. A Framework for the Measurement of Software Quality. Proc. ACM Software Quality Assurance Workshop, p. 133-139: 1978.
- [CIO91] Cioch, F. A. Measuring Software Misinterpretation. The Journal of Systems and Software, n° 14, p. 85-95: 1991.

- [COA94] Coallier, F. How ISO 9001 fits into of software world. IEEE Software: 1994.
- [COU78] Cougar, J. D. and Zawacki, R. A. What motivates DP professionals. Datamation, 24 (9), 27-30 [578]: 1978.
- [COU88] Cougar, J. D. Motivation IS personnel. Datamation, 34 (9), 59-63 [578]: 1988.
- [CRO79] Crosby, P. Quality is Free. Mac Graw-Hill: 1979.
- [DeM85] DeMarco, T. and Lister, T. Programmer performance and the effects of the workplace. In Proc. 8th Int. Conf. on Software Engineering, London, 268-72 [584]: 1985.
- [FRE90] Freedman, D. P.; Weinberg, G. M. Handbook of Walkthroughs, Inspections and Technical Reviews. 3^a ed., Dorset House: 1990.
- [GUN62] Gunning, R. Techniques of Clear Writing. New York: McGraw-Hill [624,634]: 1962.
- [HIR82] Hirvensalo, T.; Sepponen R. Quality Control and Verification of Software. Reliability in Electrical and Eletronic Components and Systems: 1982.
- [HUM95] Humphrey, Watts S. A discipline for Software Engineering. Maryland: 1995.
- [IEE94] Software Engineering Standards Collection. Los Alamitos, CA: IEEE Press [619]: 1994.
- [JON91] Jones, T. C. Programing Productivity Issues for the 80s. IEEE Computer Society Press, p.13-20: 1981.
- [KRI96] Kristen, G. Total Quality Management with Object Orientation. Magazine: Business Objects Object Magazine: 1996.
- [MAR75] Marshall, J. E. and Heslin, R. Boys and girls together. Sexual composition and the effect of density on group size and cohesiveness. J. Personality and Social Psychology, 31 (5), 952-61 [582]: 1975.
- [MAS54] Maslow, A. A. Motivation and Personality. New York: Harper and Row [572]: 1954.
- [MCC77] MacCall, J. A. et. al. Factors in Software Quality. Très columes, NTIS AD-AO49-014, 015, 055: 1977.
- [McC78] IBM's Santa Teresa laboratory: architeturual design for program development. IBM Systems J., 17 (1), 4-25 [585]: 1978.

- [MIL93] Milet, E. et. al. Os Principios da Qualidade Total Aplicados à Informática. Ed. LTC: 1993.
- [MOH79] Mohanty, S. N. Models and Measurement for Quality Assessment of Software. ACM Computing Surveys, v.11, nº 3, p. 251-275: 1979.
- [ROM87] Rombach, H. D. A controlled Experiment on the Impact of Software Structured on Maintainability. IEEE Transactions on Software Engineering, Vol. SE-13, nº 03: 1987.
- [SAC68] Sackman, H. et. al. Exploratory experimentation studies comparing on-line and off-line programming performance. Comm. ACM, 11 (1), 3-11 [594]: 1968.
- [SCH87] Schulmeyer, G. C.; McManus, J. I. Handbook of Software Quality Assurance. Van Nostrand Reinhold: 1987.
- [SYM88] Symons, C. R. Function point analysis: difficulties and improvements. IEEE Trans. on Software Engineering, 14 (1), 2-11 [594]: 1988.
- [WAL77] Walston, C. E. and Felix, C. P. A method of programming measurement and estimation. IBM Systems J., 16 (1), 54-73 [583]: 1977.
- [WEI71] Weinberg, G. The Psychology of Computer Programming. New York: Van Nostrand [580]: 1971.