

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO**  
(Bacharelado)

**PROTÓTIPO DE APOIO AO PROCESSO DE VERIFICAÇÃO**  
**BASEADO NA NORMA ISO/IEC 12207**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE  
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA  
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA  
COMPUTAÇÃO — BACHARELADO

**ITABORAÍ CORDONI EBERTZ**

BLUMENAU, NOVEMBRO/2002

2002/2-33

# **SOFTWARE DE APOIO AO PROCESSO DE VERIFICAÇÃO BASEADO NA NORMA ISO/IEC 12207**

**ITABORAÍ CORDONI EBERTZ**

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO  
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE  
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

**BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO**

---

Prof. Everaldo Artur Grahl — Orientador na FURB

---

Prof. José Roque Voltolini da Silva — Coordenador do TCC

## **BANCA EXAMINADORA**

---

Prof. Everaldo Artur Grahl

---

Prof. Marcel Hugo

---

Prof. Ricardo A. Azambuja

## **DEDICATÓRIA**

Dedico este trabalho aos meus pais Valentin e Yara por toda ajuda, incentivo, compreensão e carinho que tiveram comigo durante mais esta etapa da minha vida.

## **AGRADECIMENTO**

Agradeço a Deus, por ter me dado forças em mais um momento decisivo da minha vida.

Agradeço especialmente ao Professor Everaldo Artur Grahl, por toda sua dedicação e orientação durante a confecção deste trabalho.

Aos professores do curso que me repassaram da melhor maneira possível todos seus conhecimentos e orientações.

A Adriana, pelo amor e incentivo nas horas difíceis deste trabalho e também da minha vida.

Volto a agradecer aos meus pais pelo incentivo e apoio que sempre obtive e com certeza sempre o terei.

# SUMÁRIO

LISTA DE FIGURAS E TABELAS .....	VII
RESUMO .....	VIII
ABSTRACT .....	IX
1 INTRODUÇÃO .....	1
1.1 OBJETIVOS .....	3
1.2 ESTRUTURA DO TRABALHO .....	3
2 VERIFICAÇÃO DE SOFTWARE .....	4
2.1 CONCEITOS .....	4
2.2 ROTEIRO DE VERIFICAÇÃO DE SOFTWARE .....	6
2.3 ATIVIDADES RELATIVAS À VERIFICAÇÃO .....	9
2.3.1 VERIFICAÇÃO DO CONTRATO .....	9
2.3.2 VERIFICAÇÃO DO PROCESSO .....	11
2.3.3 VERIFICAÇÃO DOS REQUISITOS .....	11
2.3.4 VERIFICAÇÃO DO PROJETO .....	13
2.3.5 VERIFICAÇÃO DO CÓDIGO .....	13
2.3.6 VERIFICAÇÃO DA INTEGRAÇÃO .....	14
2.3.7 VERIFICAÇÃO DA DOCUMENTAÇÃO .....	15
3 NORMAS E MODELOS DE QUALIDADE .....	17
3.1 NORMA ISO/IEC 15504 .....	17
3.2 PROCESSO DE VERIFICAÇÃO SEGUNDO A NORMA ISO/IEC 15504 .....	17
3.3 NORMA ISO 9000-3 .....	18
3.4 PROCESSO DE VERIFICAÇÃO SEGUNDO A NORMA ISO 9000-3 .....	19
3.5 MODELO CMMI .....	22

3.6 PROCESSO DE VERIFICAÇÃO SEGUNDO O MODELO CMMI .....	23
3.7 QUADRO COMPARATIVO.....	25
4 ESPECIFICAÇÃO DO PROTÓTIPO.....	29
4.1 REQUISITOS DO PROBLEMA .....	29
4.2 LISTA DE EVENTOS .....	29
4.3 DIAGRAMA DE CONTEXTO .....	30
4.4 DIAGRAMA ENTIDADE-RELACIONAMENTO .....	31
4.5 DIAGRAMA DE FLUXO DE DADOS .....	34
4.6 DICIONÁRIO DE DADOS .....	35
5 IMPLEMENTAÇÃO DO PROTÓTIPO .....	38
5.1 OPERACIONALIDADE DA IMPLEMENTAÇÃO.....	38
6 CONCLUSÕES .....	48
REFERÊNCIAS BIBLIOGRÁFICAS .....	50
ANEXO - <i>CHECKLIST</i> .....	53

## LISTA DE FIGURAS E TABELAS

FIGURA 1 – ROTEIRO DE VERIFICAÇÃO DE SOFTWARE.....	7
FIGURA 2 – DIAGRAMA DE CONTEXTO .....	31
FIGURA 3 – DIAGRAMA DE ENTIDADE-RELACIONAMENTO – NÍVEL LÓGICO ...	32
FIGURA 4– DIAGRAMA DE ENTIDADE-RELACIONAMENTO – NÍVEL FÍSICO .....	33
FIGURA 5 – DIAGRAMA DE FLUXO DE DADOS DE 1 A 5 .....	34
FIGURA 6 – DIAGRAMA DE FLUXO DE DADOS DE 6 A 9 .....	35
FIGURA 7 – DICIONÁRIO DE DADOS .....	36
FIGURA 8 – TELA INICIAL DO SOFTWARE DE APOIO AO PROCESSO DE VERIFICAÇÃO DE SOFTWARE .....	38
FIGURA 9 – TELA DE CADASTRO DA ORGANIZAÇÃO .....	39
FIGURA 10 – TELA DE CADASTRO DO AVALIADOR.....	39
FIGURA 11 – TELA DE CADASTRO DAS PERGUNTAS DO CONTRATO .....	40
FIGURA 12 – TELA DE REGISTRO DO PLANO DE VERIFICAÇÃO.....	41
FIGURA 13 – TELA DE SELEÇÃO DAS PERGUNTAS DA ATIVIDADE DO CONTRATO .....	42
FIGURA 14 – CÓDIGO FONTE DA IMPLEMENTAÇÃO .....	43
FIGURA 15 – TELA DA CONSULTA DAS PERGUNTAS DA ATIVIDADE DO CONTRATO .....	44
FIGURA 16 – TELA DA VERIFICAÇÃO DO CONTRATO.....	45
FIGURA 17 – RELATÓRIO DAS PERGUNTAS DA ATIVIDADE DO CONTRATO.....	46
FIGURA 18 – RELATÓRIO DAS RESPOSTAS ATENDIDAS DO CONTRATO.....	47
TABELA 1 – COMPARATIVO DO PROCESSO DE VERIFICAÇÃO DE SOFTWARE.	26
TABELA 2 – QUANTIDADE DE PERGUNTAS .....	40

## **RESUMO**

Este trabalho descreve conceitos de verificação de software e apresenta um estudo comparativo entre o processo de verificação de software existente nas normas de qualidade ISO/IEC 12207, ISO 9000-3, ISO/IEC 15504 (SPICE) e no modelo CMMI. Propõe-se um processo de verificação de software baseado na norma ISO/IEC 12207, bem como o desenvolvimento de um software para o apoio a este processo.



## **ABSTRACT**

This work describes concepts of software verification and presents a comparative study between existing software verification process in ISO/IEC 12207, ISO 9000-3 and ISO/IEC 15504 (SPICE) quality norms and the CMMI model. It proposes software verification process based in ISO/IEC 12207, as well as it developed software to the support this process.

# 1 INTRODUÇÃO

Nos dias atuais a melhoria da qualidade do software torna-se um processo cada vez mais comum nas organizações devido a necessidade de obtenção de melhores resultados em todas as fases do ciclo de vida de software. É com o objetivo de obter esta melhoria da qualidade que surgiram normas e modelos como ISO/IEC 12207, ISO 9000-3, CMMI e ISO/IEC 15504. Cada vez mais as empresas investem na diferenciação de seu produto no mercado. Isto faz com que o termo qualidade seja usado diariamente, tanto em propagandas quanto em conversas informais. Por este motivo, a qualidade é definida como a totalidade de requisitos e características de um produto ou serviço que estabelecem a sua capacidade de satisfazer necessidades implícitas e explícitas (Bizzoto, 1992).

A qualidade de software é determinada pela qualidade dos processos utilizados para o desenvolvimento. Deste modo, a melhoria da qualidade de software é obtida pela melhoria da qualidade dos processos. Esta visão orienta a elaboração de modelos de definição, avaliação e melhoria de processos de software (Iahn, 1999).

Segundo Junckes (1999), a comunidade de informática vem criando normas e modelos de qualidade como ISO/IEC 12207, ISO 9000-3, CMMI e ISO/IEC 15504 para regular e orientar a atividade de produção de software. A norma ISO/IEC 12207 (Processos de Ciclo de Vida de Software) tem por objetivo principal estabelecer uma estrutura comum para os processos de ciclo de vida do software. A norma ISO 9000-3 aborda basicamente situações em que um software específico é desenvolvido como parte de um contrato, atendendo as especificações do comprador. O Modelo de Capacidade e Maturidade de Software Integrado (CMMI) é uma estrutura que descreve os elementos de um processo eficiente de software e um caminho evolucionário que aumenta a maturidade dos processos nas organizações de software. Já a futura norma ISO/IEC 15504, também conhecida como SPICE, constitui-se de um padrão para avaliação do processo de software, visando determinar a capacidade de uma organização.

A estrutura descrita na norma ISO/IEC 12207 utiliza-se de uma terminologia bem definida e é composta de processos, atividades e tarefas a serem aplicadas em operações que envolvam, de alguma forma o software, seja através de aquisição, fornecimento, desenvolvimento, operação ou manutenção (ABNT, 1998). Entre esses processos encontra-se o processo de verificação. O processo de verificação define as atividades para verificação dos

produtos de software. É um processo para determinar se os produtos de software de uma atividade atendem completamente aos requisitos ou condições impostas a eles.

Segundo Rocha (2001) os procedimentos de verificação não são mutuamente exclusivos, mas sim complementares. A verificação pode ser aplicada no início e ao longo de todo o ciclo de vida do processo de desenvolvimento. Na essência, as técnicas de verificação têm por objetivo identificar a presença de defeitos ou erros início do processo de desenvolvimento e de certa maneira aprender com essa atividade e evoluir o desenvolvimento, inclusive pela própria evolução das técnicas de verificação utilizadas na empresa.

Conforme Rocha (2001), um ponto relevante para a condução efetiva das atividades de verificação é o estabelecimento de modelos de erros que captem os erros típicos, os riscos associados, a frequência de ocorrência e outros aspectos, de maneira que a experiência e o conhecimento de desenvolvimento de software de uma equipe ou empresa sejam utilizados para o planejamento das futuras atividades de desenvolvimento. Se essa perspectiva não for armazenada em uma base histórica e analisada posteriormente para propiciar a evolução do processo de desenvolvimento, a equipe ou empresa estará fadada à ineficiência e à repetição de erros prévios.

De acordo com Mills (1994), verificação da qualidade é a avaliação contínua da situação de procedimentos, métodos, condições, produtos, processos e serviços, e a análise dos registros para assegurar que os requisitos da qualidade serão satisfeitos.

Freedman (1993) cita que a verificação contribui significativamente, para redução do número de defeitos contidos em programas. Como tende a ser muito demorada e cara a correção de programas contendo defeitos decorrentes de erros de especificação, arquitetura ou projeto, a verificação contribui ainda, para a aceleração do desenvolvimento e para a redução do custo total do desenvolvimento do programa. Durante a verificação, o artefato que é um subproduto do desenvolvimento deve ser examinado segundo uma lista de quesitos (*check list*). Esta lista depende do tipo de artefato, da linguagem utilizada, dos padrões adotados, etc. Depende pois, da organização responsável pelo desenvolvimento dos programas.

## 1.1 OBJETIVOS

O objetivo deste trabalho é especificar e implementar um software que auxilie o processo de verificação baseado na norma ISO/ IEC 12207.

Os objetivos específicos do trabalho são:

- a) comparar as atividades de verificação previstas em normas conhecidas como ISO 9000-3, ISO/IEC 15504 e modelo CMMI, visando ampliar as atividades de verificação previstas na norma ISO/IEC 12207;
- b) criar instrumentos para facilitar a adoção do processo de verificação como a utilização de *checklists*.

## 1.2 ESTRUTURA DO TRABALHO

A seguir é apresentada uma síntese dos próximos capítulos constantes desse trabalho.

O segundo capítulo apresenta conceitos estudados do processo de verificação da norma ISO/IEC 12207 e um roteiro mostrando a especificação do processo e suas atividades.

O terceiro capítulo apresenta as três normas e o modelo de qualidade estudados, demonstrando detalhadamente o processo de verificação de software de cada alternativa e mostra também um comparativo entre as mesmas.

O quarto capítulo trata da especificação do software de apoio a verificação de software, onde constam a lista de eventos, diagrama de contexto, diagrama de fluxo de dados, e o diagrama de entidade e relacionamento (DER).

O quinto capítulo apresenta a implementação do software de apoio a verificação de software, apresentando as principais telas e relatórios.

Por fim mostra-se no sexto capítulo a conclusão com principais resultados do trabalho e as sugestões para possíveis melhoramentos.

## 2 VERIFICAÇÃO DE SOFTWARE

Este capítulo oferece uma visão geral sobre o processo de verificação de software, onde serão enfocados assuntos como conceitos, roteiro e atividades.

### 2.1 CONCEITOS

O processo de verificação de software tem o objetivo de detectar erros o mais breve possível no ciclo de vida de software, que constitui uma tentativa de introduzir a verificação em todo o desenvolvimento do software. Conforme citado por Sommerville (2001), verificações de software são representações de análise e checagem de sistemas semelhantes aos documentos requeridos, projetos de diagramas e o código fonte de programas. Pode ser aplicado em todo os estágios do processo. Enquanto uma nova organização ganha experiência no processo de verificação, pode usar resultados daquele processo como um significado de um processo de melhoramento. Sempre que possível o processo poderia ser modificado para eliminar as razões de defeitos. Assim eles não iriam mais ocorrer novamente em sistemas futuros.

Segundo Rocha (2001), os produtos de software consistem em um conjunto de informações em diferentes níveis de abstração e num conjunto de transformações e decisões associadas a essas transformações. No processo de comunicação e transformação da informação ocorrem enganos, interpretações errôneas e outras faltas, referenciadas neste texto como defeitos ou erros que podem ocasionar o mau funcionamento do sistema, ou seja, a ocorrência de falhas. Salienta-se que os defeitos podem ser introduzidos ao longo de todo o processo de desenvolvimento de software e é necessário que eles sejam determinados o quanto antes, de preferência na própria fase em que são cometidos, pois os custos associados são maiores quando os defeitos são identificados em fases posteriores.

Com o uso crescente de software nos mais diversos setores da sociedade, a demanda e a preocupação com a produção de software de alta qualidade a baixo custo passou a ser um dos motivos que culminou na introdução de atividades agregadas sob o nome de garantia de qualidade de software ao longo de todo o processo de desenvolvimento de software.

O objetivo da verificação é assegurar que o software, ou uma determinada função do mesmo esteja sendo implementado corretamente. Permitindo, inclusive, se os métodos e processos de desenvolvimento foram adequadamente aplicados. Em geral são conduzidas com

base em um *checklist* (lista de verificação ou de conferência) adequada a cada fase ou produto. A própria atividade de revisão é passível de auditoria para certificação do processo e de evolução com base na experiência da equipe (Rocha, 2001).

Uma outra definição é emitida por Pressman (1995), a verificação refere-se ao conjunto de atividades que garante que o software implemente corretamente uma função específica. É importante notar que a verificação abrange um conjunto amplo de atividades de que incluem revisões técnicas formais, auditorias de configuração e qualidade, monitoração do desempenho, simulação, estudo da viabilidade, revisão da documentação, revisão de bancos de dados, análise de algoritmos, teste de desenvolvimento, teste de qualificação e teste de instalação. Não obstante a atividade de teste desempenhe um papel extremamente importante na verificação, muitas outras atividades também são necessárias.

Uma definição simples e ao mesmo tempo completa é emitida por Inthurn (2001), verificação refere-se ao conjunto de atividades que garante que o software implementa corretamente uma função específica. Na verificação deve-se utilizar a seguinte pergunta: “Estamos construindo certo o produto?”. As verificações podem ser aplicadas no início e ao longo de todo o ciclo de vida do processo de desenvolvimento. A validação refere-se ao conjunto de atividades que garante o software que foi construído é rastreável às exigências do cliente. Na validação deve-se utilizar a seguinte pergunta: “Estamos construindo o produto certo?”. Assim cabe ressaltar que verificação e validação de software são conceitos diferentes onde os de verificação podem ser aplicado no início e ao longo de todo o ciclo de vida do processo de desenvolvimento, enquanto que os de validação tendem ao ocorrer de uma forma concentrada e intensiva nas faces que abrange os testes modulares e de integração.

Já para Hetzel (1987), a verificação é uma avaliação realizada no final de uma fase, com o objetivo de confirmar se as necessidades estabelecidas durante a fase anterior foram atendidas. Em termos gerais, a verificação diz respeito à atividade global de avaliação de software, englobando a revisão, inspeção, teste, análise de desempenho e auditoria.

Uma preocupação da verificação é assegurar que a representação está em conformidade com todos os padrões selecionados. Padrões estabelecem restrições de uso das linguagens de representação. Procura-se através do uso de padrões reduzir a probabilidade do cometimento de erros. Outro objetivo dos padrões é atuar sobre o formato, ou estilo, de apresentação da representação, assegurando, assim, completeza, legibilidade e uniformidade

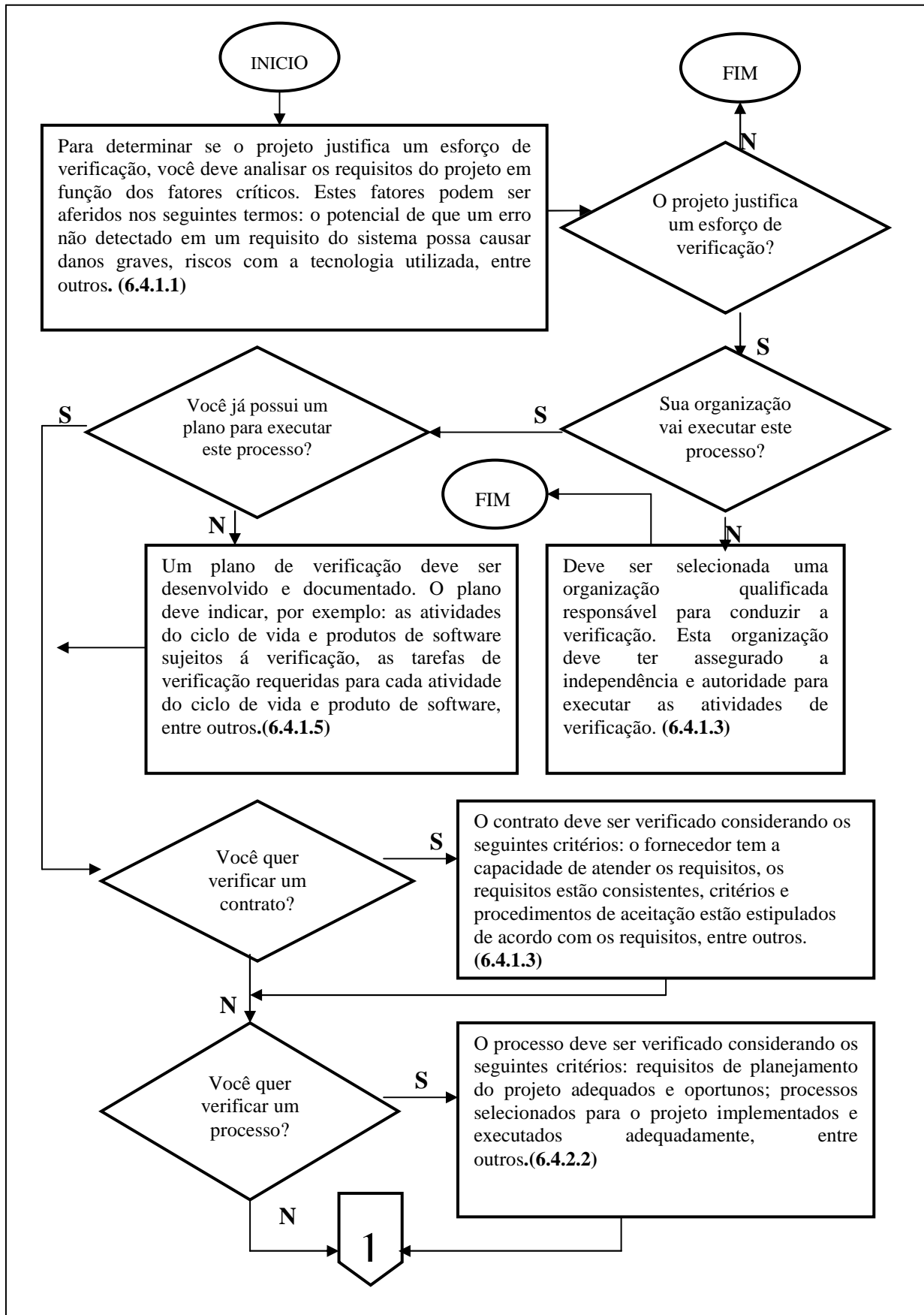
dos documentos. Estas são características muito relevantes ao desenvolver artefatos que deverão ser lidos e mantidos por pessoas diferentes das que desenvolveram o artefato. Deseja-se, em particular, eliminar aspectos personalistas dos programas (Staa, 2000).

O processo de planejamento da verificação deve decidir um padrão e procedimento para a verificação de software. Deve-se estabelecer um *checklist* para guiar a verificação e definir um plano de verificação. O esforço relativo devotado à verificação depende do tipo do sistema a ser desenvolvido e da técnica organizacional.

## **2.2 ROTEIRO DE VERIFICAÇÃO DE SOFTWARE**

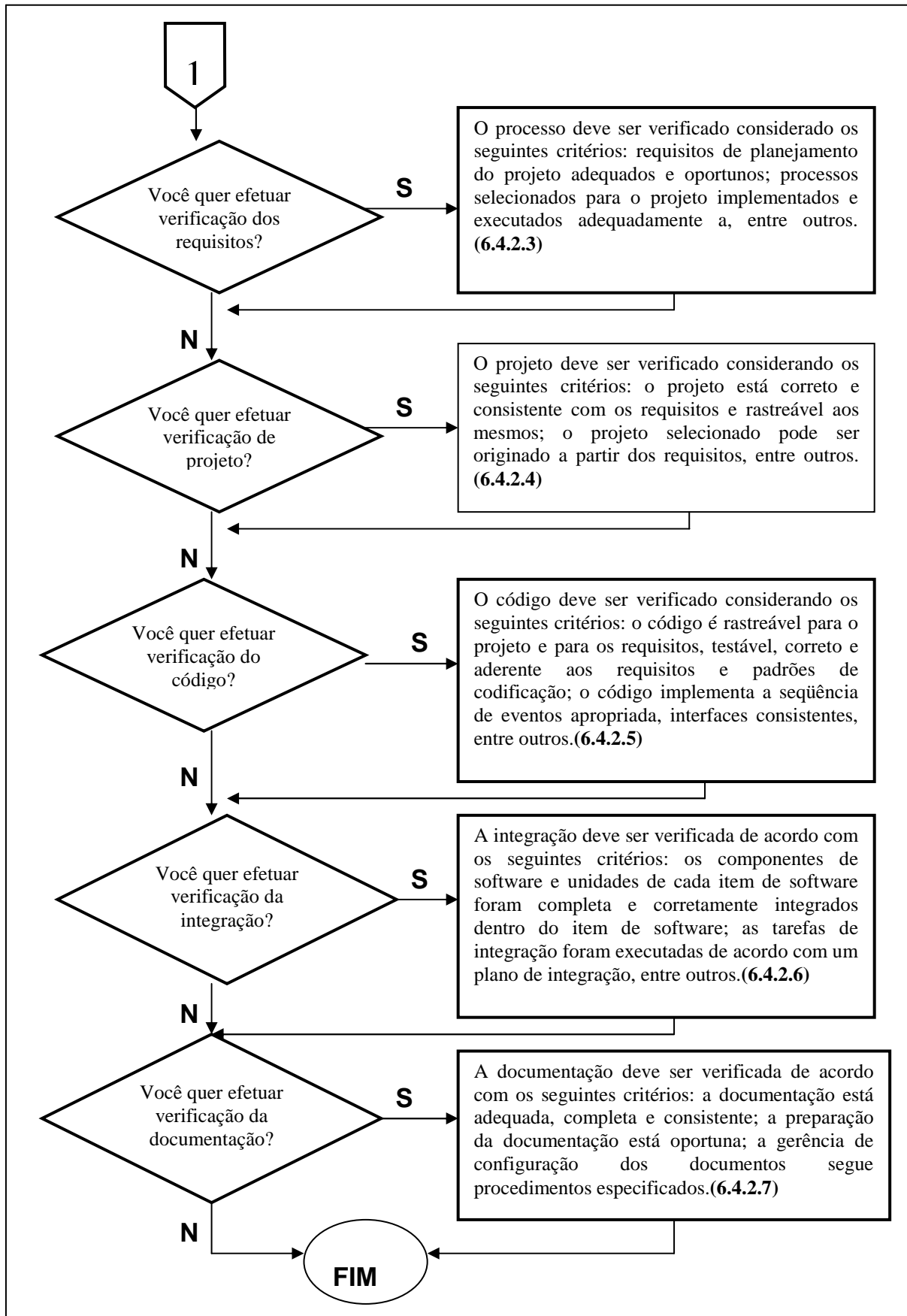
A seguir é demonstrado um roteiro proposto para a verificação de software segundo a norma ISO/IEC 12207, que é demonstrada no trabalho de Frare (1998). Este roteiro será adotado para fundamentar o software de apoio. O roteiro facilita a interpretação da adoção do processo de verificação definido na norma ISO/IEC 12207. A figura 1 utiliza uma representação baseada em fluxograma bastante difundida e acessível para interpretação. As atividades citadas são referenciadas pelo número original na norma.

**FIGURA 1 – ROTEIRO DE VERIFICAÇÃO DE SOFTWARE**





**FIGURA 1 - ROTEIRO DE VERIFICAÇÃO DE SOFTWARE (CONTINUAÇÃO)**



## 2.3 ATIVIDADES RELATIVAS À VERIFICAÇÃO

O processo de verificação de software é realizado ao longo de todo o desenvolvimento do sistema. Podem ser utilizadas para controlar a qualidade de uma variedade de atividades, tais como verificação do contrato, verificação do processo, verificação dos requisitos, verificação do projeto, verificação do código, verificação da integração e verificação da documentação. Estas atividades que agora serão relatadas servirão para a realização do sistema de apoio ao processo de verificação que será implementado. Para cada uma dessas atividades do processo de verificação da norma ISO/IEC 12207 o software realizará um *checklist* para verificar se o desenvolvimento da atividade está de acordo com a norma ISO/IEC 12207.

### 2.3.1 VERIFICAÇÃO DO CONTRATO

Segundo Rocha (2001), verificação do contrato observa se o fornecedor possui capacidade para satisfazer os requisitos, se esses são coerentes e abrangem as necessidades do usuário, se existem procedimentos adequados para manipular as mudanças nos requisitos e critérios de aceitação.

Nesta atividade o adquirente deve primeiro estabelecer um procedimento para selecionar o fornecedor, incluindo critérios de avaliação da proposta e de ponderação com relação à aderência aos requisitos de aquisição levantados na fase de iniciação do processo. Feito isso, ele pode então selecionar um fornecedor com base na avaliação das propostas recebidas dos fornecedores, das capacidades de cada fornecedor e de outros fatores que também precisam ser considerados.

Para a fase de adaptação do processo ao projeto, o adquirente pode envolver outras partes além dos fornecedores potenciais, antes do fechamento do contrato. Entretanto, o adquirente deverá tomar a decisão final sobre essa adaptação, bem como incluir ou referir a norma adaptada no contrato. Esse contrato deve ser preparado e negociado com o fornecedor e deve tratar dos requisitos de aquisição e incluir custos e cronograma, bem como os direitos de uso, propriedade, autoria, garantia e licença associados com os produtos de software de prateleira que serão utilizados.

Quando o contrato estiver em andamento, o adquirente também deve controlar possíveis alterações no mesmo por meio de negociação com o fornecedor. Essas alterações

devem ser investigadas quanto ao impacto nos planos, nos custos, nos benefícios, na qualidade e no cronograma do projeto.

De acordo com Mills (1994), avaliação do fornecedor é uma técnica usada por vários grandes compradores para avaliar a capacidade de um fornecedor potencial de fornecer um determinado produto, serviço ou processo. Uma avaliação de fornecedor normalmente cobre todos os aspectos do fornecedor potencial, incluindo fatores como estabilidade financeira, capacidade de projeto, capacidade de manufatura, sistema da qualidade, etc. As técnicas de verificação podem ser úteis na avaliação de todos esses aspectos. Uma vez firmado o contrato, as avaliações podem ser realizadas para confirmar a observância do sistema da qualidade estabelecido.

A verificação do fornecedor é a verificação ou observação de determinadas atividades. Frequentemente ela é utilizada pela indicação de inspeções visuais ou atividades de teste realizadas nas instalações do fornecedor, em vez de controles posteriores. A verificação do fornecedor frequentemente constitui um método eficaz de assegurar que determinados critérios estejam sendo observados, em vez de adotar uma “inspeção de recebimento” pelo comprador.

Uma exigência para determinar a adequação ou conformidade ao vários controles aplicados ao processo sendo fornecido está agrupada sob o termo “verificação do fornecedor”. Um caso de avaliação de fornecedor é o desenvolvimento de uma lista de fornecedores aprovados pela corporação. A lista forma então a base para contactar fornecedores potenciais em relação a negócios futuros.

Cientes de que a compra foi iniciada e que sabem as datas esperadas de recebimento dado início aos passos necessários para a verificação dos bens e serviços: contrato da área da qualidade do fornecedor, verificação do fornecedor, inspeção do fornecedor, inspeção de recebimento, verificação de dados gerados pelo fornecedor, isto é, certificados, resultados, gráficos de controle de processo...

De acordo com Pádua (2001), o esforço de uma organização para melhorar a qualidade de seus sistemas informatizados pode ser perdido por causa de falhas de contratados. Para que isso não aconteça, a organização contratante deve estar capacitada em gestão de contratos.

### **2.3.2 VERIFICAÇÃO DO PROCESSO**

Segundo Rocha (2001), verificação do processo observa se o planejamento do projeto e a atribuição de tempos estão adequados, se o que foi determinado no projeto está sendo seguido e está de acordo com o contrato, se a equipe possui o treinamento desejado.

Cabe ao fornecedor conduzir uma revisão dos requisitos de aquisição, como o objetivo de definir a estrutura e estabelecer os planos, os quais serão utilizados para gerenciar o projeto e garantir a qualidade do produto ou serviço de software que será entregue. No levantamento dos requisitos para os planos é interessante que o fornecedor inclua as necessidades de recursos, bem como o modelo de envolvimento do adquirente.

Um ponto importante a se considerar é o modelo de ciclo de vida que será usado. Se não estiver estipulado no contrato, o fornecedor deve definir ou selecionar um modelo de ciclo de vida de software apropriado para abrangência, o objetivo, a magnitude e a complexidade do projeto. Os processos, as atividades e as tarefas relacionadas com a norma ISO/IEC 12207 devem ser selecionados e mapeados nesse modelo de ciclo de vida adotado.

Já para Staa (2000), é reconhecido que um dos principais fatores para os problemas com projetos de desenvolvimento de software não se situam na área técnica, mas sim na gerência do processo. A grosso modo, a falta de coordenação, acompanhamento e controle do desenvolvimento é uma das principais causas de insucesso não só financeiro como também técnico. A falta de uma especificação de boa qualidade, a falta de uma definição precisa do que vai ser o resultado do desenvolvimento, a falta de um planejamento abrangente e consistente com os requisitos do sistema a desenvolver, o acompanhamento inexistente ou errado, a falta de controle da qualidade e de controle das alterações, são os principais vilões.

### **2.3.3 VERIFICAÇÃO DOS REQUISITOS**

De acordo com Rocha (2001), verificação dos requisitos observa se os requisitos do sistema são coerentes, são factíveis e testáveis, se os requisitos do software são coerentes, são factíveis e testáveis e se refletem com precisão os requisitos do sistema.

Segundo Pádua (2001), os requisitos são as características que definem os critérios de aceitação de um produto. Uma especificação de requisitos pode conter requisitos incompletos, inconsistentes ou ambíguos. Alguns desses problemas decorrem da própria linguagem natural,

que normalmente é usada para expressá-los. Outros decorrem de técnicas deficientes de elaboração dos requisitos.

Um problema comum no desenvolvimento de software é a instabilidade dos requisitos, que ocorre quando cliente e usuários trazem novos requisitos, ou alterações de requisitos, quando o desenvolvimento já está em fase adiantada. A instabilidade dos requisitos costuma ter um custo muito alto, geralmente significa perder trabalho já feito, desfazer algumas coisas e remendam outras.

A boa engenharia de requisitos reduz a instabilidade destes, ajudando a obter os requisitos corretos em um estágio anterior ao desenvolvimento. Entretanto, alterações dos requisitos são às vezes inevitáveis. A engenharia de requisitos é sujeita a limitações humanas, e mesmo que o levantamento seja perfeito, podem ocorrer alterações de requisitos por causas externas aos projetos.

Note-se que defeitos incluem situações de falta de conformidade com requisitos explícitos, normativos e implícitos. Os defeitos associados a requisitos são os mais difíceis de tratar. Eles levam a desentendimentos sem solução entre o fornecedor e o cliente do produto. Além disso, com esses requisitos, por definição, não são documentados, é bastante provável que eles não tenham sido levados em conta no desenho do produto, o que tornará a correção dos defeitos particularmente trabalhosa.

O fluxo de requisitos reúne as atividades que visam a obter o enunciado completo, claro e preciso dos requisitos de um produto de software. Estes requisitos devem ser levantados pela equipe do projeto, em conjunto com representantes do cliente, usuários-chaves e outros especialistas da área de aplicação. O conjunto de técnicas empregadas para levantar, detalhar, documentar e validar os requisitos de um produto forma a engenharia de requisitos. O resultado principal do fluxo dos requisitos é um documento de Especificação de Requisitos de Software.

Os requisitos de alta qualidade são claros, completos, sem ambigüidade, implementáveis, consistentes e testáveis. Os requisitos que não apresentam estas qualidades são problemáticos: eles devem ser revistos e renegociados com os clientes e usuários.

A revisão dos requisitos tem por objetivo assegurar que a especificação dos requisitos do Software:

- esteja conforme com o respectivo padrão, e com outros padrões aplicáveis ao projeto em questão;
- atenda aos critérios de qualidade dos requisitos;
- forneça informação suficiente para o desenho do produto, de seus testes de aceitação e do seu manual de usuário.

### **2.3.4 VERIFICAÇÃO DO PROJETO**

Segundo Rocha (2001), verificação do projeto observa se o projeto está correto e coerente com os requisitos, se ele implementa apropriadamente as seqüências de eventos, entradas, saídas, requisitos de segurança com métodos rigorosos.

Em Freedman (1993), o local onde a maioria dos erros se encontram depende do estado da arte da instalação. Onde não são praticadas revisões de espécie alguma, nem mesmo as revisões informais de código dentro da própria equipe de programação, a maioria dos erros é encontrada no nível do código. Entretanto, os erros de projetos podem mostrar-se em média mais sérios, do que os erros de código. Os erros de projeto são, na média, mais difíceis de serem detectados, corrigidos quando detectados, e requerem modificações muito mais caras e extensas.

### **2.3.5 VERIFICAÇÃO DO CÓDIGO**

Segundo Rocha (2001), verificação do código observa se o código está de acordo com os requisitos, se é testável e correto, se está de acordo com os requisitos e padrões de codificação, se implementa requisitos críticos e de segurança com métodos rigorosos.

Conforme Rocha (2001), o desenvolvedor deve codificar e documentar cada unidade de software e definir a base de dados usando técnicas e implementação que produzam um código eficiente e livre de erros. Como resultado de uma implementação bem-sucedida, as unidades de software devem ser obtidas e critérios de verificação devem ser definidos para todas elas. A atividade de teste combina várias etapas com uma séria de métodos de projeto de casos de teste que ajudaram a garantir que erros sejam efetivamente detectados.

Segundo Staa (2000), cada padrão estabelece regras que são de aplicações obrigatória, exceções que estabelecem alternativas para regras e recomendações que devem ser seguidos na medida do possível. Os padrões podem ser adaptados a cada organização, desde que o seu

espírito seja mantido. Ou seja, organizações que já possuem padrões podem ajustar os seus padrões e os aqui apresentados de modo a minimizar o impacto na organização. Já organizações que não possuam um programa de qualidade estabelecido são encorajadas a adotar padrões.

Como consequência da adoção dos padrões, espera-se uma redução significativa de ocorrência dos erros de programação mais comuns. Os padrões estabelecem um estilo de programação uniforme.

Para poder assegurar níveis de qualidade satisfatória, é necessário atuar sobre o processo de desenvolvimento de modo que se comentam poucos erros (idealmente nenhum). Em particular é necessário atuar sobre o processo de codificação de modo que, entre outros requisitos de qualidade, os programas estejam (quase) corretos por construção e que sejam de fácil compreensão e manutenção.

O benefício de um padrão somente será percebido caso seja efetivamente adotado por todos os participantes do desenvolvimento. A adoção é comprovada verificando a conformidade dos artefatos com os padrões. Em uma grande parte das vezes, os controles de conformidade são realizados por meios de inspeções.

### **2.3.6 VERIFICAÇÃO DA INTEGRAÇÃO**

Segundo Rocha (2001), verificação da integração observa se os componentes e unidades de código, bem como itens de hardware e software foram integrados completa e corretamente de acordo como um plano de integração.

Conforme Anacleto (1996), o gerenciamento integrado de software tem o propósito de integrar as atividades de engenharia e gerenciamento de software em um processo de software coerente e definido, que é adaptado a partir do processo padronizado de software da organização e seus respectivos recursos. Esta área envolve o desenvolvimento do processo de software para o projeto e o gerenciamento do projeto de software usando este processo. O processo de software definido para o projeto é uma versão adaptada do processo padronizado de software da organização. O projeto é planejado e gerenciado de acordo com o processo de software definido para o projeto.

### **2.3.7 VERIFICAÇÃO DA DOCUMENTAÇÃO**

Segundo a norma ABNT (1998), verificação da documentação observa se a documentação está adequada, completa, coerente, se está sendo desenvolvida no prazo e se o gerenciamento de configuração da documentação dos documentos segue os procedimentos especificados.

Conforme Rocha (2001), a documentação de software é uma atividade fundamental no processo de desenvolvimento de software. Cabe à documentação registrar a evolução do software para que sejam criadas as bases necessárias para melhor utilização e manutenção do software.

Tradicionalmente, pouca atenção tem sido dada à documentação gerada durante os projetos de desenvolvimento que tem resultado em documento mal-elaborados, de difícil compreensão, inadequados ou até mesmo incompletos. Os maiores especialistas na área de engenharia de software reconhecem que a falta de um projeto de documentação tem atrapalhado a manutenção do software durante toda a história da computação. A utilização do software também tem sido degradada por não se ter disponíveis informações corretas sobre todas as possibilidades oferecidas pelo produto de software desenvolvido. Por essas razões, a elaboração de uma documentação técnica de qualidade relacionada ao processo de software é tão importante quanto à qualidade do software em si.

Segundo Inthurn (2001), fornecer documentos completos e consistentes durante todo o ciclo de vida de desenvolvimento do software aumenta as chances de se obter um produto final de alta qualidade. Além disso, checar as falhas dos documentos do software antes de continuar com a próxima fase do desenvolvimento contribui para a qualidade global do software.

Segundo Freedman (1993), a documentação é certamente uma forma importante que o sistema assume. Embora ela possa incluir toda ou parte das especificações, projeto e código, essas partes não incluirão, geralmente, o conjunto completo de documentação. Conseqüentemente, partes adicionais deverão ser submetidas às revisões de documentação. Além disso, pode ser interessante, em alguns casos, revisar especificações, projetos ou códigos em termos de sua adequação à documentação do sistema, em contraste com sua adequação à documentação funcional durante o desenvolvimento.



Diferentes tipos de documentação podem ser revisados em diferentes tempos, ou mesmo em diversas épocas com objetivos diferentes, mas toda a documentação deve ser revisada pelo menos uma vez. Por exemplo, os comentários dentro do código devem ser revisados à medida que o código é revisado, antes de sua inclusão no sistema. Os comentários são considerados parte do código propriamente, mas são também claramente uma documentação. Na verdade, o código em si é um tipo de documentação e em alguns casos é revisado separadamente quanto à sua qualidade como documento.

Conforme Duarte (2000), manter padrões na documentação é a base para a garantia de qualidade dos documentos. Documentos produzidos de acordo com padrões apropriados possuem qualidade, estrutura e aspectos consistentes.

## 3 NORMAS E MODELOS DE QUALIDADE

Este capítulo trata das outras normas, da futura norma ISO/IEC 15504 (SPICE), da norma ISO 9000-3 e do modelo CMMI, a norma ISO/IEC 12207 já foi comentada no capítulo 2. Será apresentada uma breve descrição de cada alternativa, assim como uma explicação do processo de verificação previsto em cada alternativa.

### 3.1 NORMA ISO/IEC 15504

De acordo com Iahn (1999), a futura norma ISO/IEC 15504, conhecida atualmente como projeto SPICE (*Software Process Improvement and Capability dEtermination*), é uma norma em elaboração formada pela ISO (*Intenational Organization for Standardization*) e pela IEC (*International Electrotechnical Comission*), e constitui-se de um padrão para a avaliação do processo de software, visando determinar a capacitação de uma organização.

A futura norma ISO/IEC 15504 define processos e práticas que podem ser implementados para estabelecer e aprimorar a capacidade de aquisição, desenvolvimento, manutenção, operação e suporte de software em organizações. Estas práticas são organizadas utilizando-se uma arquitetura que combina duas perspectivas: uma perspectiva funcional (análoga à perspectiva da norma ISO/IEC 12207), compreendendo quais os processos que devem existir numa organização e uma outra perspectiva que avalia o nível de capacitação de cada um destes processos (análoga ao CMMI). O uso da norma permite que as organizações possam perceber a existência ou não de processos específicos, bem como a capacitação dos que existem, traçando um caminho para melhoria.

Além dos processos, o SPICE define também os seis níveis de capacitação de cada processo, que pode ser incompleto, executado, gerenciado, estabelecido, previsível e otimizado.

### 3.2 PROCESSO DE VERIFICAÇÃO SEGUNDO A NORMA ISO/IEC 15504

Conforme Emam (1998), o processo de verificação se encontra na categoria de suporte 4, que descreve como executar a verificação dos produtos de trabalho e inclui atividades como:

- confirmar se cada produto de software e/ou serviço de um processo ou projeto estão de acordo com os requisitos;
- confirmar se cada produto de trabalho ou serviço resultado de um processo reflete corretamente às especificações de entrada do processo;
- verificar desempenho e conformidade dos processos com padrões e procedimentos;
- verificar correteza e completeza do software entregue;
- avaliar documentos;
- revisar atividades de engenharia do software;
- assegurar que os defeitos encontrados serão removidos dos produtos de trabalho e que os resultados serão disponibilizados para os clientes;
- realizar testes para verificar atendimento aos requisitos.

### **3.3 NORMA ISO 9000-3**

Segundo Bento (2000), a ISO 9000-3 é uma guia de aplicação da ISO 9001 (NBR 19001) para o desenvolvimento, fornecimento e manutenção de software. A Norma ISO 9001 faz parte da série de normas ISO 9000, 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 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.

Conforme a norma ISO 9000-3 o processo de certificação de uma empresa de software segundo as normas ISO 9001 / 9000-3 segue um conjunto de passos bem definidos (ABNT, 1993):

- a) a empresa estabelece o seu sistema de qualidade;
- b) 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;

- c) o órgão certificador faz uma visita à empresa, colhe mais dados e explica o processo de certificação;
- d) o órgão certificador verifica se a documentação do sistema de qualidade está de acordo com a norma ISO;
- e) 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;
- f) o órgão certificador emite o certificado de qualidade;
- g) o órgão certificador realiza visitas periódica à empresa par assegurar que o sistema continua sendo efetivo.

Apesar de um processo de desenvolvimento de software ser considerado distinto dos demais processos de fabricação de outros produtos conforme a norma ISO 9000-3 da é extremamente necessário que esse campo de tecnologia seja provido de orientações adicionais para o estabelecimento de sistemas da qualidade onde estejam envolvidos os produtos de software (ABNT 1993).

A norma ISO 9000-3 define algumas diretrizes, a fim de tornar possível a aplicação da NBR 19001 nas organizações que desenvolvem, fornecem e mantém software. Estas diretrizes destinam-se a descrever os controles e métodos sugeridos para a produção de software que atendam aos requisitos do comprador, evitando-se não conformidades em todos os estágios.

### **3.4 PROCESSO DE VERIFICAÇÃO SEGUNDO A NORMA ISO 9000-3**

A norma ISO 9000-3 determina que uma organização deve definir, documentar, compreender, implementar e manter uma política de qualidade, definir responsabilidade e autoridade para pessoas que gerenciam projetos, verificar a qualidade do serviço, identificar e providenciar recursos para a verificação.

Com o grande crescimento de produção de sistemas, é necessário estabelecer um sistema de gestão de qualidade e fornecer orientação para a garantia da qualidade de software. Ela relata sobre as definições de software, produto de software, item de software, desenvolvimento, fase, verificação de software e validação para software (Barbaresco, 2000).

O fornecedor deve elaborar as atividades de verificação de cada fase. A verificação do desenvolvimento deve estabelecer que as saídas de cada fase atendam aos requisitos de entrada correspondentes, mediante a adoção de medidas de controle de desenvolvimento, tais como:

- a) análise crítica do desenvolvimento em pontos apropriados;
- b) comparação do novo projeto com um projeto semelhante já comprovado, caso haja algum disponível;
- c) realização de ensaios e demonstrações.

Os resultados da verificação, e quaisquer ações suplementares requeridas para assegurar que os requisitos especificados foram atendidos, devem ser registrados e verificados quando da conclusão destas ações.

Segundo Fernandes (1995) a norma ISO 9000-3 o fornecedor deve definir e documentar uma política da qualidade. A política deve ser entendida por todos na organização. As responsabilidades e autoridades quanto às atividades relativas à qualidade, na organização, devem estar claramente definidas. Os recursos e pessoal para verificação do sistema da qualidade devem estar definidos.

O fornecedor deve indicar um representante da administração, com responsabilidade e autoridade para assegurar que o sistema da qualidade esteja em operação conforme os requisitos da norma. O fornecedor e o comprador devem realizar revisões conjuntas para verificar aderência do software às especificações e resultados de teste. O fornecedor deve estabelecer, documentar e manter procedimentos para investigar causas de não-conformidade do produto, analisar processos, iniciar ações preventivas, aplicar controles para que as ações corretivas sejam realizadas e implementar e registrar mudanças nos procedimentos em função dos resultados da ação corretiva.

Na revisão do contrato, o fornecedor deve estabelecer e manter procedimentos para a revisão do contrato e para a coordenação dessas atividades. O fornecedor deve revisar cada contrato para assegurar que o escopo e requisitos sejam definidos e documentados, que as contingências sejam identificadas, que informação proprietária seja adequadamente protegida, que quaisquer requisitos diferentes daqueles na proposta sejam solucionados, que a responsabilidade do fornecedor com subcontratos esteja definida. O fornecedor deve manter

registros dessas revisões. O contrato deve prever critérios de aceitação do software, tratamento de mudança dos requisitos do comprador durante o desenvolvimento, o tratamento de problemas detectados após a aceitação, as atividades a serem desempenhadas pelo comprador no que concerne a especificação dos requisitos, a definição das facilidades, ferramentas e itens de software a serem fornecidos pelo comprador, procedimentos e padrões a serem usados e requisitos de replicação do software.

Devem ser designadas pessoas, de ambas as partes, para o estabelecimento das especificações dos requisitos do comprador. O plano de desenvolvimento deve conter a definição do projeto em termos de objetivos e relacionamento com outros projetos do comprador ou do fornecedor, a organização dos recursos para o projeto, incluindo estrutura da equipe, responsabilidades, uso de subcontratados e recursos materiais, as fases de desenvolvimento, o cronograma do projeto e a identificação de planos correlatos, como plano da qualidade, plano de gerência de configuração, plano de integração e plano de teste.

O plano de desenvolvimento deve ser atualizado na medida do progresso do projeto para cada fase do desenvolvimento devem ser estabelecidas as entradas e saídas. O plano deve prever também como o projeto será gerenciado. O plano deve identificar as regras, práticas, convenções, ferramentas, técnicas e a gerência de configuração. O fornecedor deve verificar as saídas de todas as fases do desenvolvimento.

A implementação, o fornecedor deve desempenhar revisões e assegurar que os requisitos sejam atendidos e que os métodos de projeto e implementação sejam adequadamente utilizados. O fornecedor deve manter registros documentados de tais revisões.

Antes de desempenhar as atividades de aceitação, o fornecedor deve assistir o comprador na identificação da programação da aceitação, procedimentos para avaliação, o ambiente de hardware/software e recursos necessários e os critérios de aceitação a serem utilizados.

Na entrega e instalação deve ser verificada:

- a determinação do número de cópias do software a serem entregues;
- o tipo de meio magnético a ser utilizado, incluindo formatos;
- a estipulação da documentação requerida;
- questões relativas a direitos autorais;

- o período de obrigação do fornecedor suprir cópias;
- as cópias a serem entregues deverão ser verificadas quanto a sua completeza;
- os papéis do fornecedor e comprador deverão estar plenamente estabelecidos quanto à instalação.

### 3.5 MODELO CMMI

Segundo Junior (2000), foi desenvolvido o modelo *CMM Integrated System/ Software Engineering*. O CMM-SE/SW não é apenas uma reunião dos modelos CMM existentes, mas sim um *framework* que acomoda múltiplas disciplinas e é flexível o suficiente para suportar duas representações diferentes (de estágios e contínuo), uma para verificar o nível de maturidade dos processos, outra para o nível de maturidade da organização como um todo.

Conforme Belloquim (2002), o CMMI, além de incorporar as melhorias propostas aprendidas em mais de uma década de uso do modelo SW-CMM, compatibiliza este modelo como os demais CMMs do SEI e com a ISO/IEC 15504. Mas sua contribuição mais importante é o grande aumento da flexibilidade na implantação de projetos de melhoria dos processos de software. Com a dupla representação, a implantação do CMM atinge um ponto em que as necessidades de cada empresa podem ser levadas em conta como nunca.

Segundo Junior (2000), o CMMI é um projeto que envolve um grande número de pessoas, patrocinado pelo Departamento de Defesa Americano (DoD), desenvolvido pelo SEI em parceria com organizações da indústria e governo.

O propósito do modelo CMMI é fornecer guias para melhorar os processos organizacionais e a habilidade de gerenciar o desenvolvimento e manutenção de produtos e serviços, através da verificação do *status* da melhoria de processos, do estabelecimento de prioridades para melhoria e da implantação desses processos.

O projeto foi construído para atingir um conjunto inicial de modelos que cobrem três disciplinas: engenharia de software, engenharia de sistemas e desenvolvimento integrado de produto e processo. Entre as principais metas traçadas para o modelo CMMI estão:

- a) integração dos modelos de origem, eliminando inconsistências e reduzindo duplicações;
- b) redução de custos ao implantar uma melhoria de processos modelo-baseada;

- c) aumento da clareza e compreensão do modelo, através de uma terminologia comum, estilo consistente, regras de construção uniformes e componentes comuns;
- d) necessidade de assegurar aos produtos desenvolvidos para o modelo consistência com a futura norma ISO 15504.

Sendo assim, os principais benefícios apontados são uma eficiente e efetiva avaliação e melhoria através da disciplina de múltiplos processos em uma organização, custos de treinamento e avaliação reduzidos, uma visão comum e integrada de melhoria para todos os elementos de uma organização.

Na representação por estágios as áreas de processo estão agrupadas em estágios (níveis) de 2 a 5. Cada área de processo contém práticas (atividades) a serem implementadas para concluir o propósito da área de processo.

Os níveis de maturidade para tal representação do modelo CMMI estão assim definidos: inicial, gerenciado, definido, quantitativamente gerenciado, otimizado.

### **3.6 PROCESSO DE VERIFICAÇÃO SEGUNDO O MODELO CMMI**

Segundo Junior (2000), uma organização pode optar por alcançar a melhoria de processo sob duas abordagens: capacitação de processo ou maturidade organizacional. O modelo CMMI suporta a capacitação de processo através de uma representação contínua e a maturidade da organização através de uma representação por estágios.

O nível de maturidade dos processos onde a área de verificação esta relacionada é o definido. O propósito da verificação é assegurar que os produtos de trabalho selecionados satisfazem aos seus requisitos especificados.

O processo de verificação (nível 3), descrevem os passos para assegurar que as atividades estão sendo executadas de acordo como o processo estabelecido. O propósito da verificação é assegurar que os produtos de trabalho selecionados satisfazem aos seus requisitos específicos. As metas específicas para alcançar ou implementar esta área-chave são as seguintes:

- preparar para a verificação: a preparação para a verificação é conduzida;
- executar as revisões por parceiros: revisões por parceiros são executadas nos produtos de trabalho selecionados;



- verificar os produtos de trabalho selecionados: os produtos de trabalho selecionados são verificados com relação aos seus requisitos especificados.

A verificação também pode ser encontrado em outros níveis do modelo CMMI, na integração de produto, gerência de requisitos, planejamento de projeto, controle de monitoramento de projeto, gerência de contrato, análise casual e resolução.

O propósito da integração de produto (nível 3) é montar o produto dos componentes de produto, assegurar que o produto, como integrado, funcione corretamente e entregar o produto. As metas específicas para alcançar ou implementar esta área-chave são as seguintes:

- preparar a integração para o produto: a estratégia para conduzir a integração do produto é estabelecida e mantida;
- assegurar a compatibilidade de interface: as interfaces componentes do produto, tanto internas quanto externas, são compatíveis;
- montar os componentes de produto e entregar o produto: os componentes de produto verificados são montados e o produto integrado, verificado e validado é entregue.

O propósito da gerência de requisitos (nível 4) é gerenciar os requisitos de produtos do projeto e componentes do produto e identificar inconsistências entre o planos de projeto e produtos de trabalho e os requisitos. A meta específica para alcançar ou implementar esta área-chave é a seguinte:

- gerenciar requisitos: os requisitos são gerenciados e inconsistências com planos de projeto e produtos de trabalho são identificados.

O propósito do planejamento de projeto (nível 4) é estabelecer e manter planos que definam as atividades de projeto. As metas específicas para alcançar ou implementar esta área-chave são as seguintes:

- estabelecer estimativas: estimativas de parâmetros de planejamento de projeto são estabelecidas e mantidas;
- desenvolver um plano de projeto: um plano de projeto é estabelecido e mantido como a base para gerenciar o projeto;
- obter compromisso para o plano: compromissos para o plano de projeto são estabelecidos e mantidos.

O propósito do controle e monitoramento de projeto (nível 4) é prover conhecimento no progresso do projeto para que ações corretivas apropriadas possam ser tomadas quando o desempenho do projeto divergir significativamente do plano. As metas específicas para alcançar ou implementar esta área-chave são as seguintes:

- monitorar o projeto defronte ao plano: o desempenho atual e o progresso do projeto são monitorados contra o plano de projeto;
- gerenciar ação corretiva até sua conclusão: são administradas ações corretivas e gerenciadas até sua conclusão quando o desempenho do projeto ou resultados divergem significativamente do plano.

O propósito da gerência de contrato de fornecedor (nível 4) é gerenciar a aquisição de produtos e serviços de fornecedores externos ao projeto para os quais existem um acordo formal. As metas específicas para alcançar ou implementar esta área-chave são as seguintes:

- estabelecer contratos com fornecedores: são estabelecidos e mantidos contratos com os fornecedores;
- satisfazer os contratos com fornecedores: contratos com os fornecedores são satisfeitos por ambos, o projeto e o fornecedor.

O propósito da análise causal e resolução (nível 5) é identificar causas de defeitos e outros problemas e tomar ações para lhes evitar de acontecer no futuro. As metas específicas para alcançar ou implementar esta área-chave são as seguintes:

- determinar as causas de defeitos: as origens de causas de defeitos e outros problemas são sistematicamente determinadas;
- endereçar as causas de defeitos: as origens de causas de defeitos e outros problemas são sistematicamente endereçadas para prevenir sua ocorrência futura.

### **3.7 QUADRO COMPARATIVO**

Após realizado o estudo do processo de verificação das normas ISO/IEC 12207, ISO 9000-3, ISO/IEC 15504/SPICE e o modelo CMMI, chegou-se a um comparativo, conforme a tabela 1. Essa tabela apresenta o que cada norma diz a respeito dos processos de verificação.

Nesta tabela a primeira coluna indica as atividades que devem ser atendidas pelos processos de verificação de software, e as demais colunas indicam os nomes das normas de qualidade.

O critério adotado para especificar essa tabela foi separar cada atividade do processo de verificação das normas e modelo de qualidade e colocá-los lado a lado para fazer um comparativo para verificar se há pontos em comum nessas normas.

**TABELA 1 – COMPARATIVO DO PROCESSO DE VERIFICAÇÃO DE SOFTWARE**

<b>Atividade de verificação</b>	<b>ISO/IEC 12207</b>	<b>ISO 9000-3</b>	<b>ISO/IEC 15504 (SPICE)</b>	<b>CMMI</b>
verificação do contrato	<p>Observa se o fornecedor possui capacidade para satisfazer os requisitos, se esses são coerentes e abrangem as necessidades do usuário, se existem procedimentos adequados para manipular as mudanças nos requisitos, critérios de aceitação etc.</p> <p><b>(6.4.2.1)</b></p>	<p>O fornecedor deve estabelecer e manter procedimentos para análise crítica de contrato e para a coordenação destas atividades. Revisão do contrato, o fornecedor deve estabelecer e manter procedimentos para a revisão do contrato e para a coordenação dessas atividades.</p> <p><b>(5.2.1)</b></p>	<p>A empresa tem o objetivo de avaliar um fornecedor em potencial, obtendo o seu perfil de capacidade. Para isso ela deve definir os objetivos e o contexto da avaliação, os modelos e métodos de avaliação e os requisitos esperados. O perfil de capacidade permite ao contratante estimar o risco associado à contratação daquele fornecedor em potencial para auxiliar na tomada de decisão de contrata-lo ou não.</p> <p><b>(SUP 4)</b></p>	<p>Gerência de contrato de fornecedor tem o objetivo de gerenciar a aquisição de produtos e serviços de fornecedores externos ao projeto para os quais existem um acordo formal. Estabelecer contratos com fornecedores. Satisfazer os contratos com fornecedores.</p> <p><b>(nível 4)</b></p>
verificação do processo	<p>Observa se o planejamento do projeto e a atribuição de tempos estão adequados, se o que foi determinado no projeto está sendo seguido e está de acordo com o contrato, se a equipe possui o treinamento desejado etc.</p> <p><b>(6.4.2.2)</b></p>	<p>Organização dos recursos do projeto, programação do projeto, fases de desenvolvimento, planos da qualidade, procedimentos de verificação, interfaces organizacionais e técnicas, responsabilidades organizacionais e controle da execução.</p> <p><b>(5.5.2)</b></p>	<p>Confirmar que cada produto de trabalho ou serviço resultado de um processo reflete corretamente às especificações de entrada do processo.</p> <p><b>(SUP 4.3, SUP 4.4)</b></p>	<p>Planejamento do projeto de software tem o propósito de estabelecer planos razoáveis para a execução de atividades de engenharia de software e para o gerenciamento do projeto de software. Esta área envolve o desenvolvimento de estimativas para o trabalho a ser executado, o estabelecimento de compromissos necessários e a definição de um plano de execução do trabalho.</p> <p><b>(nível 3)</b></p>

verificação dos requisitos	<p>Observa se os requisitos do sistema são coerentes, são factíveis e testáveis, se os requisitos do software são coerentes, são factíveis e testáveis e se refletem com precisão os requisitos do sistema etc.</p> <p><b>(6.4.2.3)</b></p>	<p>A verificação do desenvolvimento deve estabelecer que as saídas de cada fase atendam aos requisitos de entrada correspondentes, mediante a adoção de medidas de controle de desenvolvimento.</p> <p><b>(5.4.6)</b></p>	<p>Envolve a definição de uma estratégia de verificação, de critérios de verificação para todos os produtos de trabalho e para as atividades de verificação para confirma que cada produto de software e/ou serviço de um processo ou projeto estão de acordo com os requisitos.</p> <p><b>(SUP 4.6)</b></p>	<p>A gerência de requisitos tem como objetivo gerenciar os requisitos de produtos do projeto e componentes do produto e identificar inconsistências entre o planos de projeto e produtos de trabalho e os requisitos.</p> <p><b>(nível 4)</b></p>
verificação do projeto	<p>Observa se o projeto está correto e coerente com os requisitos, se ele implementam apropriadamente as seqüências de eventos, entradas, saídas, requisitos de segurança com métodos rigorosos etc.</p> <p><b>(6.4.2.3)</b></p>	<p>O fornecedor deve elaborar um plano para verificação de todas as saídas ao final de cada fase de desenvolvimento, comparação do novo projeto com um projeto semelhante já aprovado, caso haja algum disponível, assegurar que os requisitos especificados foram atendidos.</p> <p><b>(5.6.2)</b></p>		<p>O propósito do controle e monitoramento é de prover conhecimento no progresso do projeto para que ações corretivas apropriadas possam ser tomadas quando o desempenho do projeto divergir significativamente do plano. Monitorar o projeto defronte ao plano. Gerenciar ação corretiva até sua conclusão.</p> <p><b>(nível 4)</b></p>
verificação do código	<p>Observa se o código está de acordo com os requisitos, se é testável e correto, se está de acordo com os requisitos e padrões de codificação, se implementa requisitos críticos e de segurança com métodos rigorosos etc.</p> <p><b>(6.4.2.5)</b></p>	<p>Desempenha e verifica atividades de implementação das soluções.</p> <p><b>(4.1.1.2.1)</b></p>	<p>Normalmente envolve a realização de testes e está relacionado aos processos ENG.1.6 teste de software para verificar atendimento aos requisitos (inclui regressão) e ENG.1.7 integração e teste de sistema (obs: toda a parte de verificação é semelhante ao processo SUP.4). Pode também fazer uso de técnicas como <i>peer reviews</i>, provas formais e análise de rastreabilidade.</p> <p><b>(SUP 4.5)</b></p>	
verificação da integração	<p>Observa se os componentes e unidades de código, bem</p>	<p>As atividades de verificação devem incluir inspeção, ensaio e</p>	<p>Verificar desempenho, verificar conformidade dos processos com padrões e</p>	<p>Integração de produto tem como objetivo montar o produto dos componentes de produto, assegurar que</p>

	como itens de hardware e software foram integrados completa e corretamente de acordo como um plano de integração etc. <b>(6.4.2.6)</b>	monitorização de processos e produto de projeto, produção, instalação e de assistência técnica; análises críticas de projeto. <b>(4.1.1.2.2)</b>	procedimentos, verificar conformidade dos produtos, verificar correteza e completeza do software entregue. <b>(SUP 4.2)</b>	o produto, como integrado, funcione corretamente e entregar o produto. Preparar a integração para o produto. Assegurar a compatibilidade de interface. Montar os componentes de produto e entregar o produto. <b>(nível 3)</b>
Verificação da documentação	Observa se a documentação está adequada, completa, coerente, se está sendo desenvolvida no prazo e se o gerenciamento de configuração dos documentação segue os procedimentos especificados. <b>(6.4.2.7)</b>	Os requisitos de entrada para cada fase do desenvolvimento devem ser definidos e documentados, de modo que o seu atendimento possa ser verificado. <b>(6.2)</b>	Deve-se avaliar a documentação. <b>(SUP 4.1)</b>	

O resumo do comparativo foi realizado da seguinte maneira: tornou-se uma atividade como referência, que foi retirada da norma ISO/IEC 12207, cuja atividade correspondia a um processo ou uma cláusula das normas. Essa atividade foi colocada na primeira coluna do quadro. Nas outras colunas foram colocadas lado a lado as atividades correspondentes as normas, sendo que em alguns casos não havia uma atividade correspondente, nesse caso foi deixado um espaço em branco. Na coluna Atividades, foram colocados os processos (cláusulas) que apresentavam maior facilidade de interpretação e abrangência ao assunto proposto, sendo que a maior parte dessas atividades foi obtida da norma ISO/IEC 12207 por ela ser a mais completa referente ao processo de verificação de software.

Foi realizado um mapeamento das normas e modelos com o objetivo da possibilidade de uma análise a partir da leitura e comparação do processo de verificação dos itens em cada norma e modelo. Os trechos da tabela 1 foram retirados das suas respectivas normas e modelos e também de algumas monografias que já faziam o comparativo das mesmas. No texto aparece a referência ao segmento da norma (cláusula, artigo, seção, etc).

## 4 ESPECIFICAÇÃO DO PROTÓTIPO

Este capítulo trata da especificação do software. Serão apresentadas a lista de eventos, com uma breve descrição sobre cada evento, o diagrama de entidade relacionamento, tanto em nível lógico como físico, o dicionário de dados, o diagrama de contexto e os diagramas de fluxos de dados.

### 4.1 REQUISITOS DO PROBLEMA

O software a seguir foi desenvolvido segundo a proposta de processo de verificação demonstrada nos capítulos anteriores. Cabe lembrar que o objetivo do software proposto é apoiar o processos de verificação de software baseando-se na norma ISO/IEC 12207. Além disso o comparativo feito entre as normas dará subsídio para ampliar o processo de verificação. Este software deverá ser capaz de auxiliar o engenheiro de software no processo de verificação. Utilizou-se para isto um *checklist* para facilitar o apoio ao processo de verificação.

Segundo Pompilho (1994), a análise essencial considera três perspectivas: funções, dados e controles. Em relação ao grau de abstração, a análise essencial considera dois níveis: o nível essencial e o nível de implementação. O nível essencial é representado pelo chamado modelo essencial e o nível de implementação é representado pelo chamado de implementação. O método para a especificação do software utilizada foi a técnica de análise essencial.

### 4.2 LISTA DE EVENTOS

Segundo Pompilho (1994), “um evento pode ser definido informalmente como um acontecimento do mundo exterior que requer do sistema uma resposta”.

A rigor, o valor de um sistema está na sua capacidade de responder com eficácia a todos os estímulos a que for submetido. Assim, um sistema é construído para responder a estímulos. A cada estímulo, o sistema deve reagir produzindo uma resposta predeterminada.

Com isso, o software proposto possui a seguinte lista de eventos:

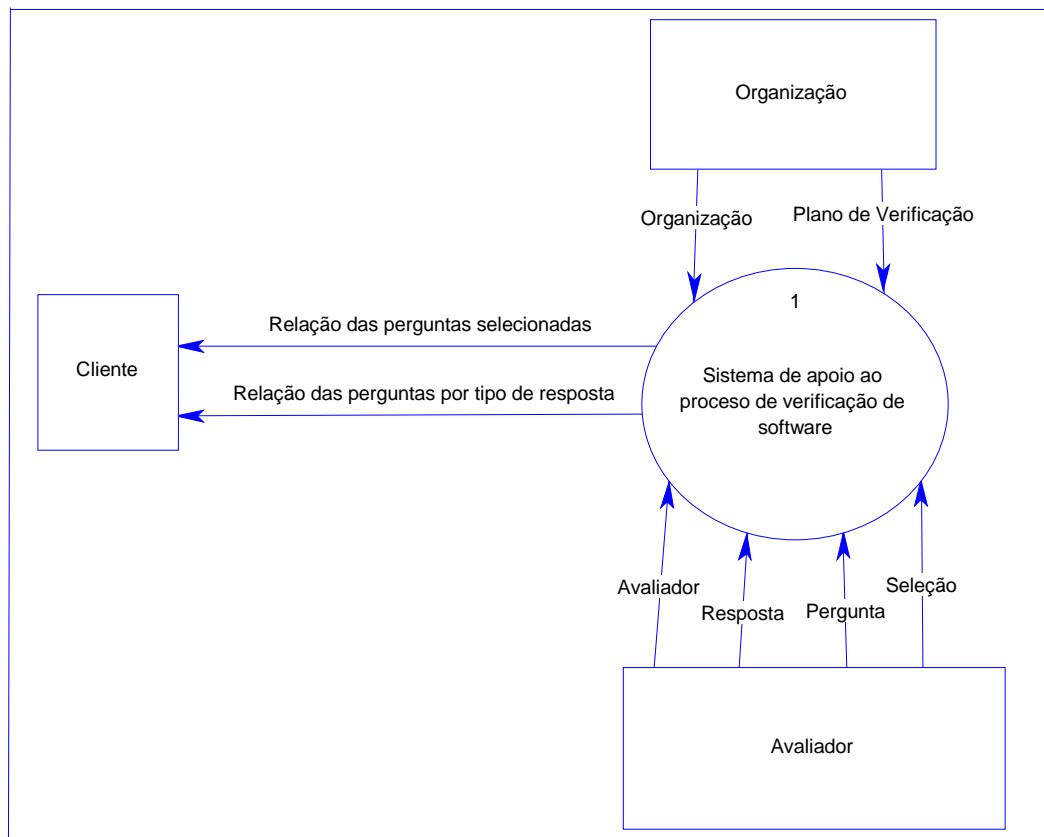
1. avaliador cadastra organização: neste evento o avaliador responsável ou quem estiver operando o software deverá realizar o cadastramento da organização que será avaliada;

2. avaliador é cadastrado: o cadastro do avaliador tem por objetivo manter o nome da pessoa responsável pela verificação do software;
3. é momento de efetuar pergunta: neste processo o avaliador do sistema irá proceder a avaliação da organização, relacionada a cada atividade específica anteriormente cadastrada;
4. avaliador registra plano de verificação: neste cadastro é possível a inclusão do software a ser verificado e neste processo o avaliador irá proceder também o cadastramento do plano de verificação;
5. é momento de selecionar pergunta: neste processo as perguntas correspondente a atividades que serão configuradas conforme o avaliador necessitar para a verificação do processo;
6. é momento de registrar resposta: neste momento o avaliador realizará a verificação do processo através do *checklist* para cada atividade escolhida.
7. cliente recebe relação das perguntas selecionadas: o cliente do sistema poderá emitir relatórios apresentando os resultados obtidos sobre as perguntas da atividade do contrato, processo, requisitos, projeto, código, integração e documentação;
8. cliente recebe relação das perguntas por tipo de resposta: o cliente do sistema poderá emitir um relatório sobre as perguntas atendidas versus não atendidas das atividades do processo;

### 4.3 DIAGRAMA DE CONTEXTO

O Diagrama de Contexto estabelece os limites entre o sistema e o seu ambiente. É utilizado para se obter uma visão macro do software, comunicações entre o sistema, o ambiente e as entidades com as quais se comunica. Na figura 2 é apresentado o Diagrama de Contexto do software. As ferramentas utilizadas para a especificação do software foi a ferramenta CASE *Power Designer 6.1* da *Sybase Inc.*

**FIGURA 2 – DIAGRAMA DE CONTEXTO**



#### 4.4 DIAGRAMA ENTIDADE-RELACIONAMENTO

O Diagrama Entidade-Relacionamento é um diagrama utilizado para detalhar as associações existentes entre as entidades de dados do sistema. Todas as entidades do software, como já mencionado se originaram de um estudo particular realizado com base nas normas utilizadas.

Para um melhor entendimento da representação dos modelos lógico e físico de dados das figuras 3 e 4, uma breve descrição de cada entidade é apresentada:

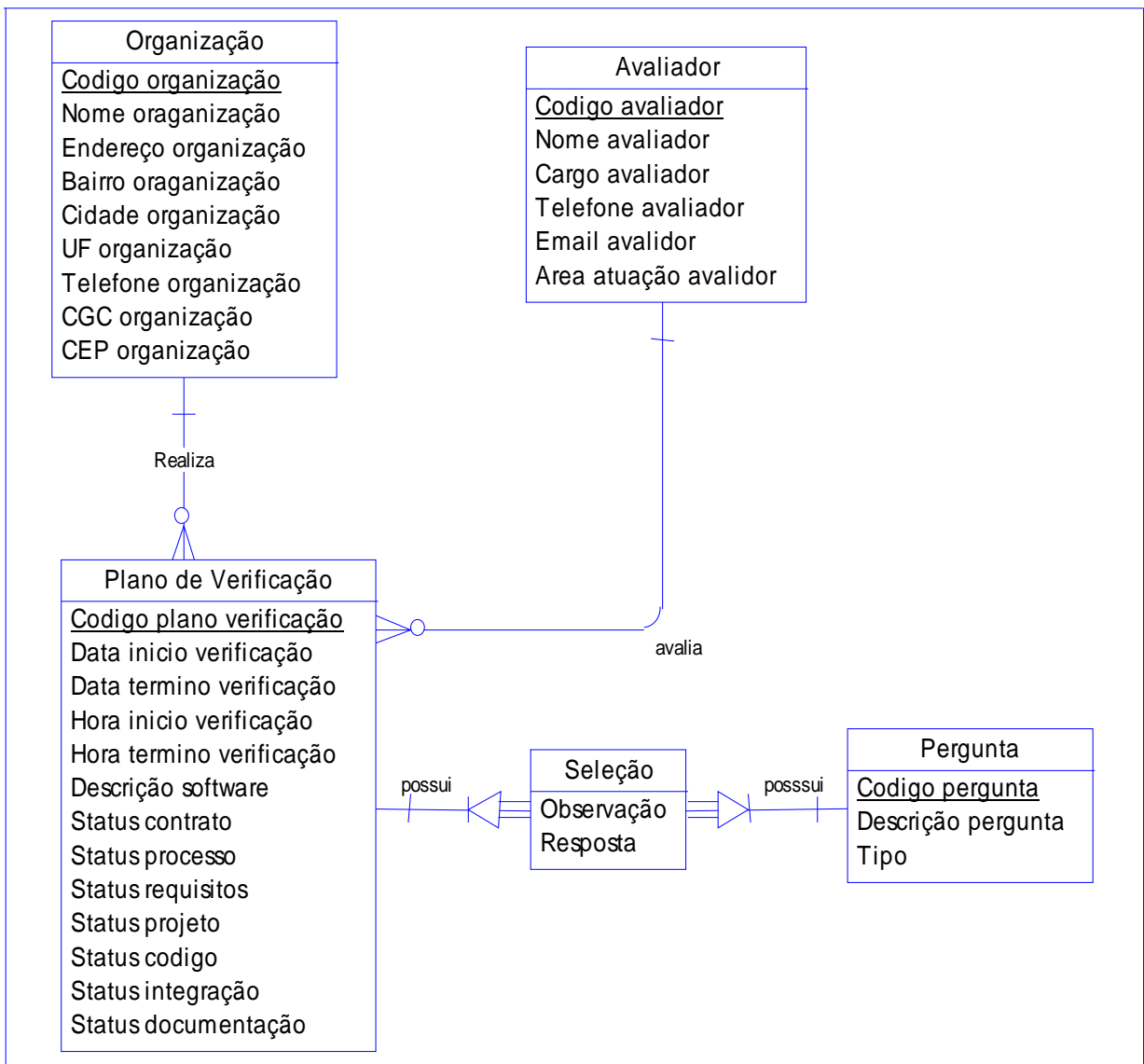
- organização: esta entidade é responsável por armazenar as organizações cadastradas no sistema;
- avaliador: esta entidade é responsável por armazenar os avaliadores cadastrados no sistema;
- plano de verificação: esta entidade é responsável por armazenar o plano de verificação das atividades cadastradas no sistema;



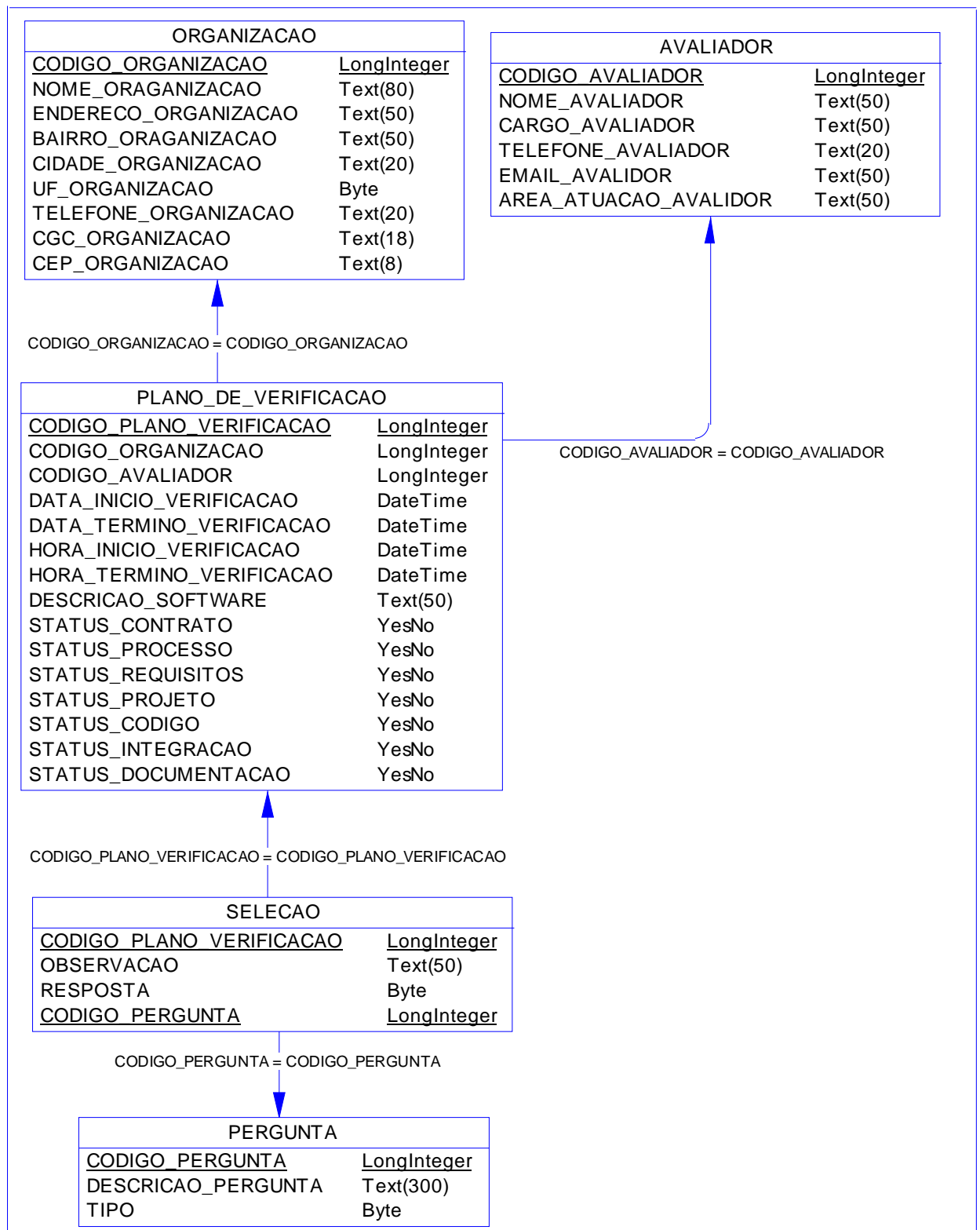
- d) pergunta: esta entidade é responsável por armazenar as perguntas propostas para avaliação cadastradas no sistema, relacionada a cada atividade específica;
- e) seleção: nesta entidade é selecionada a pergunta da atividade para realizar a verificação do *checklist*.

A figura 3 apresenta este diagrama no nível lógico e em seguida na figura 4 a nível físico.

**FIGURA 3 – DIAGRAMA DE ENTIDADE-RELACIONAMENTO – NÍVEL LÓGICO**



**FIGURA 4- DIAGRAMA DE ENTIDADE-RELACIONAMENTO – NÍVEL FÍSICO**

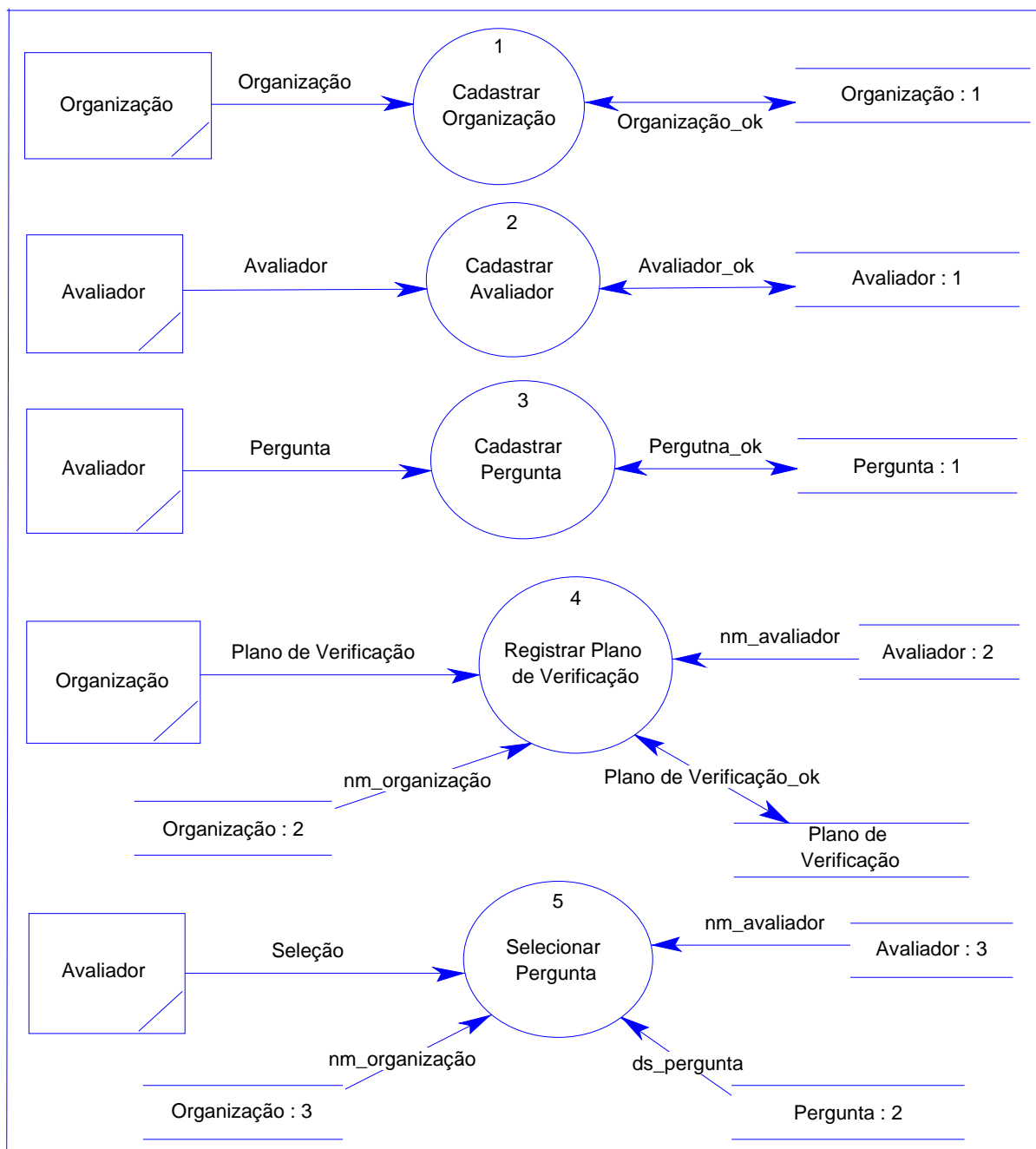


## 4.5 DIAGRAMA DE FLUXO DE DADOS

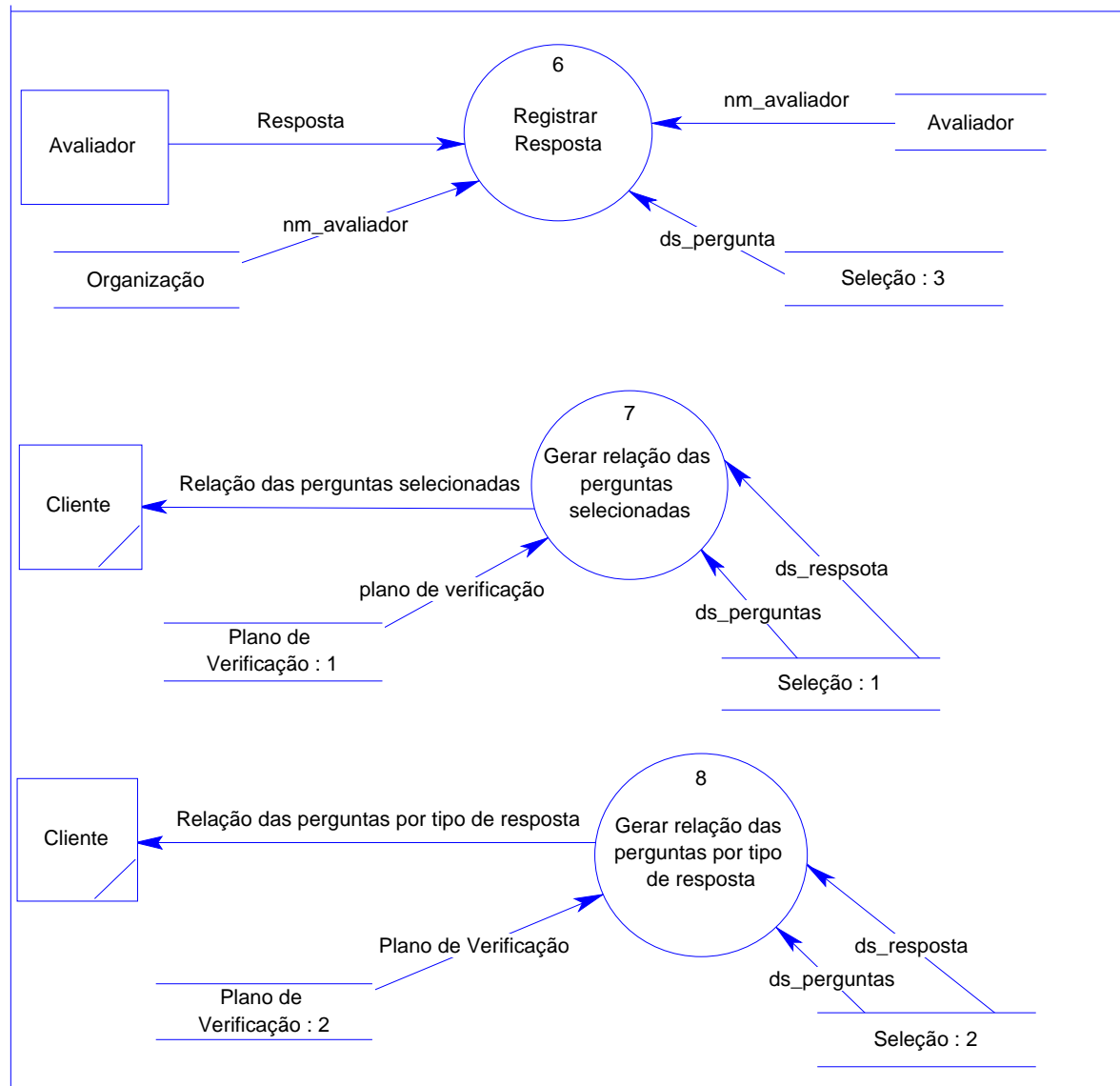
O objetivo do Diagrama de Fluxo de Dados é mostrar um sistema completo ou parte dele, de onde os dados surgem, para onde vão, quando são armazenados, que processos os transformam e as interações entre armazenamento de dados e processos.

Nas figuras 5 e 6 são apresentados os diagramas de fluxo de dados para cada evento conforme descrito no item 4.2 – Lista de Eventos:

**FIGURA 5 – DIAGRAMA DE FLUXO DE DADOS DE 1 A 5**



**FIGURA 6 – DIAGRAMA DE FLUXO DE DADOS DE 6 A 9**



## 4.6 DICIONÁRIO DE DADOS

O Dicionário de Dados do software foi gerado através da Ferramenta CASE *Power Designer*, que tem por objetivo fornecer suporte textual para complementar a informação mostrada no Diagrama Entidade-Relacionamento, sendo considerado um grupo organizado de definições, de todos os elementos de dados do sistema. A figura 7 apresenta este dicionário de dados.

Para a documentação do Dicionário de Dados é utilizado o seguinte formato:

- a) a coluna *Name* apresenta uma breve descrição do atributo;

- b) a coluna *Code* apresenta o nome que identifica o atributo na tabela;
- c) a coluna *Type* apresenta o tipo do atributo;
- d) a coluna *P* identifica se o atributo é chave primária da tabela;
- e) a coluna *M* identifica se é obrigatório o preenchimento do atributo.

**FIGURA 7 – DICIONÁRIO DE DADOS**

<b>Organização</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>P</b>	<b>M</b>
Código organização	CODIGO_ORGANIZACAO	LongInteger	Yes	Yes
Nome organização	NOME_ORAGANIZACAO	Text(80)	No	No
Endereço organização	ENDERECO_ORGANIZACAO	Text(50)	No	No
Bairro organização	BAIRRO_ORAGANIZACAO	Text(50)	No	No
Cidade organização	CIDADE_ORGANIZACAO	Text(20)	No	No
UF organização	UF_ORGANIZACAO	Byte	No	No
Telefone organização	TELEFONE_ORGANIZACAO	Text(20)	No	No
CGC organização	CGC_ORGANIZACAO	Text(18)	No	No
CEP organização	CEP_ORGANIZACAO	Text(8)	No	No
<b>Avaliador</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>P</b>	<b>M</b>
Código avaliador	CODIGO_AVALIADOR	LongInteger	Yes	Yes
Nome avaliador	NOME_AVALIADOR	Text(50)	No	No
Cargo avaliador	CARGO_AVALIADOR	Text(50)	No	No
Telefone avaliador	TELEFONE_AVALIADOR	Text(20)	No	No
Email avaliador	EMAIL_AVALIDOR	Text(50)	No	No
Area atuação avaliador	AREA_ATUACAO_AVALIDOR	Text(50)	No	No

FIGURA 7 – DICIONÁRIO DE DADOS (CONTINUAÇÃO)

<b>Plano de Verificação</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>P</b>	<b>M</b>
Codigo plano verificação	CODIGO_PLANO_VERIFICACAO	LongInteger	Yes	Yes
Codigo organização	CODIGO_ORGANIZACAO	LongInteger	No	Yes
Codigo avaliador	CODIGO_AVALIADOR	LongInteger	No	Yes
Data inicio verificação	DATA_INICIO_VERIFICACAO	DateTime	No	No
Data termino verificação	DATA_TERMINO_VERIFICACAO	DateTime	No	No
Hora inicio verificação	HORA_INICIO_VERIFICACAO	DateTime	No	No
Hora termino verificação	HORA_TERMINO_VERIFICACAO	DateTime	No	No
Descrição software	DESCRICAO_SOFTWARE	Text(50)	No	No
Status contrato	STATUS_CONTRATO	YesNo	No	No
Status processo	STATUS_PROCESSO	YesNo	No	No
Status requisitos	STATUS_REQUISITOS	YesNo	No	No
Status projeto	STATUS_PROJETO	YesNo	No	No
Status codigo	STATUS_CODIGO	YesNo	No	No
Status integração	STATUS_INTEGRACAO	YesNo	No	No
Status documentação	STATUS_DOCUMENTACAO	YesNo	No	No
<b>Pergunta</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>P</b>	<b>M</b>
Codigo pergunta	CODIGO_PERGUNTA	LongInteger	Yes	Yes
Descrição pergunta	DESCRICAO_PERGUNTA	Text(300)	No	No
Tipo	TIPO	Byte	No	No
<b>Seleção</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>P</b>	<b>M</b>
Codigo plano verificação	CODIGO_PLANO_VERIFICACAO	LongInteger	Yes	Yes
Observação	OBSERVACAO	Text(50)	No	No
Resposta	RESPOSTA	Byte	No	No
Codigo pergunta	CODIGO_PERGUNTA	LongInteger	Yes	Yes

## 5 IMPLEMENTAÇÃO DO PROTÓTIPO

Este software foi desenvolvido no ambiente de programação Visual Delphi 5.0 da Borland. Para o armazenamento dos dados foi utilizado o Paradox. A seguir será demonstrado passo a passo o uso do software

### 5.1 OPERACIONALIDADE DA IMPLEMENTAÇÃO

Ao executar o aplicativo, será apresentada a tela principal do software que é composto de cinco itens no menu principal, que dará acesso aos demais recursos do mesmo, tais como cadastro, configuração das perguntas, plano de verificação, relatórios e ajuda conforme a figura 8.

**FIGURA 8 – TELA INICIAL DO SOFTWARE DE APOIO AO PROCESSO DE VERIFICAÇÃO DE SOFTWARE**



Para iniciar o processo de verificação em si, primeiramente todos os cadastros devem ser preenchidos. Os cadastros existentes são demonstrados a seguir conforme figuras 9, 10, 11 e 12.

A figura 9 demonstra a tela de Cadastro da Organização, onde são informados os dados pessoais e jurídicos das organizações. Estas organizações aqui cadastradas serão utilizadas no registro do Plano de Verificação.

**FIGURA 9 – TELA DE CADASTRO DA ORGANIZAÇÃO**

**Cadastro Da Organização**

Codigo: 1

Nome: Empresa de Vendas de Produtos S.A

Endereco: Rua 7 de setembro, 95.756

Bairro: Centro

Cidade: Blumenau

Uf: SC

Cep: 89010-001

CGC: 78.025.451/0001-45

Fone: 047 328-1162

Buttons: Incluir, Salvar, Excluir, Cancelar, Finalizar

A figura 10 demonstra a tela de Cadastro do Avaliador que cadastra os dados pessoais. Este avaliador será utilizado posteriormente no registro do Plano de Verificação.

**FIGURA 10 – TELA DE CADASTRO DO AVALIADOR**

**Cadastro Do Avaliador**

Codigo: 2

Nome: Itaboraí Cordoni Ebertz

Area: Garantia da Qualidade

Cargo: Consultor de Software

Fone: 047 333-1505

Email: ita@inf.furb.br

Buttons: Incluir, Salvar, Excluir, Cancelar, Finalizar

Na figura 11 é apresentada a tela que permite o cadastramento das perguntas do *checklist* que compõem todas as atividades do processo de verificação da norma ISO/IEC 12207. Para cada atividade deverão ser cadastradas as perguntas específicas. Além do Cadastro de Pergunta do Contrato mostrado na figura 11, ainda existem o cadastro de perguntas para processo, requisitos, projeto, código, integração e documentação. Após o



avaliador entrar com a pergunta e clicar no botão Incluir, as perguntas serão armazenadas em uma lista que possibilitará a exclusão, alteração e verificação das perguntas já cadastradas.

O *checklist* foi criado a partir das informações encontradas em Mills (1994), Staa (2000), Freedman (1993), Menezes (2001), Rocha (2001), Pádua (2001), Longworth (1985), Fernandes (1995), Gil (1989), Anacleto (1996) e Junior (2000). Na tabela 2 são demonstradas as quantidades de perguntas criadas em cada atividade. Estão cadastradas no sistema para apoio ao processo de verificação 1160 perguntas para o *checklist* das atividades. No anexo são demonstradas algumas delas. A criação do *checklist* foi elaborado na busca das perguntas que descrevessem sobre o processo de verificação das atividades da norma ISO/ IEC 12207, onde algumas perguntas foram novamente elaboradas para seguir apenas um tipo de resposta e outras foram excluídas, pois existiam semelhança entre elas. Teve-se também a necessidade de verificar a onde cada uma das perguntas se encaixariam do processo de verificação das atividades da norma.

**TABELA 2 – QUANTIDADE DE PERGUNTAS**

ATIVIDADE	QUANTIDADE
Contrato	77
Processo	157
Requisitos	174
Projeto	165
Código	328
Integração	196
Documentação	140

**FIGURA 11 – TELA DE CADASTRO DAS PERGUNTAS DO CONTRATO**

Na figura 12, para iniciar o registro do Plano de Verificação, obrigatoriamente todos os outros cadastros já devem ter sido concluídos. O avaliador deverá executar os seguintes passos:

- a) clicar no botão Incluir;
- b) selecionar a organização a ser avaliada;
- c) informar a data da verificação;
- d) informar a hora da verificação;
- e) selecionar os avaliadores que executarão a verificação;
- f) selecionar as atividades que deverão ser acordados pelas partes envolvidas na verificação, só apenas aquelas atividades escolhidas irá ativar as telas correspondente a esta atividade.

O avaliador só vai poder passar para a próxima tela se alguns campos das atividades foram selecionadas.

**FIGURA 12 – TELA DE REGISTRO DO PLANO DE VERIFICAÇÃO**

Registro do plano de verificação

Codigo: 3

Nome Organização: Empresa de Vendas de Produtos S.A.

Data Inicio: 15/11/2002

Data Termino: 15/11/2002

Hora Inicio: 08:00:00

Hora Termino: 10:00:00

Descricao Software: Sistema de controle de vendas

Nome Avaliador: Itaborai Cordoni Ebertz

Contrato     Processo     Requisitos     Projeto

Codigo     Integração     Documentação

⏪ ⏩ ⏴ ⏵

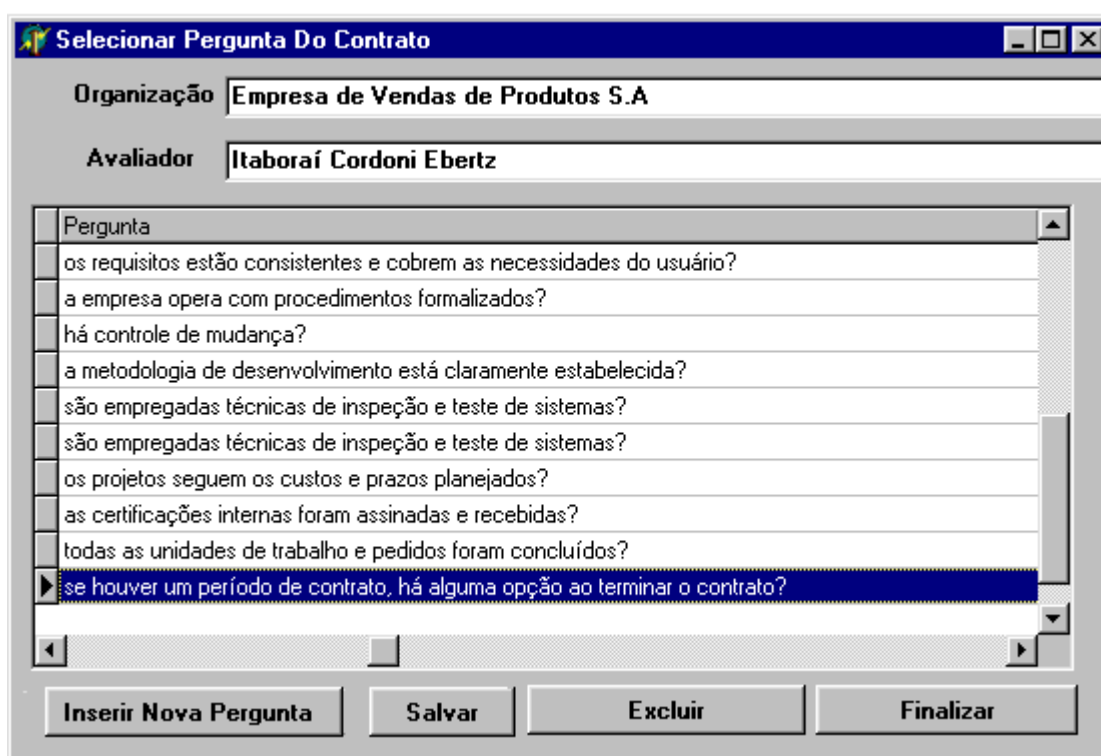
Incluir     Salvar     Excluir     Cancelar     Finalizar

Em seguida deverão ser selecionadas as questões propostas para avaliação do processo de verificação do software, com base na estrutura anteriormente informada para a norma

ISO/IEC 12207. Cada pergunta deverá estar relacionada a uma atividade específica do processo, a fim de verificar se a mesma está sendo contemplada ou não pela organização.

No menu Configuração de Perguntas só aparecerão as atividades ativas escolhidas na tela anterior do Cadastro do Plano de Verificação. Ao clicar na opção do menu ativado será aberto a tela de Seleção das Perguntas da Atividade do Contrato. Nesta tela será possível incluir apenas as perguntas requeridas pelo avaliador. Ao entrar nesta tela já será dado o nome da organização a ser avaliada e o nome do avaliador. Para inserir uma pergunta clique no botão Inserir Pergunta, conforme mostra a figura 13. De forma análoga acontecerá com as outras atividades selecionadas.

**FIGURA 13 – TELA DE SELEÇÃO DAS PERGUNTAS DA ATIVIDADE DO CONTRATO**



Na figura 14 é apresentado um trecho do código criado para tratar a inclusão da pergunta para a tela de Seleção de Pergunta.

FIGURA 14 – CÓDIGO FONTE DA IMPLEMENTAÇÃO

```
procedure TFrmSelePerCod.BitBtn3Click(Sender: TObject);  
begin  
FrmConResCod:=TFrmConResCod.Create(Self); {chama a tela de consulta}  
FrmConResCod.ShowModal;  
if FrmConResCod.Inserir then {inserir a pergunta da tabela de consulta}  
begin  
    TbRes.Append; {incluir pergunta selecionada}  
    TbRes.FieldName('Pergunta').asString:=  
        FrmConResCod.TbPer1.FieldName('Pergunta').asString;  
    TbRes.Post; {grava a pergunta na tela de seleção}  
end;  
FrmConResCod.Free; {fecha a tela de consulta}  
TbRes.Append;  
end;
```

Na figura 15 apresenta a tela de Consulta da Pergunta do Contrato. Nesta tela serão mostradas todas as perguntas cadastradas na tela cadastro da Pergunta do Contrato conforme já mostrado na figura 11. Esta tela é acionada ao clicar no botão Inserir Nova Pergunta da tela da Seleção da Pergunta da Atividade do Contrato. Para inserir uma pergunta na tela onde serão armazenadas as perguntas que irão ao *checklist* basta dar dois clique rápidos na pergunta escolhida.

**FIGURA 15 – TELA DA CONSULTA DAS PERGUNTAS DA ATIVIDADE DO CONTRATO**

Codigo	Pergunta
4	o fornecedor tem a capacidade de atender os requisitos?
5	os requisitos estão consistentes e cobrem as necessidades do usuário?
6	a empresa opera com procedimentos formalizados?
7	são empregados métodos de estimativas de custo, prazos e recursos?
8	os requisitos estão consistentes e cobrem as necessidades do usuário?
9	critérios e procedimentos de aceitação estão estipulados de acordo com os requisitos?
10	os resultados reais e desempenho do fornecedor contratado são acompanhados com base n
11	a metodologia de desenvolvimento está claramente estabelecida?
12	há controle quantitativo (estático) da qualidade do software?
13	são empregadas técnicas de inspeção e teste de sistemas?
14	o método de embalagem é controlado pelo fornecedor?
15	é assegurado que tudo o que foi prometido foi fornecido e foi devidamente aprovado?
16	as certificações internas foram assinadas e recebidas?
17	as contingências e riscos possíveis são identificados?
18	quanto aos contratos o escopo do contrato e os requerimentos são definidos e documentado
19	são estabelecidos e mantidos procedimentos para revisão de contrato e para a coordenação

Ao clicar no menu Plano de Verificação só estarão ativadas as atividades selecionadas na tela do Cadastro do Plano de Verificação e também só vai ser ativada se na tela de Seleção da Pergunta da Atividade foi selecionada alguma pergunta para esta atividade. Efetuando-se os passos anteriormente apresentados é possível realizar as respectivas verificações do processo da atividade de software da organização.

Além da tela Verificação do Contrato mostrada na figura 18, existem no menu Plano de Verificação uma tela de verificação para cada atividade: Verificação do Processo, Verificação dos Requisitos, Verificação do Projeto, Verificação do Código, Verificação da Integração e Verificação da Documentação.

Conforme a figura 16, a tela da Verificação do Contrato já vem com o nome da organização e do avaliador, logo abaixo serão enviadas as perguntas selecionadas, para fazer a verificação basta clicar em uma das opções de resposta: Atende, Não Atende ou Desconhece/ Não Sabe. Para seguir o *checklist* basta clicar no botão próximo.

FIGURA 16 – TELA DA VERIFICAÇÃO DO CONTRATO

**Verificação Do Contrato**

**Organização** Empresa de Vendas de Produtos S.A

**Avaliador** Itaboraí Cordoni Ebertz

**Pergunta**  
o fornecedor tem a capacidade de atender os requisitos?

**Resposta**

- Atende**
- Não Atende**
- Desconhece/Não Sabe**

**Iniciar Checklist**

**Finalizar**

**Observação**

Finalmente, para as verificações já efetuadas, poderá ser emitido um relatório apresentando os resultados deste procedimento, onde poderão ser verificadas as respostas fornecidas em cada avaliação. Este relatório permite efetuar uma relação de todas as perguntas atendidas e suas respectivas respostas referente a uma determinada atividade verificadas. Serão mostrados também o nome do avaliador, datas da verificação, horas da verificação, descrição do software e nome da organização.

No menu Relatórios, pode-se visualizar ou emitir uma relação de metas concluídas, basta clicar na opção da atividade que você queira realizar a impressão. Conforme figura 17, nestes relatórios o cliente poderá ter uma relação de todas as perguntas selecionadas e suas respectivas respostas referente as Atividade do Contrato, Atividade do Processo, Atividade dos Requisitos, Atividade do Projeto, Atividade do Código, Atividade da Integração e Atividade da Documentação.

FIGURA 17 – RELATÓRIO DAS PERGUNTAS DA ATIVIDADE DO CONTRATO

**Relatorio do Contrato**

**Nome Organização:** Empresa de Vendas de Produtos S.A  
**Nome Avaliador:** Itaboraí Cordoni Ebertz  
**Data Inicio Plano:** 15/11/2002      **Hora Inicio Plano:** 08:00:00  
**Data Termin Plano:** 15/11/2002      **Hora termino Plano:** 10:00:00  
**Descrição do software:** Sistema de Controles de Vendas S.A

Resposta	Pergunta
Atende	há controle quantitativo (estático) da qualidade do software?
Atende	o fornecedor tem a capacidade de atender os requisitos?
Atende	o método de embalagem é controlado pelo fornecedor?
Atende	os requisitos estão consistentes e cobrem as necessidades do usuário?
Atende	são empregadas técnicas de inspeção e teste de sistemas?
Atende	são empregadas técnicas de inspeção e teste de sistemas?
Atende	a empresa opera com procedimentos formalizados?
Não Atende	são empregados métodos de estimativas de custo, prazos e recursos?
Não Atende	é assegurado que tudo o que foi prometido foi fornecido e foi devidamente aprovado?
Não Atende	critérios e procedimentos de aceitação estão estipulados de acordo com os requisitos?
Não Sabe	a metodologia de desenvolvimento está claramente estabelecida?
Não Sabe	as certificações internas foram assinadas e recebidas?

Page 1 of 1

No menu Relatórios, pode-se ainda visualizar ou emitir uma relação de perguntas por tipo de resposta, isto é, perguntas que satisfazem as exigências dos padrões versus aquelas que não satisfazem. Conforme figura 18, nestes relatórios o cliente poderá ter uma relação de todas as perguntas por grau de atendimento.

FIGURA 18 – RELATÓRIO DAS RESPOSTAS ATENDIDAS DO CONTRATO

**Relatorio das resposta atendidas do contrato**

**Nome Organização:** Empresa de Vendas de Produtos S.A  
**Nome Avaliador:** Itaboraí Cordoni Ebertz  
**Data Inicio Plano:** 15/11/2002      **Hora Inicio Plano:** 08:00:00  
**Data Termin Plano:** 15/11/2002      **Hora termino Plano:** 08:00:00  
**Descrição do software:** Sistema de Controles de Vendas S.A

Resposta	Pergunta
Atende	há controle quantitativo (estático) da qualidade do software?
Atende	o fornecedor tem a capacidade de atender os requisitos?
Atende	o método de embalagem é controlado pelo fornecedor?
Atende	os requisitos estão consistentes e cobrem as necessidades do usuário?
Atende	são empregadas técnicas de inspeção e teste de sistemas?
Atende	são empregadas técnicas de Inspeção e teste de sistemas?
Atende	a empresa opera com procedimentos formalizados?
Atende	quanto aos contratos o escopo do contrato e os requerimentos são definidos e documentados?
Atende	se houver um período de contrato, há alguma opção ao terminar o contrato?
Atende	as informações proprietárias são adequadamente protegidas?
Atende	existem produtos pendentes?
Atende	se houver um período de contrato, há alguma opção ao terminar o contrato?

Page 1 of 1

De acordo com o roteiro da figura 1, o sistema de apoio ao processo de verificação de software consegue atender todas as atividades do roteiro através de um *checklist* específico para cada uma das áreas de verificação. Seguindo o roteiro foram criados os cadastrados da organização e do avaliador para a verificação de um software, além do plano de verificação que faz o gerenciamento de todo o ciclo de vida do produto, gerenciando as atividade, perguntas e os relatórios.



## 6 CONCLUSÕES

O objetivo inicialmente proposto foi realizado, visto que o software implementado consegue atender ao conjunto de atividades presentes no processo de verificação de software que são a verificação do contrato, requisito, projeto, código, integração e documentação. Assim o software especificado e implementado mostrou-se eficaz no auxílio ao processo de verificação de softwares. Este tipo de verificação ajuda os desenvolvedores não cometerem erros em novas aplicações com *feedback* de problemas passados para auxiliá-los em dificuldades futuras. Além disso, o software pode ser usado como guia para a realização de auditorias.

Uma das principais dificuldades deste trabalho foi a construção do *checklist* para o processo de verificação. Isto porque necessitou-se de uma ampla pesquisa para encontrar as perguntas para cada atividade do processo de verificação da norma ISO/IEC 12207, exigiu a necessidade de verificar se existiam perguntas iguais ou semelhantes e outra dificuldade foi a necessidade de formular novamente algumas perguntas pois deveriam seguir apenas um tipo de resposta.

A atividade de verificação é essencial ao desenvolvimento de software, pois procura garantir o desenvolvimento de produtos de software de alta qualidade e de baixo custo com base em um processo de software bem estabelecido, com alta qualidade e produtividade. A qualidade de software é determinada pela qualidade dos processos utilizados para o desenvolvimento. Deste modo, a melhoria da qualidade de software é obtida pela melhoria da qualidade dos processos.

Após o estudo das normas, chegou-se a conclusão que a norma ISO/IEC 12207 é a mais completa, relativa ao processo de verificação de software por ter se mostrado mais abrangente no aspecto da verificação de sistemas. As outras normas estudadas ISO 9000-3, ISO/IEC 15504 e o modelo CMMI foram utilizadas no comparativo pelo motivo de que boa parte de seus processos são contemplados pela norma ISO/IEC 12207. Na criação das perguntas do *checklist* foi bastante útil para o comparativo. O processo proposto tomou como base a norma ISO/IEC 12207 para auxiliar o processo de verificação de software como pôde-se observar no decorrer deste estudo. A norma, apesar de flexível quanto ao uso, sem o uso de um roteiro de implantação, exige um alto grau de dificuldade para sua adoção. A proposta

deste trabalho de criar um software para o apoio ao processo de verificação baseou-se no roteiro de Frare (1998), que facilitou a implantação e a compreensão da mesma.

Atualmente as normas e modelo não possuem nenhum tipo de questionário, por isso, optou-se pela utilização de *checklist*, conseguindo assim facilitar o processo de verificação. A base para identificação das atividades foram realizadas pela norma ISO/IEC 12207. Apesar de extenso, o *checklist* está sujeito a pequenas correções no que diz respeito a melhor formulação do questionamento ou até sua adequada classificação. O software permite a inclusão e alteração das perguntas o que torna o processo mais dinâmico.

É de fundamental importância para a adoção da proposta apresentada, o uso do *checklist* disponível no anexo. Eles foram especificados para abranger todas as atividades da norma ISO/IEC 12207 contempladas na proposta, garantindo dessa maneira uma verificação em todo o ciclo de vida do software.

Como sugestão para trabalhos futuros o software poderia ser testado em várias empresas que desejam realizar a verificação de software, visando uma maior validação do trabalho.

Uma outra sugestão seria a inclusão do suporte ao processo de revisão e inspeção de software que completaria o processo de verificação.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 12207**: tecnologia de informação - processos do ciclo de vida do software. Rio de Janeiro, 1998.

ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO 9000-3**: Normas de gestão da qualidade e garantia da qualidade - diretrizes para aplicação da NBR 19001 desenvolvimento, fornecimento e manutenção de software. Rio de Janeiro, 1993.

ANACLETO, Ana Lúcia. **Mensuração do processo de software baseado no modelo CMM/SEI**. 1999. 93 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

BARBARESCO, A. Eduardo. **Software de apoio ao processo de gerencia da configuração segundo normas e modelos da qualidade**. 2000. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

BELLOQUIM, Átila. **Qualidade de Software: o que há de novo no mercado?**. Revista Developers. Rio de Janeiro, n. 68, p. 11, abr. 2002.

BIZZOTO, Carlos E. Negrão. **Influência da utilização de metodologia de desenvolvimento sobre a qualidade do software: um enfoque quantitativo**. 1992. 139 f. Dissertação (Mestrado em Engenharia da Produção) – Departamento de Engenharia de Produção e Sistemas, Universidade Federal de Santa Catarina, Florianópolis.

BENTO, S. Deisy. **Software de apoio ao processo de aquisição segundo normas de qualidade**. 2000. 49 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

DUARTE, S. Alexandre. **Software de apoio ao processo de documentação baseado em normas de qualidade**. 2000. 66 f. Trabalho de Conclusão de Curso (Bacharelado em

Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

EMAM, Khaled El. **Spice**: the theory and practice of software process improvement and capability determination. Los Alamitos, Califórnia: IEE Computer Society, 1998.

FERNANDES, Aguinaldo Aragon. **Gerência de software através de métricas**: garantindo a qualidade do projeto, processo e produto. São Paulo: Atlas, 1995.

FRARE, Alexandre. **Proposta de roteiro de implantação da norma internacional ISO/IEC 12207 – processos do ciclo de vida de software**. 1998. 97 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

FREEDMAN, P. Alexandre; GERALD M. Weinberg. **Manual de walkthroughs**: inspeções e revisões técnicas de especificações de sistemas e programas. São Paulo: Makron Books, 1993.

GIL, Antonio de Loureiro. **Auditoria de computadores**. São Paulo: Atlas, 1989.

IAHN, Anísio. **Avaliação de processos de software utilizando a norma ISO/IEC 15504**. 1999. 43 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

INTHURN, Cândida. **Qualidade & teste de software**. Florianópolis: Visual Books, 2001.

JUNCKES, A. Fabio. **Software de apoio ao processo de auditoria segundo normas de qualidade**. 1999. 66 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

JUNIOR, Uno T. **Análise de uma organização de software utilizando o modelo CMMI/SEI V 1.0**. 2000. 92 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

KRAUSE, Conrad. **Análise de uma metodologia de desenvolvimento de sistemas baseada na norma ISO/IEC 12207**. 1998. 47 f. Trabalho de Conclusão de Curso (Bacharelado em

Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

LONGWORTH, G. **Padrões em programação: métodos e procedimentos.** Rio de Janeiro: Campus, 1985.

MENEZES, Luís César de Moura. **Gestão de projetos.** São Paulo: Atlas, 2001.

MILLS, A. Charles. **A auditoria da qualidade: uma ferramenta para a avaliação constante e sistemática da manutenção da qualidade.** São Paulo: Makron Books, 1994.

PÁDUA, Wilson P. Filho. **Engenharia de software: fundamentos, métodos, padrões.** Rio de Janeiro: LTC, 2001.

POMPILHO, S. **Análise essencial.** Rio de Janeiro: Infobook, 1994.

PRESSMAN, Roger S. **Engenharia de software.** São Paulo: Makron Books, 1995.

ROCHA, Ana R. Cavalcanti da; MALDONADO, C. José; WEBER C. Kival. **Qualidade de software: teoria e prática.** São Paulo: Prentice hall, 2001.

SOMMERVILLE, Ian. **Software engineering.** Harlow: Addison-Wesley, 2001.

STAA, Von Arndt. **Programação modular: desenvolvendo programas complexos de forma organizada e segura.** Rio de Janeiro: Campus, 2000.

## **ANEXO - CHECKLIST**

### **VERIFICAÇÃO DO CONTRATO**

1. o fornecedor tem a capacidade de atender os requisitos?
2. os requisitos estão consistentes e cobrem as necessidades do usuário?
3. procedimentos adequados, para tratar alterações nos requisitos e priorização de problemas, estão estipulados?
4. procedimentos e sua abrangência para interação e cooperação entre as partes são estipulados, incluindo propriedade, garantia, direitos autorais e confidencialidade?
5. critérios e procedimentos de aceitação estão estipulados de acordo com os requisitos?
6. a empresa opera com procedimentos formalizados?
7. um procedimento documentado é utilizado para estabelecer contratos com fornecedores externos para aquisição de seus produtos ou serviços?
8. Os resultados reais e desempenho do fornecedor contratado são acompanhados com base nos compromissos assumidos no projeto do software?
9. são empregados métodos de estimativas de custo, prazos e recursos?
10. há controle de mudança?
11. a metodologia de desenvolvimento está claramente estabelecida?
12. são empregadas técnicas de inspeção e teste de sistemas?
13. há controle quantitativo (estático) da qualidade do software?
14. os projetos seguem os custos e prazos planejados?
15. seus novos fornecedores são selecionados?
16. são estabelecidos os termos e as condições do contrato? por exemplo, planos de inspeção de produto ou contrato, procedimentos, dados de contrato, outros.
17. o método de embalagem é controlado pelo contrato? por exemplo, informação do contrato, plano contratual de garantia da qualidade, plano de contrato de produção, outros.
18. o método de embalagem é controlado pelo fornecedor?
19. o fornecedor dispõem de procedimentos e instruções estabelecidos, documentados e atualizados para verificar o projeto do produto em questão, para assegurar que os requisitos são cumpridos?
20. é assegurado que tudo o que foi prometido foi fornecido e foi devidamente aprovado?

21. existem produtos pendente?
22. as certificações internas foram assinadas e recebidas?
23. existem comprometimentos internos pendente?
24. todas as unidades de trabalho e pedidos foram concluídos?
25. se houver um período de contrato, há alguma opção ao terminar o contrato?
26. a identificação do item de software propicia o relacionamento com os requisitos estabelecidos em contrato?
27. são estabelecidos e mantidos procedimentos para revisão de contrato e para a coordenação destas atividades?
28. quanto aos contratos o escopo do contrato e os requerimentos são definidos e documentados?
29. as contingências e riscos possíveis são identificados?
30. as informações proprietárias são adequadamente protegidas?
31. os requerimentos que diferem daqueles na proposta são resolvidos?
32. as responsabilidades com subcontratos estão definidas?
33. a terminologia a ser empregada é entendida por ambas as partes?
34. em relação aos itens da qualidade no contrato estão definidos os critérios de aceitação?
35. está definido o tratamento de mudança nos requisitos do cliente durante o desenvolvimento?
36. está definido o tratamento de problemas detectados após a aceitação do produto/serviço, incluindo reclamações e queixas do cliente acerca de itens relacionados com a qualidade?
37. as atividades a serem desempenhadas pelo cliente, especialmente às relativas a especificação de requisitos, instalação e aceitação estão definidos?
38. as facilidades, ferramentas e itens de software a serem entregues ao cliente estão definidas?
39. os padrões e procedimentos a serem utilizados estão definidos?
40. os requisitos para replicação do software estão definidos?
41. quanto requerido por contrato, há procedimentos para o tratamento após a entrega do produto?
42. há procedimentos para assegurar que produtos ou serviços adquiridos para agregar ao produto estejam em conformidade com os requisitos?
43. há registros sobre a qualificação de desempenho de subcontratados?

45. de ordens de compra ou subcontratos para assegurar que as seções de trabalho responsáveis pela verificação dos bens ou serviços sendo comprados estão conforme o padrão?
46. o presente padrão tem por objetivos assegurar a existência e facilitar a manutenção de informações gerenciais relativas aos módulos e demais arquivos que compõem programas?
47. é assegurado a existência e a facilitação da manutenção de especificações detalhadas relativas aos diferentes componentes de um módulo?
48. os módulos tenham um estilo de documentação consistente, uniforme e independente do autor?
49. a uma facilidade a modificação de programas redigidos por outrem, sem que isto acarrete perda de qualidade?
50. o projeto interno as unidades de trabalho incompletas foram documentadas e justificadas?
51. a gerência foi notificada quanto à disponibilidade do pessoal do projeto?
52. a gerência foi notificada quanto à disponibilidade das instalações do projeto?
53. o plano do projeto foi arquivado com todos os dados de suporte?
54. os excedentes do projeto foi arquivado com todos os dados de suporte?
55. os excedentes de material do projeto foram administrados?
56. o projeto externo foi obtido acordo com o proprietário do projeto sobre a disposição dos produtos restantes?
57. as certificações e autorizações externas foram assinadas e aprovadas?
58. os fornecedores foram notificados quanto a compromissos pendentes?
59. estão todas as partes cientes do encerramento pendente?
60. as instalações do projeto foram fechadas?
61. os procedimentos de auditoria e manutenção foram conduzidos?
62. o pessoal interno as preocupações da equipe do projeto referentes a empregos futuros foram abordados?
63. a equipe está dedicada a manter os compromissos restantes do projeto?
64. ainda existem fatores de motivação presentes para as tarefas e obrigações restantes?
65. as preocupações referentes à identidade da equipe foram abordadas?
66. o pessoal foi recolocado ou notificado da metodologia de realocação?
67. o pessoal externo estão sendo feitos esforços para assegurar que o interesse do contratante permaneça atendido?



68. estão sendo feitos esforços para assegurar que as atitudes e percepções do contratante referentes ao projeto permaneçam estáveis?
69. as questões de transferência de pessoal estão sendo abordadas com o contratante do projeto?
70. o pessoal chave e o contratante estão sendo notificados sobre os status do projeto?
71. existe uma metodologia de comunicação para manter as relações entre o contratante e os gerentes do projeto (seu e do contratado)?
72. o fornecedor desempenha revisões e assegura que os requisitos sejam atendidos e que os métodos de projeto e implementação sejam adequadamente utilizados?
73. o fornecedor mantém registros documentados de tais revisões?
74. antes das atividades de aceitação o cliente é auxiliado na determinação de prazo?
75. o cliente é auxiliado na determinação dos procedimentos de avaliação?
76. o cliente é auxiliado na determinação do ambiente de hardware e software necessário para a avaliação de aceitação?
77. o cliente é auxiliado na definição dos critérios de aceitação?

## **VERIFICAÇÃO DO PROCESSO**

1. os requisitos de planejamento do projeto estão adequados e oportunos?
2. os processos selecionados para o projeto estão adequados, implementados, sendo executados como planejados e conforme o contrato?
3. os padrões, procedimentos e ambientes para os processos do projeto estão adequados?
4. o projeto dispõe de equipe e pessoal capacitado, como requerido no contrato?
5. o escopo do software é definido e delimitado sem ambigüidades?
6. as atividades de revisões são planejadas?
7. a atividade de prevenção de defeitos são planejadas?
8. uma vez identificadas, as causas comuns de defeitos são priorizadas e eliminadas sistematicamente?
9. a terminologia é clara?
10. os recursos são adequados para o escopo?
11. os recursos estão prontamente disponíveis?
12. os riscos em todas as categorias importantes foram definidos?
13. um plano de gerenciamento dos riscos está em andamento?
14. os participantes das revisões recebem o treinamento necessário para executar tais tarefas?
15. você conduz os testes como uma atividade sistemática e organizada, segundo uma metodologia bem definida e planejada?

16. os métodos e instrumentos de teste são bem compreendidos e utilizados nos momentos certos?
17. você usa controle de configuração de software?
18. você revisa formalmente cada contribuição de projeto de software antes de se comprometer?
19. você estima formalmente o tamanho, esforço e custos do software?
20. você planeja formalmente agendas de software?
21. você mede o tamanho e a complexibilidade de cada módulo de software, no decorrer do tempo?
22. você mede e rastreia os erros/defeitos através da manutenção do ciclo de vida?
23. você usa um mecanismo para controlar as alterações nas exigências do software?
24. você usa um mecanismo para controlar quem altera o código e quando?
25. você usa um processo de manutenção de software padronizado e documentado em cada projeto?
26. existe um programa de treinamento de engenharia de software exigido para os profissionais de software?
27. existe um programa de formal de treinamento para os líderes de revisão de projetos e codificação?
28. as coberturas de revisão de código e projeto são medidas e registradas?
29. a cobertura de testes é medida e registrada em cada fase de testes funcionais?
30. as análises de erros são conduzidas para determinar suas causas relacionadas ao processo?
31. as tarefas são definidas e colocadas em seqüência adequadamente? o paralelismo é razoável, dados os recursos disponíveis?
32. a base para a estimativa de custos é razoável? a estimativa de custos foi desenvolvida usando-se dois métodos independentes?
33. foram usados dados de produtividade e de qualidade históricos?
34. as diferenças de estimativas foram conciliadas?
35. os orçamentos e prazos finais predefinidos são realísticos?
36. o cronograma é consistente?
37. existe uma política documentada, definindo os objetivos e o compromisso para com a qualidade?
38. esta política é compreendida, implementada e mantida em todos os níveis da organização?
39. estão definidas as responsabilidades e autoridades de todo o pessoal que administra, desempenha e verifica atividades que influem na qualidade?
40. os requisitos, recursos e pessoal treinado para atividades de verificação estão identificados?
41. o sistema da qualidade é analisado criticamente pela administração em intervalos apropriados?
42. são mantidos registros destas análises?
43. o comprador quando da contratação dos serviços designa um representante com autoridade para lidar com os aspectos contratuais?
44. revisões conjuntas com o cliente são realizadas a intervalos programados para verificar conformidades do software com os requisitos, verificar resultados de teste de aceitação?
45. os resultados de tais revisões são documentadas?

46. uma política da empresa é estabelecida, requerendo que cada novo produto ou release seja medido?
47. uma política da empresa é estabelecida, requerendo que todos os projetos estabeleçam planos para aperfeiçoar os métodos-padrões?
48. a ênfase organizacional é sobre o planejamento e acompanhamento da qualidade?
49. recursos são disponível para apoiar a introdução de tecnologia?
50. cada grupo de desenvolvimento principal conduz, periodicamente, um trabalho de avaliação?
51. revisões gerenciais periódicas são realizadas para avaliar o desempenho em termos dos planos de qualidade e ações de melhoria da qualidade?
52. são oferecidos cursos padrões sobre planejamento da qualidade, gerência quantitativa de processo, métodos avançados de desenvolvimento, prototipação.
53. planos de qualidade são produzidos para cada projeto?
54. planos de ação visando à melhoria da qualidade são produzidos sempre quando o projeto não atinge as metas da qualidade?
55. o desempenho é medido em função dos planos de qualidade e melhorias?
56. as customizações de cada projeto em relação aos padrões do processo são retidas sob o controle da gerência de configuração de software?
57. as métricas utilizadas no processo são mantidas sob controle da gerência de configuração de software?
58. métricas de qualidade dos subcontratados são estabelecidas e acompanhadas?
59. o desempenho da área de garantia da qualidade é acompanhado e revisado?
60. padrões documentados são produzidos para inspeções, ferramentas e métodos, planos de qualidade e monitorando de qualidade por atividade do processo?
61. procedimentos documentados são produzidos para a customização do processo e do ambiente?
62. procedimentos documentados são desenvolvidos para os planos de qualidade, acompanhamento, inspeções e customização do processo e do ambiente?
63. métodos documentados são estabelecidos para a prototipação e avaliação quantitativa do projeto?
64. protótipos são desenvolvidos para demonstrar a viabilidade de requisitos críticos antes que entrem no escopo do projeto?
65. medições e análise são feitas sobre os níveis de defeitos nos produtos, eficiência e cobertura das inspeções e testes, distribuições dos erros, produtividade de tarefas e eficácia das ferramentas?
66. um banco de dados sobre o processo é estabelecido?
67. um banco de dados sobre o processo e o sistema de qualidade são centralmente mantidos e protegidos?

68. mecanismos e responsabilidades são estabelecidos para assegurar a definição e o uso de métricas padrões, a produção e acompanhamento dos planos de qualidade, e a inserção apropriada de novas tecnologias?
69. mecanismos e responsabilidades são definidos para assegurar que os profissionais técnicos dos projetos aprovem e aceitem os planos de qualidade dos produtos e tornem-se pessoalmente comprometidos em mantê-lo?
70. apoio à inserção de tecnologias é estabelecido, incluindo apoio para a instalação e uso de ferramentas, assistência à utilização de novos métodos, bem como à customização do ambiente?
71. o procedimento de planejamento de projeto exige a identificação de quem é o responsável por cada atividade de projeto e desenvolvimento?
72. os procedimentos exigem que o planejamento do projeto seja atualizado à medida que o projeto evolui?
73. nessa segunda fase a preocupação central é com a estruturação e viabilização operacional do projeto. Nela, a proposta de trabalho, já aprovada, é detalhada por meio de um plano de execução operacional?
74. detalhamento das metas e objetivos a serem alcançados, são com base na proposta aprovada?
75. há alguma definição do gerente do projeto?
76. há um detalhamento das atividades e estruturação analítica do projeto?
77. a programação das atividades no tempo disponível ou necessário?
78. a determinação dos resultados tangíveis a são alcançados durante a execução do projeto?
79. a programação da utilização e provisionamento dos recursos humanos e materiais necessários ao gerenciamento e à execução do projeto?
80. delineamento do procedimentos de acompanhamento e controle a serem utilizados na implantação do projeto?
81. estabelecimento e estrutura orgânica formal a ser utilizada para o projeto?
82. a alguma estruturação do sistema de comunicação e de decisão a ser adotado?
83. designação e comprometimento dos técnicos que participarão do projeto?
84. existe treinamento do envolvidos com o projeto?
85. o plano de desenvolvimento deve conter a definição do projeto em termos de objetivos e relacionamento com outros projetos do comprador ou do fornecedor?
86. a organização dos recursos para o projeto inclui estrutura da equipe, responsabilidades, uso de subcontratados e recursos materiais?
87. as fases de desenvolvimento é utilizado o cronograma do projeto e a identificação de planos correlatos, como plano da qualidade, plano de gerência de configuração, plano de integração e plano de teste?

88. o plano de desenvolvimento é atualizado na medida do progresso do projeto?
89. para cada fase do desenvolvimento é estabelecidas as entradas e saídas?
90. o plano prever também como o projeto será gerenciado?
91. o plano identifica as regras, práticas, convenções, ferramentas, técnicas e a gerência de configuração?
92. o fornecedor verifica as saídas de todas as fases do desenvolvimento?
93. o processo de planejamento é monitorado com base em padrões e procedimentos aprovados pela área de garantia da qualidade?
94. métodos padrões são desenvolvidos para estimativa e planejamento de projeto?
95. medições e análise são feitas entre o planejamento e o executado?
96. a ênfase organizacional é sobre o planejamento e acompanhamento da qualidade?
97. procedimentos documentados são desenvolvidos para o planejamento da produtividade, geração de códigos reutilizáveis, customização de componentes reutilizáveis?
98. o plano de desenvolvimento contempla a definição do projeto, incluindo a declaração de seus objetivos, bem como referencia o relacionamento com outros projetos da empresa e do cliente?
99. a organização dos recursos do projeto, incluindo a estrutura da equipe, responsabilidades, utilização de subcontratos, recursos materiais e outros?
100. o cronograma e calendarização do projeto, as tarefas, recursos e esforços requerido para cada tarefa e qualquer inter-relacionamento entre elas?
101. a identificação dos planos relacionados, como: plano da qualidade, plano de gerência de configuração, plano de integração, plano de teste?
102. o plano de desenvolvimento é atualizado na medida em que o projeto avança?
103. antes do início de cada fase há uma revisão e aprovação dos produtos da fase anterior?
104. há uma metodologia claramente definida para o desenvolvimento do produto?
105. nesta metodologia há a identificação das fases de desenvolvimento a serem executadas?
106. há identificação das entradas para cada fase?
107. há identificação das saídas requeridas de cada fase?
108. há procedimentos de verificação a serem realizados em cada fase?
109. há análise de problemas potenciais associados com as fases de desenvolvimento e com a construção dos requisitos?
110. o plano de desenvolvimento estabelece como os projetos devem ser gerenciados, incluindo a identificação de cronograma de desenvolvimento e implementação, os produtos a serem entregues, o

- controle do progresso, as responsabilidades organizacionais, os recursos e atribuição de tarefas à equipe do projeto, bem como os interfaces técnicos e organizacionais entre diferentes grupos?
- 111.o plano de desenvolvimento identifica as regras, práticas e convenções par o desenvolvimento, as ferramentas e técnicas e a gerência de configuração?
  - 112.há revisões de progresso planejadas?
  - 113.as revisões de progresso são documentadas?
  - 114.as entradas para cada fase de desenvolvimento são definidas e documentadas?
  - 115.cada requisito pode ser verificado quanto a sua consecução?
  - 116.as saídas requeridas de cada fase de desenvolvimento são definidas e documentadas?
  - 117.há um plano para a execução de verificações dos produtos de cada fase de desenvolvimento?
  - 118.os resultados das verificações são registrados?
  - 119.para o desenvolvimento do projeto é preparado um plano da qualidade específico?
  - 120.o plano da qualidade contempla objetivos da qualidade expressos em termos mensuráveis?
  - 121.existem critérios de entrada e saída para cada fase de desenvolvimento?
  - 122.a identificação dos tipos de teste, bem como atividades de verificação e validação a serem desempenhadas?
  - 123.existem planejamento detalhado de teste, atividades de verificação e validação a serem realizadas, definindo cronograma, recursos e autoridade de aprovação?
  - 124.existem responsabilidades para revisões e testes, gerência de configuração, controle de mudança, controle de defeitos e ação corretiva?
  - 125.os documentos de planejamento descrevem a evolução do relacionamento com o cliente?
  - 126.os operadores recebem treinamento em relação aos métodos e normas de trabalho?
  - 127.no treinamento o exemplo apresenta as principais funções do produto?
  - 128.o restante do exemplo é coerente com o que já foi aprendido?
  - 129.o exemplo é coerente com os padrões?
  - 130.o exemplo introduz nova terminologia? essa terminologia é definida antes do exemplo? ela é necessária?
  - 131.o aluno conseguiria reproduzir o exemplo com o conhecimento autal, ou o exemplo requer conhecimento especial (por exemplo, procedimentos de sistema operacional)?
  - 132.o exemplo está correto?
  - 133.o exemplo é relevante?

- 134.o exemplo encontra-se disposto na mesma seqüência que os materiais?
- 135.os pontos principais do exemplo são reforçados adequada e corretamente?
- 136.o exemplo é inteiramente coerente?
- 137.os pontos principais do exemplo estão diretamente relacionados com uma necessidade do mundo real?  
em caso afirmativo, como? em caso negativo, por que não?
- 138.o método de apresentação é coerente com nossos métodos de reprodução? se for necessário o uso de cores, dispomos de recursos de reprodução em cores?
- 139.o método de apresentação é coerente com nossos métodos de projeção (35mm, 16mm, super 8, filme contínuo, videocassete, projetor de transparência, projetos de slides)?
- 140.o método de apresentação faz amplo uso dos recursos de edição?
- 141.ele usa recursos de edição apenas por usar ou para melhorar o produto?
- 142.a notação e os símbolos estão de acordo com as práticas atuais?
- 143.ele é coerente com nossos sistemas de computador onde necessário?
- 144.o método de representação é consistente com o design do materiais (a instrução auto-ajustável é realmente auto-ajustável ou precisa ser programada)?
- 145.a forma do produto melhora ou reduz seu valor educacional?
- 146.é dado algum tipo de treinamento?
- 147.a qualidade do treinamento e do programa de instalação é requerido?
- 148.necessita-se de pré-requisitos?
- 149.as despesas quanto o treinamento são em separado?
- 150.no treinamento o fornecedor estabelece e mantém procedimentos para identificar as necessidades de treinamento e prover o treinamento para todo o pessoal que desempenha atividades que afetam a qualidade?
- 151.o fornecedor deve mantêm registros apropriados sobre a realização de treinamento?
- 152.as revisões gerenciais periódicas são realizadas para tratar sobre treinamento, inserção de tecnologia, status do processo de software e planos de aperfeiçoamento do processo?
- 153.um plano de treinamento é produzido, definindo os recursos requerimentos para cada função no ambiente de desenvolvimento?
- 154.o pessoal requerido e requisitos de treinamento?
- 155.há procedimentos para identificar as necessidades de treinamento do pessoal envolvido com atividades que afetam a qualidade?

156. são mantidos registros sobre as atividades de treinamento?

157. os procedimentos exigem adequadamente a designação de pessoal qualificado, apoiado pelos recursos necessários?

## **VERIFICAÇÃO DOS REQUISITOS**

1. os requisitos do sistema são consistentes, viáveis e testáveis?
2. os requisitos do sistema foram distribuídos apropriadamente para os itens de hardware, itens de software e operações manuais, de acordo com os critérios do projeto?
3. os requisitos de software são consistentes, viáveis, testáveis e refletem precisamente os requisitos do sistema?
4. os requisitos de software relacionados à proteção, à segurança e aos fatores críticos estão corretos, conforme demonstrado por métodos adequadamente rigorosos?
5. as fases de teste importantes foram adequadamente identificadas e dispostas seqüencialmente?
6. a capacidade de acompanhar os critérios/requisitos de validação foi estabelecida como parte da análise de requisitos de software?
7. sua organização desenvolve e mantém um processo padronizado do software?
8. a organização coleta, revisa e torna disponíveis informações relacionadas ao uso do processo padronizado de software da organização (por exemplo, estimativas e dados reais de tamanho de software, esforços e custos, dados de produtividade, e avaliações de qualidade)?
9. as ações, associadas a defeitos identificados nos produtos intermediários de software durante as revisões, são acompanhadas até que os defeitos sejam resolvidos?
10. as atividades executadas para desenvolver e melhorar o processo de software são revisadas periodicamente pela gerência superior?
11. medidas são usadas para determinar a funcionalidade e qualidade dos produtos de software (por exemplo, quantidade, tipos e gravidade dos defeitos identificados)?
12. medidas são usadas para determinar o estado das atividades de revisão (por exemplo, número de revisões executadas, esforços dispendidos nestas revisões e o número de produtos intermediários revisados comparados com o plano)?
13. as atividades de revisão e produtos intermediários são submetidos a auditoriais e revisões da Garantia da Qualidade de software (por exemplo, revisões planejadas são conduzidas e ações posteriores são acompanhadas)?
14. medidas são usadas para determinar o estado das atividades de prevenção de defeitos (por exemplo, o tempo e custo para identificar e corrigir defeitos e a quantidade de ações propostas e completadas)?
15. quando os requisitos definidos para o software mudam, são feitos ajustes nos planos, produtos de software e atividades relacionadas?
16. você faz revisões periódicas dos planos e registros de teste, submetendo-os à aprovação de outros profissionais?



17. os dados de revisão de projetos são analisados para avaliar o produto e reduzir defeitos futuros?
18. as principais funções são logo demonstradas?
19. o plano de testes é consistente com o plano de projeto global?
20. um cronograma de teste foi explicitamente definido?
21. os recursos e ferramentas de teste foram identificados e es disposição?
22. um mecanismo de manutenção de registros de teste foi estabelecido?
23. a organização utiliza-se de um plano definido para conduzir a verificação dos componentes em relação aos seus requisitos especificados?
24. a organização provê um procedimento para que um exame metódico de artefatos de software selecionados seja efetuado por parte de parceiros do produtor, de maneira a identificar defeitos e áreas onde são necessárias correções ou melhorias?
25. os *drivers* de teste foram identificados e o trabalho para desenvolvê-los foi planejado?
26. o *teste de fadiga* para o software foi especificado?
27. procedimento de teste foram especificados tanto testes de caixa branca como de caixa preta?
28. foram testados todos os caminhos lógicos independentes?
29. foram identificados e listados casos de teste com seus resultados esperados?
30. o tratamento de erros será testado?
31. os valores-limites serão testados?
32. a temporização e o desempenho serão testados?
33. foi especificada uma variação aceitável dos resultados esperados?
34. há auditorias internas da qualidade para verificar a eficácia do sistema da qualidade?
35. as auditorias são programadas com base na importância das atividades relativas à qualidade?
36. as auditorias e ações de acompanhamento são desempenhadas de acordo com procedimentos documentados?
37. os resultados das auditorias são documentados e levados à atenção do pessoal que tem responsabilidade pela área auditada?
38. na ação corretiva é mantida documentação e procedimentos para investigar as causas de não-conformidades do produto e a ação corretiva necessária para prevenir a repetição?
39. são analisados todos os processos, operações de trabalho, concessões, registros da qualidade, relatórios de assistência técnica e reclamações do cliente para detectar e eliminar causas potenciais de serviços?
40. é aplicado ações preventivas para tratar dos problemas em um nível correspondentes aos riscos encontrados?
41. são executados controles para assegurar que as ações corretivas são realizadas e que as mesmas são efetivas?
42. ao implementar e registrar alterações nos procedimentos resultantes de ações corretivas?

43. há utilização de métricas para a medição da qualidade do produto?
44. as métricas relativas ao produto são coletadas em intervalos regulares?
45. são identificados os níveis de desempenho com as métricas?
46. com os resultados das medições, e quanto há indicação, há tomada de ação corretiva?
47. são estabelecidas metas de melhoria em função das medições?
48. há métricas para a medição da qualidade do processo de desenvolvimento de software?
49. software produto incluso: há procedimentos para a validação, armazenamento, proteção e manutenção de produto de software do cliente ou de terceiros?
50. são feitas testes de unidade em relação de faltas e deficiências encontradas, do ponto de vista das especificações?
51. há alguma relação de faltas e deficiências encontradas, do ponto de vista dos padrões de programação?
52. há alguma relação de faltas e deficiências encontradas pelas ferramentas de apoio ao desenvolvimento?
53. existe alguma definição dos casos de teste?
54. os resultados obtidos ao executar os testes são verificados?.
55. são feitos laudos de avaliação dos testes?
56. são realizados relatórios de problemas indicando o estado do problema, a causa do problema e a solução dada?
57. este relatório é cumulativo e descreve a história de evolução do módulo?
58. é assegurado que cada programa é adequado ao usuário?
59. é assegurado que cada programa opera exatamente conforme a sua especificação?
60. é assegurado que o conjunto de programas interagem conforme especificação?
61. são verificadas as implementações de soluções?
62. são controladas os processamento subsequente, entrega e instalação do produto não-conforme, até que a deficiência ou condição insatisfatória tenha sido corrigida?
63. a identificação documentada dos requisitos de verificação e provisão dos recursos necessários, são incluído ao pessoal treinado para todas as atividades de verificação?
64. existe independência documentada daqueles que realizam atividades de verificação em relação aos que têm responsabilidade direta pelo trabalho sendo realizado?
65. existe independência documentada daqueles que realizam atividades de verificação em relação aos que têm responsabilidade direta pelo trabalho sendo realizado?

66. a implementação do sistema da qualidade é organizada, documentada e verificada com organogramas, descrições de cargos/tarefas, supervisão ou auditorias da qualidade?
67. todos os requisitos importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo à interface externa foram incluído?
68. está definida a resposta do software para todas as possíveis situações de entrada de dados?
69. todas as seções estão conforme a especificação de requisitos?
70. todas as referências de figuras, tabelas, e diagramas estão contempladas?
71. as informações no artefato de software estão de acordo com as informações presentes na especificação de requisitos ou ao conhecimento geral do domínio?
72. todos os requisitos descrevem fato que é verdadeiro, estão de acordo as condições solicitadas para o sistema?
73. os diagrama de projeto contém as representações do conceito descrito nos requisitos gerais do domínio ou no documento de requisitos?
74. a representação de um conceito dos requisitos gerais do domínio ou do documento dos requisitos estão presente em todos dos diagramas de projeto?
75. as informações em uma parte do artefato de software estão inconsistentes com outras no artefato de software?
76. nenhum requisitos entra em conflito com outro?
77. uma representação de um conceito em um diagrama de projeto está em acordo com a representação do mesmo conceito no mesmo ou em outro diagrama de projeto?
78. existe alguma informação no artefato de software que é ambígua, isto é, é possível ao desenvolvedor interpretar as informações de diferentes maneiras, podendo não levar a uma implementação correta?
79. existe algum requisito que tenha várias interpretações devido a diferentes termos utilizados para uma mesma característica ou vários significados de um termo para um contexto em particular?
80. as representação dos conceitos no projeto está clara e não causam má interpretação ou entendimento errado do seu significado por parte do usuário do documento?
81. as informações dos requisitos que são fornecidas, são necessárias e são mesmo usadas?
82. resultados da codificação produzem o código de cada programa e geram as tabelas de dados utilizados pelos programas? (por exemplo, tabelas de mensagens)
83. são produzido as definições dos recursos utilizados pelo programa? (por exemplo, diálogos, menus, ícones).
84. são produzidos os textos de auxílio (help)?

85. são produzidos os tutoriais?
86. são realizados teste de unidade em relação de faltas e deficiências encontradas, do ponto de vista das especificações?
87. são realizados teste de unidade em relação de faltas e deficiências encontradas, do ponto de vista dos padrões de programação?
88. são realizados teste de unidade em relação de faltas e deficiências encontradas pelas ferramentas de apoio ao desenvolvimento?
89. são feitos definição dos casos de teste?
90. existe algum método os dados de teste escolhidos para cada caso de teste?
91. ao ter os resultados obtidos ao executar os testes são construídos laudos de avaliação dos testes?
92. relatórios de problemas indicando o estado do problema, a causa do problema e a solução dada. Este relatório é cumulado e descreve a história de evolução do módulo?
93. é assegurado que cada programa é adequado ao usuário?
94. é assegurado que cada programa opera exatamente conforme a sua especificação?
95. o conjunto de programas interagem conforme especificação?
96. a documentação técnica e de uso (auxílios, manuais, tutoriais) existem, está completa e é consistente como o que se encontra implementado?
97. existe alguma preparação aos testes: alocação de tarefas à equipe de testes?
98. todas as condições foram testadas, identificadas e documentadas?
99. são feitas criação dos arquivos de testes para execução de todas as funções do sistema?
100. a seqüência de arquivos estão corretas?
101. estabelecimento de valores iniciais são significativos?
102. todos os tipos de registros estão incluídos?
103. o volume de arquivo é grande o suficiente para demonstrar grupos de controles, totais, quebras de páginas nos relatórios?
104. estabelecimento de período é significativo?
105. as transações de testes desenvolvidas demonstra todos os casos testes?
106. critérios de testes das unidades está satisfeitos?
107. critérios de testes do sistema está satisfeitos?
108. existem transações criadas para pelo menos dois ciclos de execução do sistema?

109. são demonstradas opções do sistema de fim de período (mês, ano)?
110. os resultados dos testes preditos e documentados para comparação posterior com os resultados do computador?
111. os resultados paralelos do sistema estão identificados para padrão de precisão?
112. padrões de aceitabilidade e precisão estão estabelecidos e documentados para posterior avaliação dos resultados?
113. a disponibilidade de memória em modo de produção?
114. a limitações de tempo em modo de produção?
115. transações de testes são alimentadas pelo departamento de entrada de dados utilizando procedimentos padrões?
116. testes executados pelo pessoal de operação tem supervisão do programador?
117. a uma listagem de transações obtida antes da execução dos testes?
118. *dumps* de arquivo são obtidos antes da execução dos programas?
119. existem algum sistema que é executado para demonstrar todos os casos de testes? (podem ser necessárias execuções múltiplas do sistema ou de módulos individuais.)
120. há obtenção de todos os relatórios de controle via *hard copy*?
121. há obtenção da transição de mensagens da console (e das informações de tempo)?
122. há *dumps* de arquivo extraídos após a execução dos testes?
123. há disponibilidade dos resultados de execução paralelas?
124. o tempo de execução aceitável de acordo co padrões preestabelecidos?
125. há memória suficiente disponível para execução e expansão moderada do tamanho de programas?
126. os dados de entrada aceitos são formatados e sem erros?
127. todos os caminhos lógicos estão executados corretamente?
128. todos os dados inválidos foram detectados?
129. combinações de transações inconsistentes foram identificadas?
130. mensagens de edição foram testadas?
131. condições de não correspondência ou *invalid key* foram testadas?
132. execução de situação de ausência de arquivo existe situação de arquivo vazio?
133. atualização de arquivos precisa e completa de acordo com as especificações?
134. processamento correto do primeiro registro?

- 135.inclusão de registros antes da leitura do primeiro registro esta feita corretamente?
- 136.lógica de correspondência entre arquivos esta funcionando?
- 137.adição de registros após processamento do último registro esta feita corretamente?
- 138.os campos estão atualizados conforme especificados: substituição de quantidades, adição de qualidade etc.
- 139.eliminação ou adição de registro estão executada corretamente?
- 140.transações múltiplas por registro do arquivo mestre estão executadas corretamente?
- 141.alterações nas chaves estão tratadas de forma precisa?
- 142.todas as quebras dos relatórios estão demonstradas?
- 143.existe quebras de controle de detalhes, pequenos e grandes, e quebra final de controle?
- 144.os arquivos estão criados de acordo com as especificações do projeto?
- 145.os tamanhos de campos nos arquivos: estão adequados e não há truncamento inesperado ou perda de dígitos significativos?
- 146.existe precisão matemática, arredondamento?
- 147.relatórios estão de acordo com o *layout* do projeto?
- 148.relatórios estão de acordo com o *layout* do projeto?
- 149.a numeração de páginas do relatório esta precisa?
- 150.a ortografia dos títulos e dos cabeçalhos está correta?
- 151.tamanhos de campos estão adequados nos relatórios?
- 152.dados para auditoria (totais de controle) estão precisos?
- 153.os tempo de execução aceitável de acordo com padrões preestabelecidos?
- 154.a situação de ausência de arquivo-vazio são testadas em todos os passos do sistema?
- 155.existe meios disponíveis de verificação cruzada dos resultados do sistema exaustivamente testados?
- 156.são realizados relatórios diferentes com as mesmas informações ou informações derivadas?
- 157.todas as interface estão funcionando de modo preciso, entre programas?
- 158.todas as interface estão funcionando de modo preciso, entre subsistemas?
- 159.todas as interface estão funcionando de modo preciso, entre outro sistemas?
- 160.todas as interface estão funcionando de modo preciso, com sistemas manuais existentes?
- 161.a seqüência de execução (fluxo do sistema) está precisa?

- 162. *loops* de procedimentos de correção de erros estão funcionando?
- 163. o sistema reinicia os procedimentos de modo satisfatório.?
- 164. procedimentos de recuperação de arquivos satisfatórios?
- 165. avaliação de aceitabilidade de acordo com padrões predeterminados?
- 166. serviços oferecidos pelo sistema de acordo com as definições originais dos requisitos do usuário?
- 167. modificações nos requisitos do usuário a partir da definição original?
- 168. teste do sistema julgado completo? aprovação dos resultados pelo usuário?
- 169. os padrões podem ser revisados separadamente em conformidade com as especificações?
- 170. código e documentação podem às vezes ser revisados separadamente?
- 171. a eficiência pode ser revisada separadamente a partir de outras especificações?
- 172. a interface com o usuário pode ser revisada separadamente?
- 173. a manutenibilidade pode ser revisada separadamente da conformidade quanto às especificações funcionais?
- 174. a conveniência de operação pode constituir um item separado de revisão?

## **VERIFICAÇÃO DO PROJETO**

1. o projeto está correto e consistente com os requisitos e rastreável aos mesmos?
2. o projeto implementa uma seqüência adequada de eventos, entradas, resultados, interfaces, fluxo lógico, alocação de tempo e de orçamentos e definição, isolamento e recuperação de erro?
3. o projeto selecionado pode ser originado a partir dos requisitos?
4. o projeto implementa projeção, segurança e outros requisitos críticos corretamente, conforme demonstrado por métodos adequadamente rigorosos?
5. os requisitos de software se refletem na arquitetura de software?
6. os grupos de engenharia identificam, acompanham e resolvem problemas entre os grupos (por exemplo, incompatibilidade de cronogramas, riscos técnicos ou problemas a nível de sistema)?
7. o projeto conduz reuniões para análise de causas para identificar causas comuns dos defeitos?
8. é conseguida uma efetiva modularidade?
9. os módulos são funcionalmente independentes?
10. as atividades para desenvolvimento e melhoria dos processos de softwares dos projetos e da organização são coordenadas através da própria organização (por exemplo, via um grupo de processo de engenharia de software)?
11. a gerência revisa formalmente os status dos projetos de software?

12. você conduz revisões/inspeções internas do projeto de software?
13. os itens resultantes das ações de revisão de projeto são feitos integralmente?
14. os itens das ações resultantes das revisões de código são feitos integralmente?
15. a arquitetura de programa é fatorada?
16. são definidas interfaces para os módulos e para os elementos de sistema externos?
17. a estrutura de dados é consistente com o domínio de informação?
18. a estrutura de dados é consistente com os requisitos de software?
19. a manutenibilidade foi levada em consideração?
20. para o *walkthrough* de projeto o algoritmo executa a função desejada?
21. o algoritmo está logicamente correto?
22. a interface é consistente com o projeto arquitetural?
23. a complexidade lógica é razoável?
24. o tratamento de erros e o *antibugging* foram especificados?
25. as estruturas de dados locais foram adequadamente definidas?
26. as construções de programação estruturadas são usadas do princípio ao fim?
27. os detalhes de projeto são adequados à linguagem de implementação?
28. existem particularidades usadas de sistema operacional ou dependentes de linguagem?
29. é usada uma lógica composta ou inversa?
30. a manutenibilidade foi levada em consideração?
31. o domínio de informação é completo, consistente e acurado?
32. a divisão do problema em partições é completa?
33. as interfaces externas e internas são adequadamente definidas?
34. o modelo de dados reflete adequadamente os objetos de dados, seus atributos e relações?
35. todos os requisitos são rastreáveis em nível de sistema?
36. a prototipação foi levada a efeito pelo cliente/usuário?
37. o desempenho é atingível, dentro das restrições impostas por outros elementos do sistema?
38. os requisitos têm consistência com os prazos, os recursos e o orçamento?
39. os critérios de validação estão completos?
40. existe política escrita, divulgada e documentada, requerendo que o padrão do processo de desenvolvimento de software seja usado?
41. é estabelecido um grupo de engenharia do processo de software?
42. revisões gerenciais periódicas são realizadas para verificar o status dos projetos internos de aperfeiçoamento do processo de software?
43. o grupo de engenharia do processo de software lidera as atividades de aperfeiçoamento do processo, bem como assegura a conscientização sobre os métodos empregados?
44. mecanismos são estabelecidos para identificar problemas sobre projeto de sistema e software e resolvê-los imediatamente?



45. cursos padrões são oferecidos sobre gerência da qualidade, gerência de profissionais de software, métodos de software e inspeções?
46. os recursos são estimativos para cada atividade-chave no processo de software?
47. contingências são estabelecidas para todas estimativas conforme a experiência histórica?
48. planos de gerência de risco dos projetos identificam termos técnicos e de negócio, os quais possam afetar o sucesso do projeto?
49. o desempenho do projeto é acompanhado e revisado por atividade-chave no processo de software?
50. inspeções de projeto e código são monitorados até a resolução dos problemas que surjam?
51. o desenvolvimento e documentação dos padrões e métodos do processo são monitorados e revistos?
52. as ferramentas e métodos de desenvolvimento de cada projeto são mantidos sob controle da gerência de configuração de software?
53. as definições e padrões do processo para cada projeto são mantidas sob controle da gerência de configuração de software?
54. procedimentos formais são estabelecidos para a subcontratação de recursos?
55. mecanismos e recursos são estabelecidos para o acompanhamento do desempenho dos subcontratados?
56. a área de garantia da qualidade da empresa monitora a garantia da qualidade do subcontratado?
57. padrões documentados são produzidos para o processo de software como um todo, abrangendo também registros de custos, métodos de teste e relatórios sobre qualidade dos produtos?
58. guias de estilo para desenvolvimento são utilizados por toda a empresa e para qualquer projeto?
59. procedimentos documentados são estabelecidos para produzir e aprovar padrões de processo e produto?
60. as definições do processo são adaptadas para, dinamicamente, atender as contingências de cada projeto?
61. métodos documentados são desenvolvidos para o projeto, codificação, inspeção e teste?
62. métodos são estabelecidos para identificar e relacionar todos os itens referentes ao projeto e codificação, desde os requisitos até o teste?
63. medições são feitas sobre os erros encontrados e os custos incorridos pela atividade do processo?
64. mecanismos e responsabilidades são estabelecidos para assegurar que cada requisito do produto seja testável?
65. mecanismos e responsabilidades são estabelecidos para assegurar a produção de padrões do processo, bem como relatórios de custos?
66. mecanismos são estabelecidos para assegurar que os padrões do processo de software sejam inteiramente revistos e que sua adoção seja aceita pela comunidade técnica da empresa?

67. mecanismos e responsabilidades são estabelecidos para a revisão da eficácia do grupo de engenharia do processo de software?
68. mecanismos do processo são estabelecidos para assegurar a validação da abordagem do projeto contra as necessidades dos usuários?
69. um plano de inserção de tecnologia é estabelecido?
70. o ambiente de desenvolvimento de software é estabelecido e instalado?
71. um conjunto de padrões de ferramentas compatíveis é definido e está disponível?
72. os projetos de desenvolvimento de software são organizados de acordo com um roteiro metodológico que represente um modelo de ciclo de vida?
73. as atividades relacionadas com a qualidade são planejadas e implementadas com relação ao modelo de ciclo de vida usados?
74. nas especificação dos requisitos do cliente os requisitos do cliente abrangem itens como desempenho, segurança, confiabilidade e privacidade?
75. estes requisitos são claramente definidos de forma a serem validados durante a aceitação do produto?
76. há aprovação destes requisitos por parte do cliente antes do desenvolvimento do produto?
77. há controle da documentação relativa aos requisitos do cliente?
78. todas as interfaces entre produto de software e outros software ou produtos de hardware são claramente estabelecidos na especificação de requisitos?
79. durante o desenvolvimento da especificação dos requisitos: há designação de pessoas de ambas as partes para estabelecer a especificação de requisitos do cliente?
80. durante o desenvolvimento da especificação dos requisitos: há formas e métodos para aprovação das especificações e para mudanças nas especificações?
81. o esforço de definição de requisitos é documentado por ambas as partes?
82. projeto e implementação no projeto de desenvolvimento é levado em consideração as regras de projeto, definições de interfaces internas?
83. projeto e implementação no projeto de desenvolvimento é levado em consideração a metodologia específica tendo em vista o tipo de software que está sendo construído?
84. projeto e implementação no projeto de desenvolvimento é levado em consideração a utilização de experiência passada para evitar a ocorrência de problemas similares?
85. projeto e implementação no projeto de desenvolvimento é levado em consideração a construção das especificações de forma que possam ser facilmente testadas e mantidas?
86. na implementação é levado em consideração as regras de programação, linguagens de programação, convenções de identificação de atributos, tabelas, índices, arquivos?

87. na implementação é levado em consideração metodologias de implementação?
88. há revisões para assegurar que os requisitos estão sendo atingidos e que os métodos de projeto e implementação estão sendo empregados?
89. estas revisões impedem o prosseguimento do projeto antes que as deficiências sejam efetivamente sanadas?
90. a verificação deve incluir os programas correspondem ao tamanho e desempenho estimados?
91. o rendimento é satisfatório, pode ser aperfeiçoado posteriormente?
92. houve alguma alteração no número de transações ou no tamanho dos arquivos?
93. a saída está de acordo com as necessidades?
94. existem pontos ásperos que possam ser suavizados?
95. é realizada alguma ação de emergência de revisão?
96. todos os itens necessários para a especificação de requisitos para a solução do problema foram incluídos?
97. todo item constante da especificação de requisitos encontram-se livre de erros?
98. cada item da especificação de requisitos é exato e não-vago? existe uma única interpretação de cada item da especificação de requisitos, o significado de cada item é compreendido e a especificação é fácil de ser lida?
99. cada item da especificação de requisitos é exato e não-vago? existe uma única interpretação de cada item da especificação de requisitos, o significado de cada item é compreendido e a especificação é fácil de ser lida?
100. todo item da especificação de requisitos é pertinente ao problema e sua solução?
101. durante o desenvolvimento de programas e testes de aceitação, será possível determinar se o item da especificação foi satisfeito?
102. cada item da especificação de requisitos pode ser rastreado até sua origem no ambiente do problema?
103. cada item da especificação de requisitos pode ser implementado com as técnicas, ferramentas, recursos e pessoal disponíveis dentro do custo estimado e limitações do cronograma?
104. as especificações de requisitos são uma declaração dos requisitos que devem ser satisfeitas pela solução do problema, e não obscurecidas pelas soluções propostas para o problema?
105. as especificações de requisitos são expressas de maneira que cada item possa ser modificados sem um impacto excessivo sobre os demais itens?
106. você se preocupou com as pessoas que terão de usar o sistema?
107. você se preocupou com as pessoas que terão de operar o sistema?

108. você se preocupou com as pessoas que terão de manter o sistema?
109. se você fosse uma dessas pessoas, você alteraria algo no projeto para facilitar sua vida?
110. todos os elementos da especificação deixam clara a sua intenção?
111. foram tiradas as redundâncias da especificação (diferentes elementos especificando a mesma coisa)?
112. todos os elementos estão contendo itens de especificação relacionados com a sua intenção?
113. todos os elementos cuja intenção seja completamente descrita pelos respectivos itens de especificação?
114. todos os itens de especificação de algum elemento acrescentam algo ao conjunto dos demais itens de especificação desse elemento?
115. é possível adicionar itens à especificação, tornando mais completa a intenção do elemento?
116. o escopo de um item de especificação é o menor possível?
117. existem funções retornando referência ou ponteiro para espaço de dados local?
118. todos os elementos inflexíveis são necessariamente utilizados?
119. todos os elementos reutilizáveis são necessariamente utilizados?
120. todos os elementos da especificação são adequados ao usuário?
121. a verificação está envolvida em qualquer um dos ciclos de controle do projeto e da produção? por exemplo, revisão do projeto, revisão da qualidade, outros?
122. como as especificações de confiabilidade, manutenibilidade e disponibilidade são administradas, do ponto de vista de projeto e verificação? por exemplo, plano da qualidade, especificação do produto?
123. o projeto é avaliado ou verificado em relação aos requisitos do mercado? por exemplo, testes de verificação, modelos de computação, outros?
124. existem controles que se aplicam durante as fases de projeto e verificação?
125. são feitas revisões de projeto?
126. existem testes e monitoração de projeto, produção, instalação e processos de serviço, ou revisões de projeto do produto e auditoria do sistema da qualidade, dos processos ou do produto?
127. os planos de projeto demonstram aderência aos objetivos do plano de desenvolvimento e aos requisitos dos procedimentos?
128. os *inputs* de projeto são identificados e documentados e sua seleção é revisada para comprovar a adequação?
129. os resultados do projeto são detalhados para a especificação precisa de cada função?
130. existe definição precisa do cabeçalho de cada função, classe e tipo de dados?

- 131.o projeto da estrutura de cada pacote está definindo precisamente os algoritmos?
- 132.o registrado do histórico do projeto foi integrado à base dos históricos de outros projetos?
- 133.existe uma análise dos históricos que permite atuar sobre o processo eliminando a deficiência observadas?
- 134.os objetivos do projeto preliminar encontra-se claramente definidos?
- 135.no controle de limites todas as responsabilidades estão definidas?
- 136.todas as pessoas estão de acordo com o grupo?
- 137.cada *input*, função ou output está claramente definido por uma parte específica e identificável do sistema?
- 138.quanto à interface homem-máquina esta tudo conforme por parte do homem ou da máquina (não existe nenhuma divergência)?
- 139.a uma adaptação por parte do projeto onde todas as faces do projeto receberam a ênfase do que parecia merecer?
- 140.tudo o que foi considerado defeito a alteração foi feita?
- 141.o projeto reflete o equipamento no qual será executado?
- 142.seu projeto é independente da linguagem de programação que será usada? você dispõe de recursos para depuração?
- 143.o projeto todo foi implementado?
- 144.o código executa o que o projeto requer (o projeto foi traduzido corretamente)?
- 145.o projeto está correto e completo?
- 146.as atividades típica dessa fase foram identificadas conforme a necessidade ou oportunidade?
- 147.foi feito algum equacionamento e definição do problema?
- 148.todos sabiam as determinações dos objetivos e metas a serem alcançados?
- 149.foi feito uma análise do ambiente do problema?
- 150.foi feito uma análise das potencialidades ou recursos disponíveis?
- 151.existe uma avaliação da viabilidade de atingimento dos objetivos?
- 152.existe uma estimativa dos recursos necessários?
- 153.existe uma elaboração da proposta do projeto?
- 154.existe uma apresentação da proposta e venda da idéia?
- 155.existe uma avaliação e seleção com base na proposta submetida?

156. existe uma decisão quanto à execução do projeto?
157. é ativada a comunicação entre os membros da equipe do projeto?
158. é executada as etapas previstas e programadas?
159. é utilizado os recursos humanos e materiais, sempre que possível, dentro do que foi programado (quantidade e períodos de utilização)?
160. é efetuado as reprogramações no projeto segundo seu *status* e adotando os planos e programas iniciais com diretrizes?
161. é marcada pela dificuldade na manutenção das atividades dentro do que foi planejado e pelo desligamento gradual de empresas e de técnicos do projeto?
162. há uma aceleração das atividades que, eventualmente, não tenham sido concluídas?
163. realocação dos recursos humanos do projeto para outras atividades ou projetos?
164. elaboração de relatórios e transferência dos resultados finais do projeto?
165. emissão de avaliação globais sobre o desempenho da equipe do projeto e os resultados alcançados?

## VERIFICAÇÃO DO CÓDIGO

1. o código é rastreável para projeto e para os requisitos, testável, correto e aderente aos requisitos e padrões de codificação?
2. o código implementa a seqüência de eventos apropriada, interfaces consistentes, dados e fluxo de controle corretos, completeza, alocação de tempo e de orçamentos apropriada, e definição, isolamento e recuperação de erros?
3. o código selecionado pode ser originado a partir do projeto ou dos requisitos?
4. o código implementa proteção, segurança e outros requisitos críticos corretamente, conforme demonstrado por métodos adequadamente rigorosos?
5. o projeto foi adequadamente traduzido em código? (os resultados do projeto procedimental devem estar disponíveis durante esta revisão).
6. há erros de ortografia ou tipográficos?
7. as convenções de linguagem foram usadas adequadamente?
8. existe concordância em relação aos padrões de codificação quanto ao estilo de linguagem, comentários e cabeçalho do modular?
9. há comentários incorretos ou ambíguos?
10. os tipos de dados e a declaração de dados são apropriados?
11. as constantes físicas estão corretas?
12. todos os itens da lista de conferência do *walthrough* de projeto foram reaplicados (conforme necessário)?

13. são aplicadas os padrões de exigência, projetos e revisões de códigos?
14. todas as interfaces de usuário que utilizem seleções por menu têm uma região definida no vídeo (janela) onde fica o menu?
15. têm títulos identificadores do menu?
16. oferecem uma ou mais listas de opções?
17. têm um processo de seleção da opção (letra, código, posição do cursor) bem definida?
18. existirá um e somente um *character* de fim de arquivo?
19. alguns *caracteres* serão designados como *caractere* de fim de arquivo?
20. haverá pelo menos um *character* de fim de arquivo?
21. o valor imediatamente após a quantidade é o número de controle, que é definido por sua posição?
22. se existir um número de controle, ele virá após a quantidade?
23. um FCB (*File Control Block*) controla todo o conjunto de arquivos?
24. cada arquivo tem sua FCB?
25. cada arquivo é controlado por uma FCB, mas uma FCB pode controlar mais de um arquivo?
26. o arquivo será '99'?
27. o campo será '9999999'?
28. independente do tamanho do campo, cada *caractere* será um '9'?
29. o *status* de interrupção é setado em "opcional" e nunca mais é resetado?
30. o *status* de interrupção é setado em "opcional" cada vez que uma rotina que poderia tê-lo setado for terminada?
31. o *status* de interrupção é "setado" em "opcional" todo o tempo?
32. um outro lugar onde a informação de exceção aparece é no arquivo xyz?
33. um outro tipo de informação que aparece no arquivo xyz é a informação de exceção?
34. a informação de exceção estará no arquivo xyz também?
35. a seqüência terminal num *flag* e também termina fim de arquivo – isto é, a seqüência termina ou *flag* ou num fim de arquivo?
36. a seqüência termina com a condição dupla de *flag* mais fim de arquivo?
37. uma FCB controla dois arquivos – isto é, ambos os arquivos?
38. cada arquivo é controlado por um FCB – isto é, existem duas FCBs para os dois arquivos, uma para cada um?
39. o total de controle é extraído do registro que é o atual no sentido contábil?
40. o total de controle é extraído do registro que está sendo atualmente considerado pelo programa?
41. uma tabela é atualizada por cada lista?
42. cada lista tem sua única tabela para atualizar?
43. cada lista tem sua tabela para atualizar, que pode não ser única?
44. a soma de verificação está no próximo registro, que é o de resumos?
45. a soma de verificação está no primeiro registro que se segue, ou podem ser vários registros, de qualquer forma?
46. o campo será setado em '2000' (que é o maior valor esperado nesta aplicação)?

47. o total de controle é extraído a partir do registro ao final do arquivo?
48. o total de controle é extraído a partir do registro mais recente?
49. o total de controle é extraído a partir do registro anterior?
50. o total de controle é extraído a partir do registro que está sendo processado?
51. o total de controle é extraído a partir do registro com data mais recente?
52. o código de retorno pode conter um inteiro ou brancos?
53. o código de retorno pode conter inteiro ou brancos, mas também pode conter qualquer outra coisa não definida aqui?
54. somente dígitos podem estar na lista de número de peças?
55. a única lista na qual os dígitos aparecem é a lista de números das peças?
56. a seqüência termina num *flag*, num fim de arquivos ou em ambos?
57. quando a seqüência termina, uma e somente uma destas condições sra válida – *flag* ou fim de arquivo?
58. todos os clientes têm o mesmo valor em seu campo de controle?
59. todos os campos de controle de cliente têm o mesmo formato?
60. um campo é usado para todos os clientes?
61. o operador tem de preparar a unidade de disco designada, senão o sistema simplesmente repetirá a solicitação?
62. o operador deve preparar a unidade de disco designada, mas, se não o fizer, o sistema tomará atitudes alternativas?
63. uma unidade familiar consiste em uma mãe e seus filhos e em um pai e seus filhos?
64. uma unidade familiar consiste em uma mãe e um pai e todos os filhos cujos pais sejam aquela mãe e aquele pai?
65. a inclusão dos elementos sempre cessará com a inclusão do elemento com a data atual?
66. a inclusão dos elementos cessará como o elemento com a data atual, se não for interrompida de outra forma, como o término da lista, por exemplo?
67. o usuário (que escreveu esta especificação) criará um guia de operador?
68. o usuário e o analista (que são os leitores desta especificação) criarão um guia de operador?
69. o único modo de terminar uma sessão do terminal é através do comando “desliga”?
70. se o comando “desligar” for emitido, a sessão do terminal termina? se não, deverá haver outras formas de terminar?
71. a sessão do terminal finaliza quando o comando de desliga for exibido?
72. o contador é setado em zero quando a sub-rotina foi invocada?
73. o contador é setado em zero assim que a sub-rotina for invocada – e somente neste tempo?
74. o contador é setado em zero sempre (toda vez) que a sub-rotina for invocada?
75. o “pointer” será setado antes que o valor dos dados seja?
76. alguém terá setado o valor do “pointer” antes que o valor dos dados seja setado?
77. o valor do “pointer” deve estar setado antes que alguém tente setar o valor dos dados?
78. na função existe algum conceito ou uma idéia fundamental que possa ser facilmente expresso numa linguagem comum no código implementado?



79. a função dessa parte do código está bem-definida em relação à função geral do todo, e tal função encontra-se claramente expressa?
80. a rotina encontra-se protegida adequadamente, de forma a poder desempenhar sua função confiavelmente, apesar de um possível mau uso?
81. forma: independente do estilo adotado, o código encontra-se claro e definido quanto visto como um todo?
82. ele é significativo pra todas as classes de leitores que o usarão?
83. existem segmentos de código repetido, dentro ou entre as rotinas?
84. os comentários são úteis ou simplesmente desculpas para uma codificação pobre?
85. o nível de detalhe é consistente?
86. os padrões estão sendo obedecidos?
87. a inicialização é feita adequadamente? a rotina tem um procedimento claro de desvio ou saída?
88. existem operações redundantes para as quais não há benefícios compensadores?
89. o uso do armazenamento é consistente, tanto internamente quanto em relação às especificações externas?
90. quanto custará modificar o código?
91. o código é simples?
92. os nomes dos itens de dados estão em conformidade com seu uso?
93. os itens de dados (que precisam estar numa ordem particular) estão seqüenciados corretamente?
94. os blocos de comentários estão incluídos nas áreas funcionais importantes?
95. os comentários são precisos e significativos?
96. num programa segmentado, está claro o porquê da segmentação?
97. cada segmento é independente ou existe uma forte dependência de outros segmentos?
98. os grupos *then / else* estão alinhados?
99. os ninhos de *ifs* estão “indentados” adequadamente?
100. os blocos de comentários e observações são posicionados eficientemente?
101. o projeto todo foi implementado?
102. o código executa o que o projeto requer (o projeto foi traduzido corretamente)?
103. cada campo a ser inicializado está definido corretamente?
104. antes do primeiro uso de qualquer campo, ele foi inicializado adequadamente?
105. foi especificado o campo correto?
106. cada desvio-alvo de uma simples instrução *goto* está correto e praticado ao menos uma vez?
107. o código está codificado idealmente (isto é, com instruções em menor número e mais eficientes)?
108. escreva de forma clara – não seja prolixo?
109. diga o que você quer dizer – simples e diretamente?
110. é utilize funções de biblioteca?
111. é evitado variáveis temporárias?
112. escreva de forma clara – não sacrifique a clareza pela “eficiência”?
113. existe expressões repetitivas por chamadas a uma função comum?

114. utilize parênteses pra evitar ambigüidades?
115. escolha nomes para as variáveis que não sejam confusos?
116. evite desvios desnecessários?
117. não use desvios condicionais como substitutos par uma expressão lógica?
118. se uma expressão lógica for difícil de ser compreendida, tente modifica-la?
119. utilize *arrays* de dados pra evitar seqüências de controle repetitivas?
120. escolha uma representação de dados que torne o programa simples?
121. escolha uma representação de dados que torne o programa simples?
122. use *if...else* para implementar desvios de múltiplos caminhos?
123. é usado sub-rotinas?
124. use *gotos* somente para implementar uma estrutura funcional
125. evite completamente o uso de *gotos* se você puder manter o programa legível?
126. não conserte código ruim – reescreva-o?
127. escreva e teste um programa grande em pequenas partes?
128. use procedimentos de recursividade para estruturas de dados definidas recursivamente?
129. teste inputs visando que sejam plausíveis e válidos?
130. certifique-se que de as entradas não infringem os limites do programa?
131. termine a entrada através de um “*end-of-file*” ou um marcador, e não por um contador?
132. identifique os inputs inconsistentes? recupere-os se possível?
133. torne as entradas fáceis de serem preparadas e as saídas auto-explicativas?
134. utilize formatos uniformes de entradas?
135. torne os *inputs* à prova de erros de leitura?
136. use *inputs* de formato livre sempre que puder?
137. use *inputs* auto-identificáveis? considere o uso de *defaultes*? Faça ambos repercutirem no *output*?
138. certifique-se de que todas as variações sejam inicializadas antes de serem utilizadas?
139. não perca tempo com um *bug*?
140. use compiladores com facilidades de deputação?
141. cuidado com erros que provoquem o término do programa na primeira execução?
142. atente para os desvios certos em função da igualdade?
143. teste os programas em seus valores-limites?
144. certifique-se de que código de comentário – apenas faça cada comentário valer?
145. formate um programa para ajudar o leitor a compreende-lo?
146. documente seus *layouts* de dados?
147. existe um objetivo de diretrizes para facilitar a manutenção do código, promovendo sua legibilidade e desestimulando as construções mais propensas a erros?
148. a codificação de software o padrão descreve um conjunto de diretrizes que devem ser seguidas na implementação de software?
149. algumas diretrizes para o uso de tipos numéricos em geral: usar constantes no lugar de literais (exceto no caso de 0 e 1)?

150. verificar se o valor dos divisores não pode ser 0?
151. explicar todas as conversões de tipo?
152. evitar comparação de tipos mistos?
153. no caso de inteiros, considerar: tratar possibilidades de truncamento?
154. tratar possibilidade de estouro (*overflow*), inclusive em resultados intermediários?
155. no caso de números de ponto flutuante, considerar: evitar somas e subtrações de números de magnitudes muito diferente?
156. evitar comparação de igualdade?
157. tratar possibilidades de *overflow* e *underflow*?
158. tratar possibilidades de erro de arredondamento?
159. no caso de *characters* e texto, considerar: usar constantes em lugar de literais?
160. agrupar as constantes de textos em um arquivo de recursos, para facilitar mudança e tradução?
161. no caso de variáveis *booleanas*, observar: fazer com que as variáveis *booleanas* ajudem a documentar o programa?
162. substituir testes complicados por variáveis *booleanas* intermediárias?
163. criar seu próprio tipo *booleano*, se não existir na linguagem?
164. no caso de variáveis enumeradas, observar: usar enumerações para facilitar a leitura?
165. usar enumerações para aumentar a confiabilidade, promovendo checagens em tempo de compilação?
166. usar enumerações para facilitar modificações?
167. usar enumeração para generalizar variáveis *booleanas* para mais de dois resultados possíveis?
168. testar valores inválidos?
169. reservar uma entrada para representar valores inválidos?
170. se a linguagem não tiver enumerações, simula-la por meio de uma convenção de denominação?
171. no caso de arranjos, observar: usar arranjos apenas para acesso seqüencial, e não randômico, de preferência?
172. checar os pontos extremos dos arranjos?
173. em arranjos multidimensionais, checar a ordem dos subscritos?
174. em malhas aninhadas, cuidado com as confusões entre índices?
175. por via das dúvidas, colocar um elemento extra no final do arranjo?
176. no caso de apontadores, existe um isolamento aos apontadores em rotinas especializadas?
177. checar os apontadores antes de usa-los?
178. checar as variáveis apontadas antes de usa-las?
179. usar campos de verificação de erros, em variáveis apontadas?
180. anular o valor apontadores, depois de libera-los?
181. usar apontadores extra, quando isso contribuir para tornar o código mais legível?
182. simplificar expressões com apontadores?
183. procurar formas seguras de controlar a alocação e liberação de apontadores?
184. documentar operações sobre apontadores, de preferência em formato gráfico?
185. checar, na liberação de listas, se a ordem de liberação dos apontadores é correta?

186. tipos estruturados de dados (registros e estruturas) devem ser usados para: esclarecer relacionamento de dados?
187. simplificar operações sobre blocos de dados?
188. simplificar listas de parâmetros, desde que o agrupamento seja lógico?
189. reduzir encargos de manutenção?
190. são declaradas todas as variáveis?
191. as declarações são acompanhadas de comentários descritivos?
192. é iniciada todas as variáveis em sua declaração?
193. minimizar o escopo das variáveis?
194. usar cada variável com uma única finalidade?
195. minimizar o uso de variáveis globais?
196. usar variáveis globais só em último caso?
197. todos os nomes descrevem de forma completa e acurada a entidade que representam?
198. todos os identificadores representam palavras ou frases do idioma utilizado?
199. é separado as palavras constituintes dos nomes longos?
200. abreviações são evitadas, se a linguagem o permitir?
201. os nomes são pronunciáveis?
202. os nomes são diferenciáveis pelos seus primeiros *caracteres*?
203. é evitado o uso da notação húngara e de outras notações que denotem informações de baixo nível?
204. é evitado a ocorrência de identificadores com nomes que possam ser confundidos?
205. é evitado o uso de nomes predefinidos no ambiente de desenvolvimento, a não ser que estejam sendo redefinidos?
206. quando existe uma seqüência correta de execução das instruções, ela é facilmente reconhecível pela leitura do código?
207. o código é organizado de forma que possa ser lido de cima para baixo?
208. instruções correlatas estão agrupadas, e os grupos são separados por linha em branco e linhas de comentário?
209. não utilizar o *else* final de uma cadeia de *ifs* para tratar um caso restante? reserva-lo para tratar condições de exceção?
210. ao utilizar seleções múltiplas (*case/switch*), é classificado os casos por ordem alfabética/numérica ou por freqüência de ocorrência, mantendo entretanto juntos os casos cujo tratamento usa o mesmo código?
211. é usado sempre o tipo de malha mais adequado à lógica da iteração?
212. malhas tipo *for* são usadas quando se sabe que a malha deverá ser executada um número específico de vezes?
213. é usado um formato padronizado e facilmente reconhecível para malhas eternas?
214. a condição de saída da malha é facilmente identificável?
215. é evitado malhas vazias?

- 216.o uso de efeitos colaterais dentro do teste de terminação das malhas é evitado, ou pelo menos documentado com destaque?
- 217.cada malha executa apenas uma função?
- 218.*gotos* só são usados em casos extremos, em que o código ficara mais ilegível sem o *goto* do que com ele?
- 219.só é usado o *return* para melhorar a legibilidade?
- 220.só é usado a recursão quando puder provar que ela tem limite?
- 221.é simplificada as expressões de teste usadas para controle?
- 222.são organizados os testes numéricos na direção dos pontos de uma reta?
- 223.é minimizado os níveis de aninhamento?
- 224.há pelo menos um espaço de cada lado de um operador binário?
- 225.símbolos parentéticos tem um espaço depois do símbolo de abertura e antes do símbolo de fechamento?
- 226.há pelo menos uma linha em branco deve separar um comentário do código precedente?
- 227.linhas de código excedem 80 *caracteres*?
- 228.seções mais importantes de programa são precedidas por um comentário de bloco?
- 229.todos os comentários estão sempre atualizados?
- 230.todo código não-definido são marcado através de um tipo de comentário padronizado?
- 231.é produzir o código de cada programa?
- 232.é gerado as tabelas de dados utilizados pelos programas (por exemplo, tabelas de mensagens)
- 233.é produzido as definições dos recursos utilizados pelo programa (por exemplo, diálogos, menus, ícones)?
- 234.é produzido os textos de auxílio (*help*)?
- 235.é produzido os tutoriais?
- 236.as regras e recomendações de escolha de nomes têm por objetivo tornar fácil?
- 237.é escolhido bons nomes novos ao criar ou manter programas?
- 238.é fácil de localizar a especificação e a declaração completa do nome?
- 239.lembrar-se corretamente dos nomes dos elementos já declarados ao redigir ou alterar código?
- 240.entender corretamente um trecho de código sendo lido, sem precisar recorrer à documentação na grande maioria das vezes?
- 241.assegurar que os elementos do programa possuam tipos consistentes como contexto em que estão sendo utilizados?
- 242.argumentar a corretude de trechos de programas?
- 243.acompanhar corretamente o comportamento do programa ao utilizar ferramentas de apoio à depuração?
- 244.existe uso de variáveis não definidas ou não inicializadas?
- 245.todas as variáveis foram declaradas explicitamente?
- 246.existem variáveis de tipos inconsistentes de dados e equações programadas incorretamente?
- 247.existe uso incorreto de operadores semelhantes?
- 248.há número incorreto e ordem de parâmetros em uma chamada de sub-rotina?
- 249.o programa executa testes de consistência devidos na entrada?

250. nomes de dados e rótulos significativos são significativos para o leitor?
251. para o entendimento torna-se mais claro pela especificação de nomes, é usado uma forma constituída de múltiplas partes?
252. os nomes padronizados pela instalação são usados para campos com utilizações padrões?
253. existe elemento no nome que (por exemplo, um prefixo) identifica meio físico ou o tipo de dado?
254. os rótulos referem-se ao esquema do programa?
255. eles identificam sua posição seqüencial no programa?
256. a estrutura da lógica é auto-evidente?
257. a lógica é adequada ao problema?
258. a estrutura é facilmente corrigível?
259. a lógica tem um fluxo de cima para baixo?
260. uma técnica-padrão de projeto de programa está sendo utilizada?
261. se componentes são usados, estes são sensíveis quanto à função?
262. se componentes são usados, estes têm entrada e saída únicas?
263. se componentes são usados, estes utilizam apenas desvios internos?
264. todas as sub-rotinas usadas estão agrupadas em conjunto?
265. na prática de codificação a lógica do programa é auto-evidente, pela codificação?
266. as ações realizadas pela codificação são compreendidas?
267. as finalidades de se usarem contadores e modificadores são auto-evidentes pela codificação?
268. os periféricos são abertos apenas quando necessários?
269. os periféricos são fechados quando não são mais necessários?
270. os campos de dados são formatados, possibilitando uma utilização perfeita das capacidade da máquina?
271. as sub-rotinas-padrão são utilizadas?
272. os nomes padrões para áreas são usadas?
273. apresentação de formato: a facilidade de pular para uma nova página é utilizada para destacar componentes etc.
274. linhas em branco são usada para dividir parágrafos etc?
275. os componentes são de tamanho aceitável?
276. a paragrafação é usada para possibilitar uma maior clareza?
277. cada comando com determinada função começa em uma nova linha?
278. os comentários são usados para possibilitar o entendimento?
279. os itens de tipos similares são alinhados à mesma coluna?
280. notas e comentários: há o suficiente para facilitar o entendimento?
281. existe uma descrição da função do programa no seu início?
282. existe uma descrição inicial no começo de cada componente?
283. existe uma descrição para uma codificação específica?
284. são úteis, claros e concisos?
285. são escritos ao mesmo tempo que o código?
286. são imediatamente identificados como notas, circundados por asteriscos?

287. estão em lugares-padrões no programa?
288. são utilizadas para fornecer detalhes de módulos/sub-rotinas chamados?
289. são usados para fornecer detalhe dos parâmetros de sub-rotina?
290. documentação de programa: a listagem de programa realmente existe?
291. ela registra sua geração e data?
292. está armazenada no lugar certo?
293. possui uma lista de referência cruzada?
294. possui um mapa das áreas estáticas?
295. tendo-se mais de uma listagem do programa, existe uma nota da finalidade específica cada listagem?
296. as correções são datadas?
297. as correções são claramente indicadas?
298. as correções referem-se à especificação?
299. a especificação de programa possui formato-padrão?
300. contém documentos-padrões?
301. existe um esquema de alto nível?
302. descreve as interfaces do sistema?
303. contém explicação para trabalho específico?
304. *layouts* de arquivos e descrição de registro estão em documentos-padrões?
305. os *layouts* estão incluídos ou referenciados?
306. usa-se de modo perfeito as áreas de armazenamento?
307. os nomes-padrões são usados para determinar áreas?
308. estão atualizados?
309. esquema de programa: existe um esquema global?
310. os símbolos-padrões são usados?
311. está dentro de um formato-padrão?
312. está atualizada e é uma representação real do programa?
313. os identificadores de componentes são utilizados no programa?
314. atinge um nível de detalhes satisfatório?
315. existe um esquema para componente?
316. existe uma lógica esquemática?
317. instruções operacionais estão em documentos-padrões?
318. existe um esquema operacional?
319. existem descrições de mensagem e ações para códigos de erro?
320. as instruções operacionais estão presentes?
321. estão atualizadas?
322. é de fácil entendimento?
323. existem definições de todas as macros usadas?
324. existe manual de instruções para execução?
325. pelo ponto de correção é identificada a partir da esquemática do programa?

326. é fácil descobrir a razão do uso de técnicas modulares?

327. é fácil descobrir o porquê do uso de comentários?

328. é fácil descobrir o porquê do uso de nomes significativos para rótulos ou parágrafos?

## VERIFICAÇÃO DA INTEGRAÇÃO

1. os componentes de software e unidade de cada item de software foram completa e corretamente integrados dentro do item do software?
2. os itens de hardware, de software e operações manuais do sistema foram completa e corretamente integrados ao sistema?
3. as tarefas de integração foram executadas de acordo com um plano de integração?
4. as funções principais são definidas de uma forma delimitadas sem ambigüidades?
5. as interfaces entre os elementos do sistema estão definidas?
6. foram definidos limites de desempenho para o sistema como todo e para cada elemento?
7. a organização utiliza-se de uma estratégia definida para conduzir a integração do produto através da composição de seus componentes?
8. existe uma política organizacional documentada para assegurar a compatibilidade das interfaces internas entre os componentes de software, e as interfaces de software com outros sistemas de software, com o hardware, com sistemas de suporte e com outros componentes de sistema?
9. são planejados e executados testes de integração do software e respectivas documentação de acordo com o processo de software definido do projeto?
10. as restrições de projeto foram estabelecidas para cada elemento?
11. as melhores alternativas foram selecionadas?
12. a solução é tecnologicamente viável?
13. foi estabelecido um mecanismo de validação e verificação para o sistema?
14. há consistência entre todos os elementos do sistema?
15. uma política é emitida, requerendo que cada grupo de desenvolvimento demonstre um padrão crescente de produtividade?
16. uma política da empresa requer que cada grupo de desenvolvimento utilize os métodos e ferramentas oficiais estabelecidos?
17. o ponto focal da organização é o desenvolvimento, a introdução e o apoio à reutilização de componentes?
18. revisões gerenciais periódicas são realizadas para avaliar o desempenho da produtividade?
19. meios são estabelecidos para assegurar que o processo de software e as ações de melhorias sejam objetivamente avaliadas, reconhecidas e recompensadas?



20. cursos padrões são oferecidos para justificação econômica, projetos avançados, métodos de reutilização e métodos de prevenção de erros?
21. planos de melhoria do processo são produzidos para cada projeto?
22. planos de melhoria de produtividade são produzidos para cada projeto?
23. o desempenho é avaliado em função das melhorias no processo e produtividade?
24. a reutilização de componentes e suas definições são mantidas sob o controle da gerência da configuração de software?
25. melhorias de processo e produtividade dos subcontratados são revistas, aprovadas e monitoradas?
26. a área de garantia da qualidade da empresa monitora as ações de melhoria de processo e de produtividade dos subcontratados?
27. padrões documentados são produzidos para a medição da produtividade de cada tarefa em termos de limites estatísticos?
28. guias documentados e padrões são produzidos para a reutilização de componentes, incluindo padrões de interfaces?
29. métodos documentados são desenvolvidos para o projeto de componentes reutilizáveis, análise de prevenção de defeitos e extinção de defeitos?
30. estudos econômicos e técnicos são conduzidos durante o projeto para decidir quando componentes reutilizáveis devem ser empregados?
31. medições e análises são realizadas sobre eficácia de ferramentas, métodos de prototipação, da reutilização?
32. análise de causas de defeitos são conduzidos?
33. existe um mecanismo formal para gerenciar a introdução de novas tecnologias?
34. é utilizado um mecanismo para identificar e substituir as tecnologias obsoletas?
35. mecanismos e responsabilidades são definidas para acompanhamento de produtividade, integração de tecnologia e acompanhamento da eficácia da reutilização?
36. mecanismos e responsabilidades são estabelecidas para assegurar que os profissionais sejam treinados e capazes de desempenhar prevenção de defeitos e que sejam pessoalmente comprometidos com as melhorias do processo?
37. o apoio da tecnologia inclui a análise de cada tarefa para determinar o apoio requerido, os aspectos econômicos deste apoio e a prototipação de ferramentas e métodos potenciais?
38. o conjunto de ferramentas é progressivamente aperfeiçoado para incluir o uso de novas tecnologias e métodos?

39. há plano de teste para software?
40. o plano de teste contempla: teste de item de software, teste de integração, teste de sistema e teste de aceitação?
41. o plano de teste contempla casos de testes, dados de testes e resultados esperados?
42. o plano de teste contempla tipos de testes a serem desempenhadas, ou seja, funcional, testes de interfaces, testes de desempenho, teste de usabilidade?
43. o plano de teste contempla ambiente de teste e ferramentas de testes?
44. o plano de teste contempla os critérios pelos quais o término do teste será julgado?
45. o plano de teste contempla documentação do usuário?
46. durante o teste é levado em consideração o registro dos resultados dos testes?
47. os responsáveis são notificados para a resolução de problema?
48. os responsáveis são notificados para a resolução de problema?
49. é feita uma validação do software antes da entrega para o cliente em termos de sua operação e, quando possível, em condições reais?
50. são especificadas as funções a serem testadas?
51. são definidas as responsabilidades de ambas as partes na avaliação do teste?
52. há a restauração do ambiente do usuário após o teste?
53. quando o teste sob condições de campo é exigido replicação, entrega e instalação
54. no caso de re replicação do software é especificado o número de cópias do software?
55. é especificado o tipo de meio para cada item de software, incluindo formato e versão?
56. é especificado a documentação requerida, como manuais e guias do usuário?
57. é especificado termos de licenciamento?
58. são especificados os aspectos relativos a custódia e *back-up* de cópias?
59. é especificado a obrigação de reposição de cópias ao cliente?
60. há procedimentos para verificar a exatidão das cópias após sua replicação?
61. durante a instalação do software, as responsabilidades de ambas as partes são definidas em termos de cronograma, incluindo fins de semana?
62. durante a instalação do software, as responsabilidades de ambas as partes são definidas em termos de acesso às instalações do cliente (físicas, *password* etc)?

63. durante a instalação do software, as responsabilidades de ambas as partes são definidas em termos de disponibilidade de pessoal especializado?
64. durante a instalação do software, as responsabilidades de ambas as partes são definidas em termos de acesso aos equipamentos e sistemas do cliente?
65. durante a instalação do software, as responsabilidades de ambas as partes são definidas em termos de necessidade de validação de cada instalação do produto?
66. há procedimentos para as atividades de manutenção requeridas?
67. há um plano de manutenção do software?
68. o plano de manutenção contempla o escopo da manutenção?
69. o plano de manutenção contempla a identificação do status inicial do produto?
70. o plano de manutenção contempla a organização de suporte?
71. o plano de manutenção contempla os registros de manutenção e relatórios?
72. o plano de manutenção contempla as atividades de manutenção?
73. a identificação efetuadas no software em função da manutenção são documentadas e controladas?
74. para cada item de software que sofre manutenção há uma lista de solicitação ou relato de problemas recebido do cliente e o *status* atual de cada solicitação?
75. para cada item de software que sofre manutenção há responsáveis para responder às solicitações ou para implementar a ação corretiva apropriada?
76. para cada item de software que sofre manutenção há prioridade atribuídas às ações corretivas?
77. para cada item de software que sofre manutenção há o registro das ações corretivas?
78. para cada item de software que sofre manutenção são mantidos dados estatísticos sobre a ocorrência de falhas e atividades de manutenção?
79. os registros das atividades de manutenção são utilizados para a melhoria do produto de software e do próprio sistema da qualidade?
80. há procedimentos documentados para incorporar mudanças no produto de software numa abordagem de release?
81. estes procedimentos incluem regras para determinar onde novas implementações serão realizadas ou incorporadas no produto?
82. estes procedimentos incluem regras para determinar métodos pelos quais o cliente será comunicado sobre mudança planejadas no produto?
83. estes procedimentos incluem regras para determinar métodos para confirmar que as mudanças a serem implementadas não introduzirão outros problemas?

84. existem registros indicando quais mudanças foram implementadas por instalação, para múltiplos produtos?
85. há um sistema de gerência de configuração?
86. este sistema identifica unicamente as versões de cada item de software?
87. o sistema identifica as versões de cada item de software, que juntos, constituem uma versão específica de um produto completo?
88. o sistema identifica o *status* de construção do produto de software em desenvolvimento ou entregue e instalado?
89. o sistema controla a atualização simultânea de um item de software específico por mais de uma pessoa?
90. o sistema provê a coordenação para a atualização de múltiplos produtos em uma ou mais instalações quanto requerido?
91. este plano contempla a organização ou áreas envolvidas na gerência da configuração e responsabilidades atribuídas para cada um?
92. há atividades de gerência da configuração a serem desempenhadas?
93. há ferramentas de gerência da configuração, técnicas e metodologias a serem utilizadas?
94. existe o estágio que os itens devem ser colocados sob o controle da gerência da configuração?
95. há procedimentos par identificar os itens de software durante todas as fases de desenvolvimento, desde a especificação até a replica e entrega?
96. cada item de software tem uma identificação única?
97. a identificação de cada item de software para cada versão contempla as especificações funcionais e técnicas?
98. a identificação de cada item de software para cada versão contempla todas as ferramentas de desenvolvimento que afetam as especificações funcionais e técnicas?
99. a identificação de cada item de software para cada versão contempla todas as interfaces com outros itens de software e hardware?
100. a identificação de cada item de software para cada versão contempla todos os documentos e arquivos de computador relacionados ao item de software?
101. há procedimentos para identificar, documentar, revisar e autorizar qualquer mudança aos itens de software sob a gerência da configuração?
102. todas as mudanças nos itens de software seguem estes procedimentos?
103. há métodos para notificar as mudanças às partes interessadas e para mostrar a “rastreadibilidade” entre mudanças e partes modificadas dos itens de software?

- 104.há procedimentos para registrar, gerenciar e relatar sobre o status dos itens de software, sobre solicitação de mudanças?
- 105.para registros das qualidades há procedimentos para identificação, coleta, indexação, arquivamento, armazenamento, manutenção e disposição de registro da qualidade?
- 106.há procedimentos para determinar período de tempo em que os registros devem ser retidos?
- 107.existe uma relação das funções que deverão ser desempenhadas pelo sistema de computador?
- 108.existe algum modelo da interface entre usuário e o sistema de computador?
- 109.tal modelo prevê uma descrição das linguagens disponíveis ao usuário?
- 110.há quantidades de detalhes suficientes para que você consiga assimilar o uso do sistema em sua própria mesa de trabalho?
- 111.existe informações sobre flexibilidade / adaptabilidade / extensibilidade da interface de usuário?
- 112.existe informações pra o usuário sobre tutoriais, assistência etc?
- 113.há uma descrição das funções disponíveis ao usuário e do acesso a essas funções?
- 114.há uma avaliação da facilidade de uso do sistema de computador?
- 115.o nível de detalhe necessário para a prática dos procedimentos do usuário são empregados para uso do sistema de computador?
- 116.existem modelos ou descrições de todas as outras interfaces do sistema de computador?
- 117.existe um modelo funcional de alto nível do sistema proposto?
- 118.tal modelo é acompanhado de uma descrição operacional?
- 119.o modelo é acompanhado de uma explicação dos procedimentos de testes, casos de testes, resultados de testes e análise de testes utilizados para garantir precisão do modelo?
- 120.tal modelo é acompanhado de uma avaliação do modelo quanto aos requisitos, para garantir que estes sejam satisfeitos? (o projeto preliminar não fornece resultados detalhados que permitam uma análise qualitativa e quantitativa detalhada?)
- 121.as principais alternativas de implementação e suas avaliações encontram-se presentes no documento?
- 122.o modelo é preciso e não-ambíguo que identifique módulos, conjuntos de *inputs* e *outputs* módulos e critérios para execução de cada seqüência operacional no modelo?
- 123.existe uma avaliação do modelo que garanta que os requisitos serão satisfeitos? algumas das qualidade que tal avaliação deve conter são: desempenho, requisitos de armazenamento, qualidade dos resultados, facilidade de uso, manutenibilidade, adaptabilidade, generalidade, excelência técnica, simplicidade, flexibilidade, legibilidade, portabilidade, modularidade?

- 124.há um exame desses modelos deve incluir tanto o modelo funcional quanto os modelos de dados relacionais?
- 125.existem estimativas de custo, tempo e outros recursos que serão necessários para implementação da alternativa?
- 126.essas estimativas são acompanhadas de uma descrição da técnica utilizada para a estimativa?
- 127.há uma identificação daqueles módulos que serão implementados como hardware e aqueles que serão implementados como software? (alguns serão uma combinação de hardware e software).
- 128.a recomendação da equipe de projeto preliminar está adequadamente fundamentada?
- 129.as informações apresentadas no documento do projeto preliminar e durante a revisão do projeto preliminar lhe dão a segurança de que o sistema de computador poderá ser implementado para satisfazer os requisitos de forma a convence-lo a utilizar o sistema proposto?
- 130.o sistema permite saídas e devolução de rotinas fornecidas pelo usuário?
- 131.o sistema pode processar mais de um arquivo ao mesmo tempo?
- 132.os arquivos do usuário são reformatados antes do processamento?
- 133.no caso de reformatação, é gerado um novo arquivo?
- 134.há alguma restrição especial?
- 135.mais de um arquivo de saída ode ser gerado durante uma única rodada?
- 136.os formatos são restritos?
- 137.com relação às saídas impressas, são de formatos razoavelmente flexíveis?
- 138.o sistema permite o uso de funções de tabelas?
- 139.utilização perfeita dos recursos de hardware?
- 140.o tamanho do programa está relacionado com a máquina e o sistema operacional?
- 141.o programa mantém uma interface satisfatória com o sistema?
- 142.os dados são manipulados no formato mais adequado?
- 143.o uso de comandos, considerados lentos quanto à operação, é restrito?
- 144.em nível de ambiente operacional a comunicação com os operadores é mantida em um mínimo?
- 145.a comunicação com os operadores é significativa e útil?
- 146.há um procedimento-padrão para reinício?
- 147.o programa é fácil de ser operado?
- 148.o programa é fácil de ser preparado?

149. todas as funções necessárias estão disponíveis?
150. as necessidades futuras do usuário podem ser implementadas pelo fornecedor ou pelo usuário?
151. vai rodar dentro das restrições impostas pelo hardware agora disponível?
152. se outro equipamento for necessário, qual será o custo e a disponibilidade?
153. pode ser convertido para rodar no hardware atual?
154. será fácil mudar o sistema para outros periféricos no mesmo computador?
155. pode o sistema operacional ser mudado?
156. o sistema pode ser rodado em outro computador?
157. pode ser rodado em mais de um computador sem alterar o custo, no mesmo local ou em outro?
158. as despesas quanto à documentação são separadas?
159. existe despesa quanto o suporte ou às atualizações?
160. outros usuários encontram-se satisfeitos?
161. outros usuários possuem o mesmo hardware?
162. outros usuários podem se contatar?
163. pode ser dada uma demonstração?
164. pode ser dado um julgamento?
165. há uma documentação disponível?
166. se o fornecedor abandonar o ramo, o código-fonte se mantém disponível?
167. algum elemento do sistema entra em conflito com a prática da instalação?
168. terá interface com outro software?
169. antes de desempenhar as atividades de aceitação, o fornecedor deve assistir o comprador na identificação da programação da aceitação, procedimentos para avaliação, o ambiente de hardware/software e recursos necessários e os critérios de aceitação a serem utilizados?
170. a uma determinação do número de cópias do software a serem entregues?
171. a estipulação da documentação requerida?
172. há questões relativas a direitos autorais?
173. existe um período de obrigação do fornecedor suprir cópias?
174. as cópias a serem entregues são verificadas quanto a sua completeza?
175. os papéis do fornecedor e comprador estão plenamente estabelecidos quanto à instalação?

- 176.o fornecedor desenvolve um plano de gerência de configuração que compreende: organização e responsabilidades correspondentes relativas à gerência de configuração, atividade a serem desempenhadas, ferramentas, técnicas e metodologias a serem usadas e os estágio no qual os itens de software devem entrar sob controle da gerência de configuração?
- 177.procedimentos são aplicados a fim de assegurar para cada versão do software: as especificações funcionais e técnicas, todas as ferramentas de desenvolvimentos que afetam estas especificações, todas as interfaces com outros itens de software e hardware e todos os documentos e arquivos de computador relacionados ao item de software?
- 178.o fornecedor estabelece e mantém procedimentos para identificar, documentar, revisar e autorizar quaisquer mudança no software sob controle da gerência de configuração?
- 179.qualquer equipamento pode ser usado para processar o sistema?
- 180.dispositivos de entrada/saída é preciso serem requeridos?
- 181.qualquer tipo de meio e organização de arquivos podem ser processados?
- 182.há restrições específicas?
- 183.os procedimentos fornecem instalações de armazenagem que evitam danos ou degradação do produto? exemplo, instalação para armazenagem, prateleiras, caixas?
- 184.ao transferir os programas da plataforma de desenvolvimento para a plataforma de uso a base de dados de uso está completa e corretamente povoada?
- 185.os usuários saibam utilizar completa e corretamente o sistema?
- 186.o sistema de instalação está operando corretamente?
- 187.o sistema atende efetivamente às necessidades do usuário?
- 188.o sistema está em conformidade com as suas especificações?
- 189.o usuário consegue utilizar o sistema sem requerer assistência continuada?
- 190.o sistema opera em todas a plataformas e respectivas configurações a que se destina?
- 191.existe um equilíbrio correto entre a educação humana e a da máquina?
- 192.os recursos educacionais constituem um ambiente adequado para o método de apresentação?
- 193.o equipamento é consistentemente confiável e consistentemente disponível?
- 194.os equipamentos suplementares e os materiais encontram-se confiável e consistentemente disponíveis (por exemplo, terminais, impressoras, papel, manuais)?
- 195.foram alocados tempo suficiente e recursos necessários para exercícios e sessões de problemas?
196. foi alocado tempo suficiente para problemas e exercícios de programação?



## VERIFICAÇÃO DA DOCUMENTAÇÃO

1. a documentação está adequada, completa e consistente?
2. a preparação da documentação está oportuna?
3. a gerência de configuração dos documentos segue procedimentos especificados?
4. todos os projetos produzem e documentam seus planos, incluindo projeções do tamanho do produto, estimativa de recursos, pessoal, prazos e pontos de controle?
5. revisões gerenciais periódicas (mensais ou trimestrais) são conduzidas para acompanhar o desempenho do projeto em termos de tamanho, pessoal, cronogramas e pontos de controle?
6. mecanismos são estabelecidos para monitorar e controlar as mudanças no requerimentos?
7. existe um processo organizacional documentado para checar o software e seus componentes contra os seus requisitos de modo a assegurar que implementam exatamente todos os requisitos especificados?
8. sua organização segue um plano documentado de atividades para desenvolvimento e melhoria do processo de software?
9. o projeto segue uma política organizacional documentada para executar revisões?
10. você mantém um registro do tempo gasto e dos recursos utilizados durante o teste, servindo para calcular o custo do teste?
11. você arquiva e guarda casos de testes e massas de dados para uso futuro?
12. você mantém sempre um registro dos problemas operacionais, usando-o para avaliar e aumentar a eficácia dos procedimentos de teste?
13. a organização provê um procedimento documentado para executar um plano de testes direcionado à validação do software e seus componentes quanto a sua adequação para o uso em seu pretendido ambiente operacional?
14. um sistema de gerência de configuração e de controle de mudança é estabelecido para requerimentos, codificação e resultados de testes?
15. padrões documentados são produzidos para estimativas de tamanho de software, projeção de recursos, prazos e planos de produto?
16. guias de estilo de produto são desenvolvidos para orientar as atividades de projeto e codificação?
17. procedimentos documentados são produzidos para o desenvolvimento e aprovação dos planos do produto, aprovação para mudanças nos requerimentos, projetos ou código, condução de auditorias e revisões gerenciais?
18. os métodos básicos de projeto, codificação e teste de software estão definidos e documentados?
19. dados históricos são retirados sobre o código, erros de teste etc?

20. mecanismos e responsabilidades são estabelecidos para assegurar que os planos sejam produzidos e acompanhados para todos os projetos?
21. os planos são aprovados pelas equipes de implementação?
22. todos os planos são revistos antes de haver comprometimento efetivo?
23. auditorias são conduzidas quando especificadas, controle de mudança é rigorosamente implementado e a garantia da qualidade e gerência de configuração de software são efetivamente desempenhadas?
24. procedimentos, métodos e responsabilidades são estabelecidas par assegurar que os desenvolvedores entendam tanto os requerimentos como o ambiente dos usuários da aplicação?
25. meios são estabelecidos para prover o ambiente de desenvolvimento com o conhecimento sobre ferramentas e métodos disponíveis?
26. todos os itens são cobertos por algum tipo de documentação?
27. as atividades de produção e verificação são documentadas?
28. a documentação técnica e de uso (auxílios, manuais, tutoriais) existem, está completa e é consistente como o que se encontra implementado?
29. a documentação de uso está na versão definitiva e correspondente exatamente ao sistema implementado?
30. a documentação técnica corresponde exatamente ao que está implementado?
31. a documentação e todo o código está catalogado e acessível à equipe de manutenção?
32. foram considerados possíveis efeitos colaterais associados a mudança?
33. a solicitação de mudança foi documentada, avaliada e aprovada?
34. a mudança, uma vez feita, foi documentada e relatada a todas as partes interessadas?
35. uma revisão de aceitação final foi realizada para ter a garantia de que todo o software foi adequadamente atualizado, testado e substituído?
36. o documento de projeto preliminar contém uma descrição dos procedimentos utilizados para elaboração do projeto preliminar ou existe alguma referência a tais procedimentos?
37. este procedimento deve incluir uma descrição da técnica de projeto utilizada?
38. existe uma explicação da representação do projeto?
39. há uma descrição dos procedimentos de testes, casos de testes, resultados de testes e análise dos testes utilizados?
40. há uma descrição dos procedimentos avaliação e dos critérios utilizados?
41. todas as fases do ciclo de vida da documentação foram consideradas?

42. existem condições para o feedback do usuário?
43. existem condições pra se fazer mudanças?
44. eventuais modificações no sistema causarão mudança difíceis ou dispendiosas na documentação?
45. existem condições adequadas para distribuição das alterações na documentação?
46. a documentação pode ser reproduzida facilmente?
47. a cópia dos documentos pode ser evitada ou controlada?
48. existem pessoas disponíveis para suplementar a documentação?
49. existem condições adequadas para distribuição da documentação?
50. os usuários e os produtores concordam com os objetivos da documentação?
51. existem condições adequadas para manter pessoal de suporte informado e atualizado?
52. existem recursos disponíveis (por exemplo, microfichários, terminais) para ler / acessar / armazenar esses materiais?
53. os documentos foram devidamente aprovados?
54. os documentos estão organizados adequadamente em relação à documentação como um todo?
55. os documentos fazem referência a outros documentos que podem ser usados como *follow-up*?
56. o conteúdo da documentação é adequado?
57. todos os tópicos essenciais estão completos?
58. os tópicos irrelevantes foram eliminados?
59. os tópicos completos, estão no nível de detalhe adequado?
60. o nível técnico é apropriado ao nível do documento?
61. o documento destinado existe algum erro?
62. no documento existe alguma contradição?
63. a evidência é adequada para sustentar a apresentação?
64. a evidência é realista?
65. existe uma declaração definida de metas da documentação?
66. as metas são consistentes?
67. a apresentação da documentação parece autoritária?
68. os materiais constantes da documentação estão claros?
69. os diagramas, quadros ou outros materiais visuais estão claros?

70. são usados sempre que necessários?
71. são relevantes onde foram usados?
72. contribuem para a compreensão?
73. são claramente representados?
74. a uma quantidade apropriada de informação?
75. a terminologia está clara?
76. são consistentes por todos os documentos?
77. são de acordo com os padrões?
78. as definições estão corretas?
79. as definições estão claras?
80. o glossário está completo?
81. existe terminologia técnica em excesso?
82. quantidade apropriada de informação?
83. o estilo de redação é claro?
84. os parágrafos expressam apenas idéias correlatas e nada mais?
85. as unidades lógicas maiores são segmentadas em subtítulos?
86. o nível do texto é alto demais para a audiência?
87. o documento dirige-se ao leitor típico?
88. ele faz você dormir?
89. existe algum resumo?
90. os documentos contêm referências adequadas?
91. existe um índice, se necessário?
92. o índice está bem colocado?
93. existe um índice remissivo, se necessário?
94. o índice remissivo está bem colocado?
95. o índice está correto?
96. o índice remissivo está correto?
97. as referências às páginas são precisas?
98. existe verbetes para os tipos de consulta que os usuários farão?

99. os verbetes encontram-se sob os títulos corretos?
100. existem títulos alternados para os verbetes que podem ser acessados através de terminologia diferente?
101. os verbetes importantes e não-importantes para os mesmos termos estão diferenciados?
102. os termos estão segmentados adequadamente, ou existem muitas referências às páginas sob um único termo, indicando a necessidade de subcategorias?
103. existem verbetes supérfluos?
104. existe uma bibliografia de publicações de pré-requisitos?
105. se não existem pré-requisitos, isto está expresso?
106. a bibliografia está onde ela será encontrada antes que se tente ler todo o documento?
107. as referências estão completas o suficiente para localizar as publicações?
108. existem comentários para ajudar o leitor a escolher a publicação adequada?
109. existe uma bibliografia de publicações complementares que podem conter informação adicional?
110. se o documento for a única fonte de informação, isto está expresso?
111. as referências são completas o suficiente para a localização das publicações?
112. existe comentários para ajudar o leitor a escolher a publicação adequada?
113. a organização dos documentos propriamente contribui para facilitar a localização das informações?
114. a numeração das páginas é sensata?
115. a numeração das páginas é completa?
116. existe uma esquemática de programa?
117. documentos-padrão são usados?
118. a documentação disponível está atualizada?
119. existem documentos específicos para pedidos de modificação/correção?
120. são mantidos procedimentos para controlar todos os documentos relativos ao sistema da qualidade?
121. os procedimentos para controle da documentação contemplam?
122. a determinação dos documentos sujeitos ao controle?
123. a aprovação e emissão de procedimentos?
124. o controle de mudança nos documentos?
125. o controle de documentação é aplicado a documentos que descrevem o sistema da qualidade aplicado ao ciclo de vida do software?

126. documentos de produto incluindo entradas para cada fase de desenvolvimento, saídas de cada fase de desenvolvimento, planos de verificação e validação e respectivos resultados, manuais e documentação de manutenção?
127. há procedimentos para assegurar que os documentos e respectivas cópias estejam disponíveis em locais apropriados onde operações essenciais para o efetivo funcionamento do sistema da qualidade são desempenhadas?
128. há procedimentos para que documentos obsoletos sejam removidos dos locais de uso?
129. há procedimentos para a revisão e aprovação de mudanças nos documentos?
130. há controle das versões dos documentos?
131. os documentos são reemitidos após determinado número de alterações?
132. como os requisitos são documentados?
133. as interfaces organizacionais entre os diversos grupos estão definidas de maneira satisfatória?
134. como a informação entre grupos é documentada, transmitida e periodicamente revista?
135. o programa é suficientemente documentado?
136. o vendedor proporciona manutenção?
137. há uma qualidade da documentação dos procedimentos manuais do usuário?
138. há um sistema da qualidade documentado, considerando todo o ciclo de vida, assegurando que a qualidade seja verificada durante o processo?
139. o sistema da qualidade documentados está efetivamente implementado?
140. os elementos e requisitos do sistema da qualidade estão claramente documentados numa estrutura lógica coerente?
141. para cada projeto de software existe um plano da qualidade baseado no sistema da qualidade?
142. este plano da qualidade é entendido e observado pelas pessoas e áreas da organização envolvidas com o projeto?