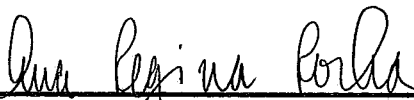


CONTROLE DA QUALIDADE DE SOFTWARE FINANCEIRO

Arnaldo Dias Belchior

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

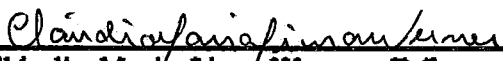
Aprovada por:



Prof. Ana Regina Cavalcanti da Rocha, D.Sc.
(presidente)



Prof. Paulo Afonso Lopes da Silva, Ph. D.



Cláudia Maria Lima Werner, D.Sc.

RIO DE JANEIRO, RJ - BRASIL
JULHO DE 1992

BELCHIOR, ARNALDO DIAS

Controle da Qualidade de Software Financeiro [Rio de Janeiro] 1992

XII, 183 p., 29.7 cm, (COPPE/UFRJ, M. Sc., ENGENHARIA DE SISTEMAS
E COMPUTAÇÃO, 1992)

TESE – Universidade Federal do Rio de Janeiro, COPPE

1 – Engenharia de Software/Qualidade de Software I

I. COPPE/UFRJ II. Título(Série).

A Jesus e a Maria

*“Pedi e dar-se-vos-á; buscai e achareis;
batei e abrir-se-vos-á” (Mt 7,7).*

Agradecimentos

Ao Banco do Nordeste do Brasil (BNB) sem o qual este trabalho não teria sido possível.

À professora Ana Regina pelo incentivo e orientação deste trabalho.

Aos professores Paulo Afonso e Cláudia Werner pela participação na banca examinadora.

À Fátima e a meus filhos Arnaldo e Mairon um especial agradecimento pela paciência e cooperação nesta caminhada.

A meus pais e irmãs pela amizade e pelo apoio ao longo de minha vida.

À Vera Werneck pela prestimosa ajuda neste trabalho.

Elisa Andréa, José Domingues Ferreira, Júlio Ciavaglia, Marília de La Cal, Morganna Diniz, obrigado pela amizade e o apoio recebidos.

Aos colegas da Agência BNB-Rio de Janeiro pelo carinho e amizade, especialmente Alberto Amorin, Ana Paula Amorin, Glauce Borba, Márcio Malta, Sebastião Costa, e Zélia Barbosa.

Aos colegas e aos funcionários da COPPE/Sistemas pelo apoio recebido, especialmente Cláudia Susie, Cristina Passos, Márcia Clímaco, Neide Santos e Teresa Aguiar.

Às empresas que me receberam e tornaram possível a realização da pesquisa de campo.

À Cidade Maravilhosa, que me acolheu nos braços abertos do Cristo Redentor e me enfeitou.

Resumo da Tese apresentada à COPPE como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M. Sc.)

Controle da Qualidade de Software Financeiro

Arnaldo Dias Belchior

Julho de 1992

Orientador: Ana Regina Cavalcanti da Rocha

Programa: Engenharia de Sistemas e Computação

Este trabalho identifica os atributos de qualidade de maior relevância ao se considerar a qualidade de software para a área financeira. São apresentados os resultados de uma pesquisa de campo em instituições financeiras, apoiada em um modelo para avaliação da qualidade de software. A princípio, foi feito um estudo dos atributos de qualidade, na literatura, que melhor se adequassem às exigências da área financeira. O estudo e análise desses resultados objetivam auxiliar desenvolvedores e usuários na construção de sistemas mais confiáveis, produzindo software financeiro de qualidade. Este trabalho poderá ajudar, também, na análise da qualidade de sistemas financeiros já desenvolvidos.

Abstract of Thesis presented to COPPE as partial fulfillment of the requirements for the degree of Master of Science (M. Sc.)

Financial Software Quality Control

Arnaldo Dias Belchior

July, 1992

Thesis Supervisor: Ana Regina Cavalcanti da Rocha

Department: Computing and Systems Engineering

This thesis identifies the quality attributes more relevant considering the quality of software for the financial area. The results of a field research in financial institutions are presented, supported by a model for software quality evaluation. At first, a study was done to identify the quality attributes found in literature that could be better adjusted to the financial area. The study and analysis of these results aim to help the developer and user in building more accurately systems, producing financial software with quality. This thesis may also help in analysing financial software already developed.

Índice

I	INTRODUÇÃO	1
I.1	A Engenharia de Software	1
I.1.1	A Crise do Software	2
I.2	Em Busca da Qualidade	3
I.3	Objetivo da Tese	5
I.4	Conteúdo da Tese	5
II	QUALIDADE DE SOFTWARE	8
II.1	Enfoques sobre Qualidade	9
II.2	A instalação do Processo de Qualidade	11
II.2.1	Custos com a Qualidade	11
II.2.2	A Satisfação do Usuário	13
II.2.3	A Relevância das Especificações	14
II.2.4	Enfoques sobre Projeto	15
II.2.5	Qualidade — o alvo a ser atingido	16
II.3	O Gerenciamento da Qualidade	17
II.3.1	Fatores Humanos	18

II.4	Controle da Qualidade de Software	18
II.4.1	Medidas da Qualidade do Software	21
II.5	Métodos para Avaliação da Qualidade de Software	24
II.5.1	Método de Rocha	25
II.6	Atributos para Avaliação da Qualidade de Programa	27
II.6.1	Notação Utilizada	28
II.6.2	Objetivo Confiabilidade da Representação	30
II.6.3	Objetivo Utilizabilidade	42
II.6.4	Objetivo Confiabilidade Conceitual	52
II.7	Conclusão	59
III	APLICAÇÕES FINANCEIRAS	60
III.1	Aplicações Administrativas	60
III.2	Administração Financeira	61
III.2.1	O Significado das Finanças nas Empresas	62
III.2.2	O Papel da Administração Financeira	62
III.2.3	Funções do Gerente Financeiro	63
III.2.4	Capital de Curto Prazo na Estrutura Financeira	64
III.2.5	Capital de Longo Prazo na Estrutura Financeira	65
III.2.6	Aplicações Financeiras	66
III.2.7	Análise Financeira	67
III.3	O Software Financeiro Atual	67
III.4	Conclusão	68

IV METODOLOGIA DE PESQUISA	69
IV.1 Pesquisa de Campo	69
IV.1.1 Hipóteses e Objetivos da Pesquisa de Campo	70
IV.1.2 Definição dos Elementos da População	71
IV.1.3 Processo de Investigação	72
IV.1.4 Técnica de Coleta de Dados	73
IV.1.5 Seleção das Empresas	74
IV.2 Tratamento Estatístico dos Dados	75
IV.3 Conclusão	77
V CARACTERÍSTICAS DE UM SOFTWARE FINANCEIRO DE QUALIDADE	78
V.1 Instituições Financeiras Pesquisadas	78
V.2 Coleta dos Dados	79
V.3 Análise dos Dados Coletados	80
V.3.1 Critérios do Objetivo Confiabilidade da Representação	81
V.3.2 Critérios do Objetivo Utilizabilidade	83
V.3.3 Critérios do Objetivo Confiabilidade Conceitual	86
V.3.4 Classificação dos Critérios obtidos dos Analistas de Sistema	88
V.3.5 Classificação dos Critérios obtidos dos Usuários	94
V.3.6 Classificação dos Critérios obtidos no Consolidado	97
V.3.7 Classificação dos Atributos de Qualidade por Subfator	102
V.3.8 Classificação dos Subfatores obtidos dos Analistas de Sistema	106

V.3.9	Classificação dos Subfatores obtidos dos Usuários	108
V.3.10	Classificação dos Subfatores obtidos do Consolidado	109
V.3.11	Classificação dos Subfatores/critérios obtidos dos Analistas de Sistema	110
V.3.12	Classificação dos Subfatores/critérios obtidos dos Usuários . .	110
V.3.13	Classificação dos Subfatores/critérios obtidos do Consolidado .	116
V.4	Avaliação das Hipóteses	120
V.5	Conclusão	126
VI	CONCLUSÃO	127
VI.1	Trabalhos Futuros	129
	Referências Bibliográficas	130
	Anexos	139
A	Carta de Recomendação	139
B	Questionário dos Analistas de Sistema	140
C	Manual dos Analistas de Sistema	146
D	Questionário dos Usuários	154
E	Manual dos Usuários	159
F	Tabelas Descritivas dos Atributos de Qualidade	163

Lista de Figuras

II.1	Estrutura do Método para Avaliação da Qualidade de Software — Fonte: [ROCHA 87]	26
II.2	Qualidade de Programas — Fonte: [ROCHA 87]	29
II.3	Confiabilidade da Representação de Programas	32
II.4	Utilizabilidade de Programas	43
II.5	Confiabilidade Conceitual de Programas	54
B.1	Quadro 01 – Analista de Sistema – Requisitos de Qualidade de Soft- ware Financeiro — Confiabilidade da Representação	143
B.2	Quadro 02 – Analista de Sistema – Requisitos de Qualidade de Soft- ware Financeiro — Utilizabilidade	144
B.3	Quadro 03 – Analista de Sistema – Requisitos de Qualidade de Soft- ware Financeiro — Confiabilidade Conceitual	145
D.1	Quadro 04 – Usuário – Requisitos de Qualidade de Software Finan- ceiro — Confiabilidade Conceitual	157
D.2	Quadro 05 – Usuário – Requisitos de Qualidade de Software Finan- ceiro — Utilizabilidade	158

Lista de Tabelas

V.1 Instituições Financeiras Pesquisadas	79
V.2 Características da Implementação dos Sistemas Pesquisados	79
V.3 Tabela dos Sistemas Financeiros Pesquisados	80
V.4 Critérios do Objetivo Confiabilidade da Representação	81
V.5 Critérios do Objetivo Utilizabilidade	84
V.6 Critérios do Objetivo Confiabilidade Conceitual	86
V.7 Classificação dos critérios obtidos dos Analistas de Sistema	89
V.8 Classificação dos critérios obtidos dos Usuários	95
V.9 Classificação dos critérios obtidos no Consolidado	98
V.10 Classificação dos Atributos de Qualidade por Subfator	103
V.11 Classificação dos Subfatores obtidos dos Analistas de Sistema	107
V.12 Classificação dos Subfatores obtidos dos Usuários	108
V.13 Classificação dos Subfatores obtidos do Consolidado	109
V.14 Classificação dos subfatores/critérios obtidos dos Analistas de Sistema	111
V.15 Classificação dos subfatores/critérios obtidos dos Usuários	114
V.16 Classificação dos subfatores/critérios obtidos do Consolidado	117

V.17	Comparação dos valores do desvio padrão, por subfator, dos <i>Sistemas de Fundos de Investimento</i> com o <i>Consolidado</i>	124
V.18	Comparação dos valores do desvio padrão, por subfator, dos <i>Sistemas de Open Market</i> com o <i>Consolidado</i>	125
F.1	Médias dos critérios do Objetivo Confiabilidade da Representação . .	164
F.2	Médias dos critérios do Objetivo Utilizabilidade	165
F.3	Médias dos critérios do Objetivo Confiabilidade Conceitual	166
F.4	Médias ordenadas dos critérios obtidos dos Analistas de Sistema . . .	167
F.5	Médias ordenadas dos critérios obtidos dos Usuários	169
F.6	Médias ordenadas por critério obtidos do Consolidado	170
F.7	Médias dos Subfatores	172
F.8	Médias ordenadas dos Subfatores obtidos dos Analistas de Sistema .	173
F.9	Médias ordenadas dos Subfatores obtidos dos Usuários	174
F.10	Médias ordenadas dos Subfatores obtidos do Consolidado	175
F.11	Médias ordenadas dos subfatores/critérios obtidos dos Analistas de Sistema	176
F.12	Médias ordenadas dos subfatores/critérios obtidos dos Usuários . . .	179
F.13	Médias ordenadas dos subfatores/critérios obtidos do Consolidado . .	181

Capítulo I

INTRODUÇÃO

I.1 A Engenharia de Software

Os computadores tornaram-se indispensáveis em nosso cotidiano. É inconcebível imaginar o funcionamento de nossa sociedade sem a presença marcante, virtualmente insubstituível, dos mesmos. No entanto, o computador não faz milagres, e somente executa o que lhe é mandado fazer – tudo é uma questão de planejamento e não de prestidigitação.

Os computadores precisam ser bem programados, uma vez que não resolvem os problemas sozinhos. Os resultados obtidos serão, no máximo, tão bons quanto forem estes programas. Neste contexto, não é mais tolerado o uso de métodos artesanais na elaboração de programas, pois é necessário assegurar um nível elevado de qualidade no software produto [STAA 87].

As exigências na confecção de um software produto tomaram novas dimensões, uma vez que este passou a ser visto não como uma obra de arte ou fruto de artifícios engenhosos, mas sim inseridos em um contexto científico. Surgiu, então, a *Engenharia de Software*, decorrente do conjunto de conhecimentos obtidos, mediante observações e experiências de fatos, fundamentada nos princípios da ciência da computação.

“Engenharia de Software é o desenvolvimento de princípios, técnicas, heurísticas e disciplinas gerenciais, visando o desenvolvimento racional e econômico

de artefatos intensivos em software¹, onde esses artefatos são concluídos dentro dos limites de recursos, são concluídos de acordo com o cronograma, operam economicamente em ambientes de produção reais, produzem serviços de qualidade aceitável, sempre que solicitados, e são capazes de evoluir mantendo e ou aprimorando a qualidade do serviço prestado” [STAA 91].

A *Engenharia de Software* vem desenvolvendo mecanismos, para enfrentar a chamada *crise do software*, desencadeada a algumas décadas, e caracterizada pelo elevado custo do software sobre o hardware.

I.1.1 A Crise do Software

Muitos fatores têm contribuído para a crise atual do software [NEIGHBORS 89]:

- *a proporção preço/execução do hardware tem decrescido 20 % ao ano;*
- *o custo do software, como uma percentagem do custo total do sistema, tem aumentado, enquanto que o custo do hardware continua diminuindo;*
- *os computadores, com seus preços em declínio, podem ser usados em muitas áreas de aplicação, que, por sua vez, necessitam de vários produtos de software;*
- *a produtividade, no processo de criação de software, nos últimos anos, tem sido incrementada de 3 % a 8 % a.a, incluindo todo o desenvolvimento em engenharia de software e desenvolvimento de linguagem de alto-nível.*

Os principais problemas surgidos, em decorrência do uso dos computadores e do desenvolvimento de sistemas, vêm sendo, largamente, pesquisados. Foram coletadas e destacadas as seguintes questões em [WERNECK 90] e [PASSOS 91]:

- *ausência de infra-estrutura de hardware e de software, a falta de infra-estrutura organizacional, além do não uso de metodologias de desenvolvimento de sistemas;*

- *falta de procedimentos normalizados de avaliação da qualidade, para a obtenção do grau de confiabilidade e do desempenho desejado;*
- *comprometimento na qualidade, em virtude das dificuldades em satisfazer as necessidades dos usuários, com relação às estimativas de tempo e de prazo;*
- *documentação desatualizada ou inexistente dos sistemas, dificultando o uso e a manutenção dos mesmos;*
- *elevada carga de manutenção, gerando a deterioração da qualidade do produto;*
- *desatualização ou despreparo de mão-de-obra especializada;*
- *aumento constante nos custos de desenvolvimento, causados pela baixa produtividade.*

A *crise do software*, alinhada ao incremento da complexidade dos programas, tem exigido soluções mais efetivas, que forneçam um suporte sistemático para a elaboração, a verificação, a validação e a manutenção dos mesmos. Deste modo, desencadeou-se o processo de busca da qualidade de software.

I.2 Em Busca da Qualidade

Não se pode pensar em qualidade de um produto como sinônimo de perfeição. Trata-se de algo factível, relativo, substancialmente dinâmico e evolutivo, amoldando-se à granularidade dos objetivos a serem atingidos. Considerá-la como algo absoluto e definitivo seria transportar-se para o inatingível e, com base neste sofisma, propiciar entraves a qualquer esforço de produzi-la.

Inicialmente, uma empresa deve despertar e desenvolver a convicção de que o desenvolvimento de software, sem controle da qualidade, é improdutivo e penoso, quando não uma relação desgastante entre usuários e desenvolvedores. Posteriormente, a instituição deve decidir-se, formalmente, pelo processo da qualidade, em uma demonstração clara dos objetivos a seguir e transparência em suas atitudes.

É necessário que se leve a sério os requisitos de qualidade. Os vícios adquiridos, ao longo do tempo, devem ser evitados. Desde o princípio da concepção do software, deve-se procurar a maneira correta de desenvolvê-lo, reavaliando-se, continua e criteriosamente, cada etapa deste processo, com o objetivo final de incremento da confiabilidade do produto. Uma vez adquirida a consciência da efetividade, a eficácia da qualidade, e o firme propósito de implementá-la, só então a empresa estará madura o suficiente, para obter êxito nesta escalada.

O software deve ser projetado e construído com qualidade, para que seus objetivos sejam alcançados. Tempo e custo são as principais variáveis dos grandes projetos, todavia não são tão importantes quanto o controle da qualidade em aplicações militares, sistemas de avaliação civil comercial, sistemas de controle nuclear, que afetam a vida de muitas pessoas [CAVANO 87].

A subjetividade e a experiência também são aplicadas na determinação da qualidade de software. Para ajudar a solucionar o problema, necessitamos de um método para avaliação da qualidade de software, através do qual possamos realizar medições quantitativas, baseadas em uma análise objetiva. O principal objetivo do estudo das medidas do software não é só a determinação estática de suas propriedades, mas sim uma predição científica durante o ciclo de vida do produto [BEAUFOND 87].

Funcionalidade, desempenho, confiabilidade e custos são, provavelmente, as características mais importantes a serem avaliadas em um produto. O cálculo da confiabilidade, durante o seu ciclo de vida, é essencial, pois permite determinar se as necessidades e as expectativas do usuário estão sendo atendidas concretamente [CARDOSO 90].

Um pré-requisito para a produção de qualquer software deveria ser um alto nível de qualidade. Entretanto, isto implica em grandes investimentos, além da conscientização e organização da equipe. Por conseguinte, caberá à própria empresa instituir processos de garantia da qualidade ou de controle da qualidade, além de determinar os níveis satisfatórios dos mesmos [ANDRADE 91].

A preocupação e a motivação com a qualidade de um produto de

software é cada vez maior em nossos dias. Este trabalho trata, especificamente, sobre do tema Controle da Qualidade de Software Financeiro.

I.3 Objetivo da Tese

O objetivo maior desta tese é colaborar no esforço de produzir-se software financeiro de qualidade. Apresenta-se, portanto, os resultados de uma pesquisa de campo, nesta área, apoiada em um modelo para avaliação de software.

Descreve-se uma metodologia de trabalho, desenvolvida na busca de delinear as principais características de um software financeiro. Entenda-se por software financeiro, aquele que é capaz de auxiliar e orientar o analista financeiro e ou o gerente de finanças de uma empresa na tomada de suas decisões, fornecendo sempre informações confiáveis, claras e em tempo hábil.

O estudo e a análise dessas características visam auxiliar os desenvolvedores na construção de produtos de software mais confiáveis, para a área financeira. Salientam-se os pontos, sobre os quais devem ser exercidos maior ou menor esforço, permitindo traçar uma estratégia de trabalho mais produtiva e voltada para o usuário.

Esta pesquisa, também, poderá auxiliar os usuários, na área em questão, desejosos de uma tempestividade acentuada, em suas solicitações, e exigindo sempre uma confiabilidade, a toda prova, das informações geradas, a terem uma visão crítica dos produtos, com os quais estão lidando ou que pensam adquirir.

I.4 Conteúdo da Tese

Esta tese constitui-se de seis capítulos e de seis anexos, descritos, resumidamente, a seguir:

CAPÍTULO I – Introdução

Comenta-se sobre a engenharia de software e a busca pela quali-

dade de software. Define-se o escopo do trabalho, descrevendo-se, sucintamente, a abrangência de cada capítulo.

CAPÍTULO II – Qualidade de Software

Discorre-se sobre a qualidade de software, evidenciando-se a sua importância. Descrevem-se critérios para a avaliação de um software, baseando-se na estrutura de um método, para a avaliação da qualidade. A escolha desses critérios foi direcionada no sentido de atender às especificidades de um software financeiro.

CAPÍTULO III – Aplicações Financeiras

Procura-se mostrar a relevância da área financeira dentro de uma empresa e, conseqüentemente, a grande importância, que deve ser dispensada na confecção de seus produtos.

CAPÍTULO IV – Metodologia de Pesquisa

Apresenta-se uma metodologia de pesquisa, para validar os critérios de qualidade, utilizando-se instrumentos metodológicos indispensáveis ao alcance de resultados confiáveis. A pesquisa de campo realizada recebe um tratamento estatístico, propiciando a verificação das hipóteses levantadas.

CAPÍTULO V – Características de um Software Financeiro de Qualidade

Com o objetivo de colaborar no esforço de gerar software financeiro de qualidade, apresenta-se as características deste produto, na visão de analistas de sistemas e de usuários da área.

CAPÍTULO VI – Conclusão

As conclusões do trabalho e suas possíveis implementações em computador são, aqui, resumidas.

ANEXO A – Carta de Recomendação

Carta enviada às instituições financeiras, selecionadas para a rea-

lização da pesquisa de campo.

ANEXO B – Questionário dos Analistas de Sistemas

Questionário de controle de qualidade e quadros submetidos aos analistas de software financeiro das instituições pesquisadas.

ANEXO C – Manual dos Analistas de Sistemas

Manual contendo as explicações dos requisitos de qualidade de software selecionados para a área financeira, e aplicados a seus analistas de sistema.

ANEXO D – Questionário dos Usuários

Questionário de controle de qualidade e quadros submetidos aos usuários de software financeiro das instituições pesquisadas.

ANEXO E – Manual dos Usuários

Manual contendo as explicações dos requisitos de qualidade de software selecionados para a área financeira, e aplicados a seus usuários.

ANEXO F – Tabelas Descritivas dos Atributos de Qualidade

Tabelas contendo a análise descritiva dos atributos de qualidade de software financeiro.

Capítulo II

QUALIDADE DE SOFTWARE

“ Vivemos em uma economia mundial e, no futuro, a qualidade não será apenas algo bonito de se ter. Será o preço necessário de admissão ao mercado. Os clientes já não precisarão pressionar para receberem o que pediram. Considerarão isto um direito seu. Assim sendo, as empresas que têm ocupado o tempo, aperfeiçoando a arte de explicar porque as coisas não estão nunca de acordo com o prometido, não terão futuro ” [CROSBY 90].

Qualidade é um atributo associado a algo, não sendo possível defini-la universalmente. Deve, portanto, ser instanciada para um determinado produto. Qualidade é, também, um conceito multidimensional, realizando-se por intermédio de um conjunto de atributos ou características. As empresas responsáveis pelo desenvolvimento de software devem assumir a responsabilidade de estabelecerem este nível aceitável de qualidade e meios para verificarem se foi alcançado. Qualidade de software é, por conseguinte, um conjunto de propriedades a serem satisfeitas, em determinado grau, de modo que o software satisfaça as necessidades de seus usuários [ROCHA 87].

Qualidade de software é o grau para que o software processe uma combinação desejada de atributos. Para realizar uma alta qualidade de software, em um sistema, essa combinação desejada de atributos deve ser claramente definida, senão a qualidade passa a ser intuição. Cada resultado intermediário, no processo de produção, deve ser examinado imediatamente após sua conclusão, procurando garantir que os erros e as inadequações no produto sejam detectados o mais cedo

possível. A qualidade final do produto, portanto, é uma função de todas as fases anteriores do ciclo de desenvolvimento.

A qualidade do software depende da satisfação do usuário com o serviço fornecido pelo software e com a facilidade de seu uso. Mas existem diversos tipos de usuários (*operador, mantenedor, auditor, etc.*) e a qualidade obtida depende da satisfação das necessidades e das expectativas de todos esses usuários, que embora pertencentes a uma mesma categoria, podem observar a qualidade de modos diversos [BEAUFOND 87].

Portanto, para que se possa produzir um software de qualidade, é necessário identificar que atributos determinam a qualidade desejada, estabelecendo seu universo de abrangência. No entanto, é preciso, de uma maneira satisfatória, adotar procedimentos, que controlem essa qualidade, com base na abordagem utilizada.

II.1 Enfoques sobre Qualidade

A *American Society for Quality Control*, que existe há quase 50 anos, e se reúne, anualmente, em conferências, não tem chegado a um acordo sobre a definição de qualidade [CROSBY 90].

“ *Qualidade é a condição necessária de aptidão, para o fim a que se destina* ”. Esta definição é da Organização Européia de Controle de Qualidade [EUROPEAN 72]. É necessário indicar com exatidão este “ *fim* ” no desenvolvimento do software e adequá-lo a nível de projeto. Quais as reais necessidades do usuário? De que ele, verdadeiramente, necessita? O produto final é viável e confiável no espaço de tempo esperado pelo usuário? “ *A condição necessária* ” não se limita a uma definição própria de qualidade, para as possíveis situações específicas, durante as fases do ciclo de vida do software.

Para se ter uma visão geral da qualidade, em função de sua organização e abrangência, [PALADINI 90] descreve as várias abordagens feitas no trabalho de [GARVIN 84]:

a) Abordagem Transcendental

- A qualidade, neste contexto, não chega a alcançar um nível abstrato e inatingível, mas, dificilmente, pode ser definida com precisão. É uma característica, que só pode ser, efetivamente, constatada na prática, na maioria vezes, proveniente da experiência acumulada, e não somente de estudos e de análises realizados no produto. No que diz respeito à qualidade de software, esta abordagem já está em desuso.

b) Abordagem Centrada no Produto

- A qualidade é vista como passível de medição, através dos atributos do produto. Um produto de qualidade teria uma maior quantidade de atributos de melhores características. Há um método de avaliação de qualidade, baseado em padrões objetivos, cujas unidades de medidas são bem definidas – dá-se um caráter preciso à qualidade.

c) Abordagem Centrada no Valor

- A qualidade é visualizada no ângulo do custo de produção versus desempenho do produto. A qualidade é evidenciada no produto de custo aceitável e de melhor desempenho.

d) Abordagem Centrada na Produção

- O produto deve ser desenvolvido de acordo com suas especificações básicas, determinadas a nível de projeto. A qualidade é alcançada, quando se atende, plenamente, às especificações, desde a sua primeira confecção.

e) Abordagem Centrada no Usuário

- A qualidade é alcançada, quando atende, prontamente, às necessidades e às conveniências do usuário. O usuário é a fonte de toda a avaliação sobre a qualidade do produto.

Qualidade, portanto, não significa somente excelência ou outro atributo de um certo produto final. A qualidade deve ser perseguida dentro da organização, pois, com certeza, é o que os usuários esperam de um produto. No entanto,

a implementação do processo de qualidade, em uma empresa, requer, sobretudo, vontade política de seus dirigentes e, para que isto aconteça, é necessário que estes estejam despertados para este propósito.

II.2 A instalação do Processo de Qualidade

Na instalação do processo de qualidade é necessário que se estabeleça uma anistia geral, ou seja, o que aconteceu até aquele momento, não é culpa de ninguém. A principal meta é enfrentar os problemas existentes em benefício de todos. Contudo, devem ser contínua e dinamicamente avaliados e melhorados a estratégia geral da empresa, o planejamento e os meios disponíveis com vistas ao estabelecimento de condições favoráveis, com o fim de se alcançar os objetivos específicos. Portanto, é essencial que todo o processo de desenvolvimento do produto contribua para a qualidade do mesmo.

A primeira atitude, que as empresas têm de tomar, para alcançarem mudanças necessárias, é estabelecerem uma estrutura organizacional, que proporcione alicerce, orientação e apoio para a implementação do processo de aperfeiçoamento da produtividade e da qualidade. A mudança começa de cima e, naturalmente, irá se disseminar para cada nível inferior na estrutura da organização [HRADESKY 89].

II.2.1 Custos com a Qualidade

A empresa precisa certificar-se das vantagens geradas pela instalação e manutenção do processo de qualidade. É essencial que ela saiba o quanto custa melhorar e manter a qualidade desejada de seus produtos. Além disso, é importante que estejam bem identificadas todas as áreas, que devem ser atacadas nesse processo, para que sejam reduzidos os custos.

A qualidade afeta a economia de uma empresa de duas maneiras básicas [NOVELINO 89]:

- *efeitos no custo*: é preciso capital inicial para se implantar a qualidade;

- *efeitos na renda*: com o aumento da qualidade, a empresa torna-se mais competitiva no mercado.

Os custos com a qualidade dividem-se nas seguintes categorias, de acordo com [BERGAMO 91]:

- *custos de prevenção*: relacionam-se, diretamente, com as equipes de desenvolvimento, implementação e manutenção do sistema de qualidade, incluindo, também, os custos das ações preventivas efetuadas durante o processo produtivo;
- *custos de avaliação*: associados às atividades de controle, de avaliação ou de auditoria de produtos ou de serviços, para assegurarem se estão em conformidade com as especificações ou são adequados ao uso;
- *custos de falhas internas*: ligados aos custos com produtos ou com serviços, que não estão em conformidade com as especificações, ou não são adequados ao uso, assim como os custos de análise das falhas;
- *custos de falhas externas*: gerados por produtos, que não atendem às especificações da qualidade, após sua entrega ao usuário.

Entre as dificuldades, que surgem na implementação e no gerenciamento do sistema de custos da qualidade, são citadas [BERGAMO 91]:

- *interpretação generalizada de que os custos da qualidade são custos do controle da qualidade*;
- *falta de apoio dos gerentes de nível mais alto, para a implementação das técnicas de custo de qualidade*;
- *falta de interesse e, conseqüentemente, pouca cooperação dos departamentos da empresa na obtenção dos dados*;
- *os custos da qualidade envolvem, praticamente, todos os departamentos da organização, enquanto que as despesas do controle de qualidade estão sob controle do seu gerente*;

Para um produto de software, os custos estão, estreitamente, relacionados com o grau de segurança desejada. Entenda-se, aqui, por segurança, como sendo a característica de um produto de software evitar falhas de funcionamento, que possam provocar danos materiais ou humanos. De acordo com [PRESSMAN 87] a segurança e a qualidade de software envolvem:

- *análise, projeto, codificação, métodos de teste e ferramentas;*
- *revisões técnicas formais, que são aplicadas durante cada passo na engenharia de software;*
- *controle da documentação do software e das mudanças realizadas na mesma;*
- *assegurar o cumprimento das normas de desenvolvimento de produtos de software na empresa;*
- *mecanismos de medição da qualidade do produto.*

II.2.2 A Satisfação do Usuário

Um programa pode estar muito bem elaborado, mas se não funcionar satisfatoriamente, todas as suas demais qualidades, tornam-se irrelevantes para os usuários [LONGWORTH 85].

A consolidação do processo da qualidade em uma organização é lenta e depende de sua estrutura. Para que isto aconteça é necessário que a gerência perceba que a satisfação do usuário está direta e intrinsecamente relacionada com a lucratividade do produto. Uma vez que, a viga mestra da qualidade é o usuário, então esta deve ser perseguida tanto quanto o próprio lucro.

A satisfação do usuário no começo, fim e sempre é, indubitavelmente, a meta a ser alcançada. A redução de custos é um subproduto da melhoria da qualidade. No entender de [CROSBY 90], um entusiasta por qualidade observa os seguintes itens:

- *decide-se por querer uma estratégia de zero defeitos, isto é, procura fazer aquilo que concorda, quando concorda fazê-lo;*
- *anuncia uma política de qualidade clara e específica;*
- *mostra o compromisso da gerência com a qualidade;*
- *certifica-se de que cada um tenha a formação necessária de forma a poder desempenhar suas funções;*
- *elimina as oportunidades de transigir com a conformidade;*
- *convence a cada um de que todos dependem uns dos outros;*
- *satisfaz o usuário no começo, no fim, e sempre.*

As necessidades do usuário devem ser todas, corretamente, traduzidas dentro das especificações do produto. Este propósito é o ponto central para o conceito de qualidade [MANN 88].

II.2.3 A Relevância das Especificações

As especificações devem fornecer aos usuários a noção exata daquilo que irão receber. Quando as especificações não são levadas a sério é inevitável que se perca mais tempo, aqui e acolá, executando-se remendos, quando a preocupação maior deveria ser evitá-los. A busca da qualidade propõe-se eliminar situações erradas, e não simplesmente acomodá-las, atuando como agente de prevenção [CROSBY 90].

A maneira de formalizar as especificações é de grande importância, e deve ser efetuada com o uso adequado das palavras, evitando-se uma deficiência de comunicação e, por conseguinte, diluição dos esforços empregados. A realização de nossos objetivos virá em consequência do cumprimento das especificações estabelecidas.

As especificações são respostas às perguntas realizadas e acordos resultantes dessas respostas. Portanto, devem ser respeitadas e jamais alteradas, a não ser por intermédio de um novo acordo entre as partes, que as elaboraram.

As especificações devem ser claras e passíveis de implementação. Não deve existir especificação, que não possa ser entendida ou cumprida. Às vezes, ocorre de não sabermos ainda tudo o que é necessário, todavia precisamos começar e avançar, definindo-se experiências e investigações. Isto resultará no entendimento da situação, e conseqüente formalização dos requisitos [CROSBY 90].

Portanto, as especificações devem retratar as reais necessidades do usuário. O seu teor conceitual e metodológico deve ser resguardado nos ajustes sofridos pelas pressões de custo e do cronograma estabelecido. Conformação com exceções aqui ou com transigências acolá pode resultar em uma deterioração geral da qualidade do produto. Inconsistências mal resolvidas na fase de especificação, e deixadas para serem corrigidas na fase de projeto ou em etapas posteriores, seguramente, comprometerão a qualidade do produto gerado.

II.2.4 Enfoques sobre Projeto

O projeto do software é uma tarefa árdua. O projetista deve recordar e, com freqüência, modificar seu modelo corrente, guardando caminhos de questões e de pendências, que ainda requerem sua atenção. Constantemente, ele analisa e testa o estado do projeto atual, quanto à correção e à adequação, e ainda direciona seus cuidados em resolver novos problemas emergentes, de importância não menos relevante.

O projeto de software não é um processo meramente mecânico, mas uma atividade humana, que requer um esforço mental considerável. Por isto, é necessário manter o projeto dentro dos limites do entendimento humano, definindo-se a sua solução em termos de uma estrutura de entidades (módulos), que descrevam uma função simples do contexto da estrutura total [PASSOS 91].

A gerência de projeto, no desenvolvimento de software, está sempre submetida a pressões, exercidas pela urgência do produto ou pelo poder de barganha dos usuários. Muitas questões importantes, algumas vezes, são postergadas na esperança de que possam ser decididas posteriormente. [FAIRLEY 85] diz que a ausência do planejamento é causa determinante de atrasos, de altos custos e de

baixa qualidade.

O planejamento do projeto não é estático, uma vez que modificações corretivas ou evolutivas podem ser introduzidas. As questões ainda não consensuais precisam ser planejadas e suas incógnitas tomam formas, com as quais se pode lidar. É indispensável que alterações estruturais no projeto realmente as especificações, que o originaram, mantendo, assim, a integridade no processo de produção do software, preservando a qualidade do mesmo.

II.2.5 Qualidade — o alvo a ser atingido

Vários fatores podem ser considerados na produção de um software com alto nível de qualidade [VICENT 88]:

- *uso de uma linguagem de programação adequada ao tipo da área de aplicação e ao produto em questão;*
- *uso de ambientes de suporte a programação;*
- *obediência aos padrões estabelecidos pela empresa;*
- *ferramentas automatizadas de apoio ao desenvolvimento de software;*
- *apoio da gerência;*
- *condições e recursos de trabalho adequados;*
- *comprometimento da equipe de desenvolvedores com a qualidade.*

É primordial ressaltar a grande importância, que possui a documentação do sistema, tanto a externa quanto a interna. Isto fornece apoio para a avaliação das características de qualidade do mesmo, além de servir de lastro para manutenções posteriores [ANDRADE 91].

Os objetivos e a política da empresa, na busca da qualidade, devem ser firmes e duradouros e não sofrerem mudanças diante das mais diversas ou adversas situações. Na concepção de [BANKS 89], o gerenciamento da qualidade é uma função organizacional responsável pela prevenção de defeitos.

II.3 O Gerenciamento da Qualidade

Pode-se distinguir dois estados, relacionados à qualidade, quanto aos seus aspectos do gerenciamento [CROSBY 90] e [BERGAMO 91]:

- **Estado arcaico**

- *a gerência não entende de qualidade e culpa o controle de qualidade pelos problemas de qualidade;*
- *a qualidade é uma característica desejável;*
- *a técnica gerencial mais usada é o “ apagar o fogo ”;*
- *custos da qualidade é um conceito desconhecido e, por conseguinte, não aplicado;*
- *não existe um sistema formal da qualidade;*
- *o produto é o principal enfoque.*

- **Estado avançado**

- *a gerência entende a função qualidade;*
- *a qualidade é definida em conformidade com os requisitos;*
- *a qualidade é conseguida através de prevenção;*
- *os custos da qualidade estão implantados e são gerenciados;*
- *existe um sistema formal da qualidade;*
- *o processo é o principal enfoque.*

A formação de pessoal tem seu papel preponderante na melhoria da qualidade. É indispensável que todos visualizem as questões em foco de uma forma homogênea, sob um mesmo prisma, direcionando-se sempre para o âmago do problema, a fim de somarem esforços.

II.3.1 Fatores Humanos

A equipe responsável pela qualidade do desenvolvimento do produto deve estar integrada ao processo global da própria instituição, para que seus feitos sejam implementados adequadamente. O ideal é que se consiga falar uma linguagem comum sobre qualidade entre gerentes e empregados, dentro da empresa. As pessoas não devem sentir-se obrigadas ou coagidas a se engajarem na missão da qualidade, mas fazê-lo porque, assim, o desejam.

Na garantia da qualidade o fator humano é fundamental. Os elementos envolvidos devem estar motivados para uma participação efetiva, conscientizados da importância do trabalho em equipe. Para que isto aconteça, é necessário levar, em consideração, os seguintes fatores [NOVELINO 89]:

- *objetivos da qualidade*: devem ser bem definidos e serem extensivos a todos os níveis de decisão;
- *comunicação dos objetivos da qualidade*: divulgação dos objetivos, simultaneamente, a todos os níveis hierárquicos;
- *participação efetiva*: os envolvidos no processo, devem ter ciência e consciência de suas obrigações com relação à qualidade;
- *motivação de pessoal*: deve ser uma atitude dinâmica e contínua.

O gerenciamento da qualidade de software está diretamente envolvido no processo de garantia da qualidade do produto. A garantia da qualidade envolve, entre outros, procedimentos de controle de qualidade, que permitem a verificação do nível de qualidade, que há em determinado produto.

II.4 Controle da Qualidade de Software

Todo o processo de desenvolvimento é ajustado de modo que se produza de acordo com as especificações do sistema. O controle de qualidade do software verifica o nível de qualidade, em um determinado produto de software, através de comparações das

características deste, com um padrão pré-estabelecido. Cada processo de produção exibe variabilidade. Este aspecto é de tal forma reconhecido que, no desenvolvimento de qualquer produto, sempre se permite uma determinada tolerância.

A qualidade de software deve ser planejada dentro de um projeto, envolvendo seus produtos e ser bem controlada. O objetivo da qualidade, além da necessidade de ser claramente definido, deve ser efetivamente monitorado, e rigorosamente seguido. A qualidade deve ser planejada dentro da estrutura do projeto, constantemente avaliada, e correções aplicadas, quando deficiências são identificadas [EVANS 87].

Baseando-se em [PARANTHAMAN 90], pode-se citar os principais objetivos do controle da qualidade:

- *avaliar os padrões da qualidade, ao longo do processo de desenvolvimento do produto;*
- *julgar a conformidade do processo a padrões estabelecidos e tomar a ação apropriada, quando são notados desvios;*
- *avaliar a qualidade ótima possível de ser obtida sob as condições dadas;*
- *melhorar qualidade e produtividade por meio de controle de processos e da experimentação;*
- *desenvolver procedimentos para se estabelecerem boas relações entre desenvolvedores e usuários;*
- *desenvolver a consciência da qualidade, tanto dentro como fora da organização.*

Definir e medir qualidade vai propiciar um método quantificável de controle do software. As etapas recomendadas para o controle de qualidade de software de acordo com [ARTHUR 85] e [BASILI 87] são:

- *seleção do produto:* é feita a seleção do produto de software, sobre o qual será realizado o processo de Controle de Qualidade;

- *escolha da medida a ser aplicada*: as métricas de qualidade são especificadas, e pode-se definir os valores a serem atingidos para essas medidas;
- *medição*: as métricas de qualidade especificadas são aplicadas no produto selecionado;
- *interpretação das medidas*: os resultados obtidos são comparados com padrões pré-estabelecidos, indicando o nível de qualidade das características avaliadas;
- *utilização das informações*: as informações, resultantes das etapas anteriores, são utilizadas para melhorarem a qualidade do produto a um nível satisfatório.

Para que o processo do controle de qualidade tenha o sucesso esperado, pode-se citar alguns procedimentos [CARD 88]:

- *atribuir a responsabilidade do processo a uma pessoa ou a uma equipe*;
- *construir um banco de dados comum, que armazene os resultados das avaliações realizadas, formando assim, um histórico para possíveis comparações e referências*;
- *proporcionar um conjunto básico de ferramentas para facilitar as atividades de coleta de dados*.
- *gerar e distribuir relatórios periódicos, destacando procedimentos considerados relevantes*;

Os programas de controle de qualidade ajudam a construir um sistema de informações, que é de fundamental importância para melhorar a qualidade do produto gerado. Pode-se listar vários benefícios de um programa de controle de qualidade de um produto de software:

- *melhoria da qualidade do produto*;
- *aumento da produtividade da equipe desenvolvedora*;
- *redução do custo do produto*;

- *redução de manutenções corretivas do software;*
- *redução dos gargalos de produção;*
- *garantia da eficiência operacional do software;*
- *conscientização da equipe a respeito da qualidade;*
- *menor número de reclamações dos usuários.*

Para que o processo de controle de qualidade seja, de fato, vantajoso, viável e utilizável, é preciso que se usem ferramentas automatizadas. A coleta manual de dados, durante a avaliação, é um processo tedioso, propenso a erros, além de demandar muito tempo. Portanto, ferramentas automatizadas implicam na melhoria da coleta e da análise dos dados, oriundos da aplicação das métricas, na redução dos custos desse processo e na disponibilidade mais rápida dos resultados sempre que for necessário [ANDRADE 91].

Portanto, não basta, apenas, identificar que atributos determinam a qualidade do produto final, mas, também, identificar que procedimentos adotar, no sentido de controlar o processo de desenvolvimento, de forma a atingir o nível de qualidade desejado, em cada fase. Esse processo é realizado através da aplicação de métricas de qualidade, que são medidas ou avaliações das características de qualidade do produto.

II.4.1 Medidas da Qualidade do Software

O uso de medidas, de uma maneira organizada e projetada, possui efeito benéfico, tornando os desenvolvedores conscientes da importância do gerenciamento e dos compromissos para com a qualidade do produto. É necessário que a Engenharia de Software disponha de parâmetros de medidas precisas e bem entendidas.

A medição dá um caráter científico ao processo de avaliação da qualidade de software. No entanto, para se realizar uma avaliação desta qualidade, de uma maneira satisfatória, pode-se levar em consideração tanto métricas quantitativas, quanto exames subjetivos feitos diretamente no código fonte [ADAM 88].

As métricas de qualidade são orientadas, para serem aplicadas sobre os produtos de software, em determinados pontos de seu processo de desenvolvimento, sendo indicadores do nível de qualidade. A utilização das métricas de qualidade também pode prever evoluções no projeto de software. A aplicação de um bem planejado conjunto de medidas, organizado de tal maneira a atender às características e aos requisitos do tipo de produto, que está sendo desenvolvido, pode fornecer um quadro consistente do seu estado real, em qualquer ponto do processo de desenvolvimento de acordo com [DOBBINS 87] e [INCE 88].

O uso de métricas de qualidade de software, apoiado por uma metodologia padrão, oferece as seguintes vantagens de acordo com [CERINO 86] e [IEEE 87]:

- *estabelecer requisitos de qualidade para um sistema, desde o princípio de seu desenvolvimento;*
- *estabelecer critérios de aceitação e de padronização;*
- *desenvolver um plano de medidas, baseado nos requisitos estabelecidos;*
- *avaliar o nível de qualidade realizado, confrontando-o com os requisitos estabelecidos;*
- *melhorar o gerenciamento do produto, oferecendo meios de serem detectados anomalias ou pontos potenciais de problemas no sistema, ao longo do desenvolvimento;*
- *predizer o nível de qualidade, que será realizado no futuro;*
- *comparar os atributos de qualidade de um sistema com outro;*
- *monitorar a degradação da qualidade, durante a fase de manutenção;*
- *previsão do custo do produto ao longo de seu ciclo de vida.*

Para se alcançar os objetivos do controle da qualidade de software, no plano das medidas do sistema, usa-se [IEEE 87]:

- *medidas do processo*: utilizadas para mensurarem características dos métodos, das técnicas e das ferramentas empregadas na aquisição, no desenvolvimento, na verificação, e na operação dos produtos de software;
- *medidas do produto*: utilizadas para mensurarem as características da documentação e do código.

A utilização de métricas de qualidade também oferece algumas limitações [CONTE 86]:

- *a medição deve ser consistente, com o mínimo de subjetividade, e apoiada em definições precisas, de modo que a análise dos dados não seja prejudicada;*
- *alguns ambientes requerem um ajuste das métricas utilizadas, para que se reduzam as possibilidades de falhas no processo de avaliação;*
- *as métricas ajudam no processo de tomada de decisão, todavia não substituem o gerente;*
- *as métricas avaliam o desempenho do produto e não da equipe técnica.*

Medida é uma escala de valores possíveis, que corresponde às variações observadas em uma determinada característica. Além da escala de medidas, necessita-se de métodos para levantamento e validação dos dados. A escala de métricas pode ser classificada em [CARD 90]:

- *nominal*: apoiada nas características elementares, como, por exemplo a cor ou o tipo;
- *ordinal*: baseada em um grau de importância, variando do pior para o melhor;
- *intervalo*: envolve distâncias iguais entre valores, significando diferenças equivalentes em medidas de qualidade, como, por exemplo, o tempo;
- *razão*: possui a propriedade de intervalo acrescida de um ponto zero, com o objetivo de indicar que nenhuma característica foi atingida.

Os métodos para medições, baseados no julgamento humano, são chamados de subjetivos; os baseados em pesos e em operações aritméticas são denominados de objetivos. Somente os tipos de medida *nominal* ou *ordinal* podem ser coletados através de questionários ou através da experiência. Neste caso, o conteúdo preciso da medida pode não ser reproduzido, porque observadores diferentes ou o mesmo observador em situações diferentes, importam razões diversas.

Segundo [BASILI 87], no método a ser utilizado, não é importante quem desempenha a parte de medição do controle da qualidade, mas sim como este é planejado e avaliado. Isto deve ser feito, de preferência, por pessoas que não participaram do desenvolvimento do produto.

II.5 Métodos para Avaliação da Qualidade de Software

Evidenciada a importância da avaliação da qualidade de software, fazem-se necessários métodos, que viabilizem esta avaliação e controle. Esses métodos envolvem um conjunto de definições e de procedimentos para que sejam avaliadas as características de qualidade de um determinado produto de software.

O uso de métodos para avaliação da qualidade de software nos permite [BACHE 88]:

- *compreender os relacionamentos existentes entre os atributos de qualidade do produto avaliado;*
- *explorar o efeito de alguma mudança de um atributo de qualidade sobre outros;*
- *fazer a previsão dos atributos de qualidade de um produto;*
- *utilizar procedimentos de avaliação;*
- *caracterizar uma situação específica e descrever propriedades individuais de um determinado atributo de qualidade.*

O método escolhido para avaliação da qualidade de Software Financeiro, objeto deste trabalho, é o proposto por [ROCHA 83], que está baseado nos trabalhos de Boehm [BOEHM 78] e de McCall [McCALL 79].

II.5.1 Método de Rocha

Este Método para avaliação da qualidade de software está baseado nos seguintes conceitos [ROCHA 87]:

1. Objetivos da qualidade:

- *são as propriedades gerais, que o produto deve possuir;*

2. Fatores de qualidade:

- *determinam a qualidade na visão dos diferentes usuários do produto — usuário final e outros;*
- *podem ser compostos por subfatores, quando estes, por si só, não definem completamente um objetivo;*

3. Critérios:

- *são atributos primitivos, possíveis de serem avaliados;*

4. Processos de avaliação:

- *determinam o processo e os instrumentos a serem utilizados, de forma a se medir o grau de presença, no produto, de um determinado critério;*

5. Medidas:

- *são o resultado da avaliação do produto, segundo os critérios;*

6. Medidas agregadas:

- *são o resultado da agregação das medidas obtidas ao se avaliar de acordo com os critérios, e quantificam os fatores.*

RELAÇÕES QUANTITATIVAS:

RELAÇÕES LÓGICAS

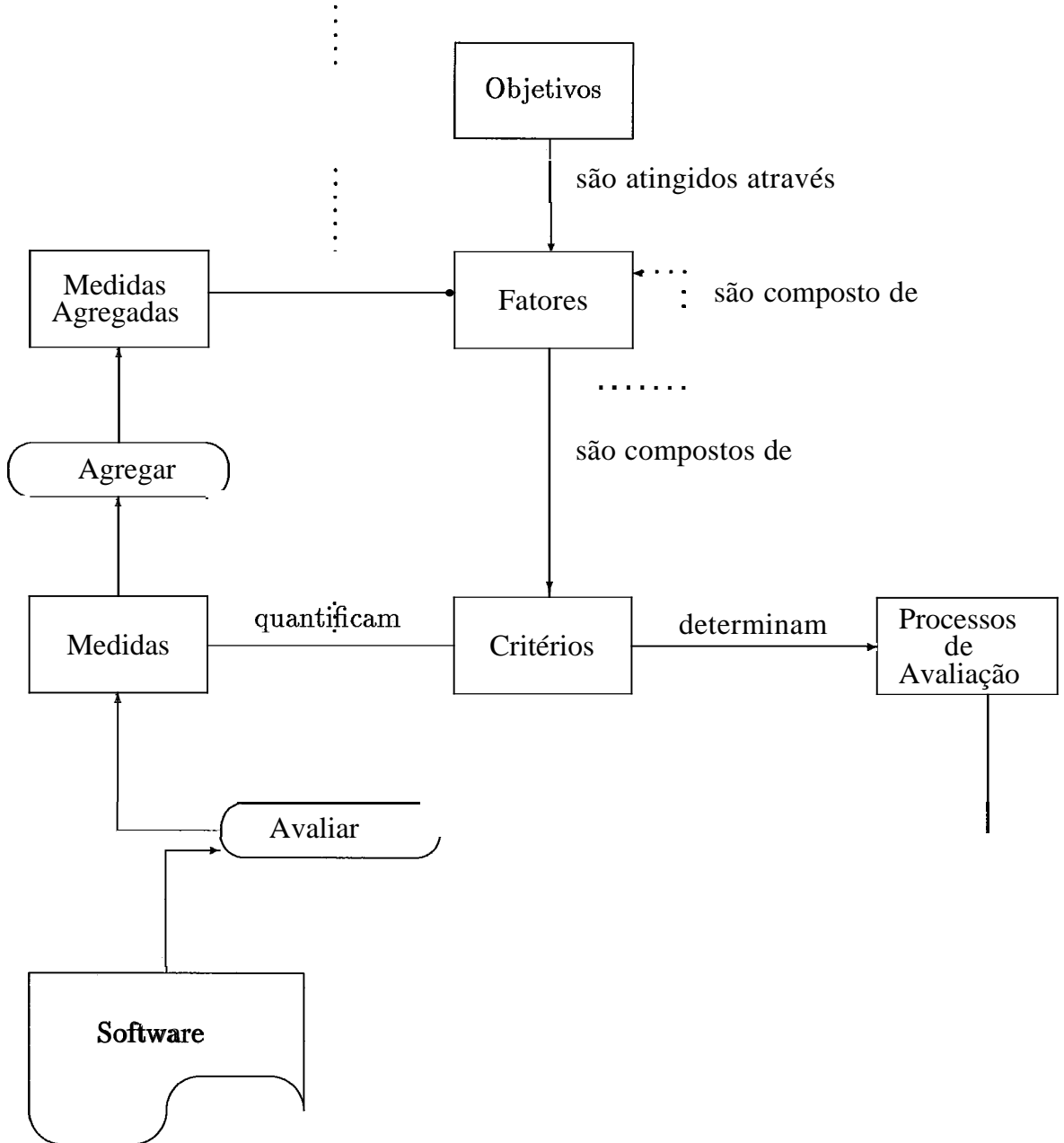


Figura 11.1: Estrutura do Método para Avaliação da Qualidade de Software —
Fonte: [ROCHA 87]

A figura II.1 mostra, a seguir, a estrutura deste método.

Os objetivos de qualidade são atingidos através dos fatores de qualidade, que podem ser compostos por outros fatores. Objetivos e fatores não são, diretamente, mensuráveis e só podem ser avaliados através de critérios. Um critério é um atributo primitivo. Nenhum critério isolado é uma descrição completa de um determinado fator ou subfator. Da mesma maneira, nenhum fator define completamente um objetivo.

Medidas são valores resultantes da avaliação de um produto, segundo um critério específico.

Este método está sendo utilizado para definir objetivos, fatores e subfatores para todas as fases do desenvolvimento de software. Neste trabalho, o método foi utilizado para definir atributos e processos de avaliação, com o objetivo de traçar o perfil de qualidade de sistemas da área financeira de empresas, aqui, denominados de Software Financeiro.

Os atributos para avaliação da qualidade de programas foram selecionados, tendo-se em vista a sua aplicabilidade para a avaliação da qualidade de software financeiro. Fundamentou-se em pesquisas sobre administração e análise financeira (enfoques sobre este assunto no capítulo seguinte) e buscas, na literatura, sobre qualidade de software. Procurou-se adequar esta, àquela através de experiências no desenvolvimento de sistemas para a área financeira, agregadas a contatos frequentes com especialistas financeiros.

II.6 Atributos para Avaliação da Qualidade de Programa

Os produtos de software são desenvolvidos para atenderem a determinadas necessidades dos usuários. Depois de serem colocados em operação, espera-se que tenham uma vida útil longa e produtiva. Para que isto se concretize, devem ser atingidos os seguintes objetivos de qualidade: *utilizabilidade*, *confiabilidade conceitual* e *confiabilidade da representação* [ROCHA 87].

Utilizabilidade é um objetivo fundamental, pois não tem sentido se pensar em um programa, se este não pode ser utilizado. Refere-se às características de utilização do programa, sob as mais diversas formas, tanto durante a implementação como na operação ou na manutenção. Entretanto, para ser utilizado é necessário que o programa satisfaça às necessidades e aos requisitos, que motivaram sua construção. Deriva-se, então, a importância da *confiabilidade conceitual*.

A *confiabilidade conceitual* refere-se às características de entendimento e de correção, em relação ao conteúdo da documentação interna e do código fonte do programa, de modo que este satisfaça às suas especificações.

A *confiabilidade da representação* refere-se às características de entendimento e de manipulação com relação à descrição, à organização e à representação da documentação interna e do código fonte, de modo que o programa possa ser manipulado pelas mais diferentes pessoas, durante sua vida útil.

A figura II.2 mostra os fatores e os subfatores relacionados a cada objetivo de qualidade, conforme proposto em [ROCHA 87].

II.6.1 Notação Utilizada

A representação, abaixo, está inspirada na organização hierárquica sugerida por Warnier [WARNIER 87] para construção de programas.

$$\begin{array}{ccccccc}
 \textit{Programa} & \left\{ & \textit{Objetivos} & \left\{ & \textit{Fatores} & \left\{ & \textit{Subfatores} & \left\{ & \textit{Critérios} \\
 (1, p) & & (1, o) & & (1, f) & & (0, s) & & (0, c)
 \end{array}$$

A representação, acima, mostra uma estrutura hierárquica do conjunto de atributos de qualidade para um determinado produto de software. Cada conjunto subdivide-se, no nível imediatamente inferior, em outros subconjuntos. Registra-se o número de vezes, que cada categoria de subconjunto está presente no conjunto tomado como referencial. O subconjunto gerado, por sua vez, também, divide-se em outros subconjuntos, até a instância dos critérios.

Assim sendo, para a avaliação dos atributos de qualidade de um programa, para cada *objetivo* são definidos *fatores* de qualidade a ele relacionados.

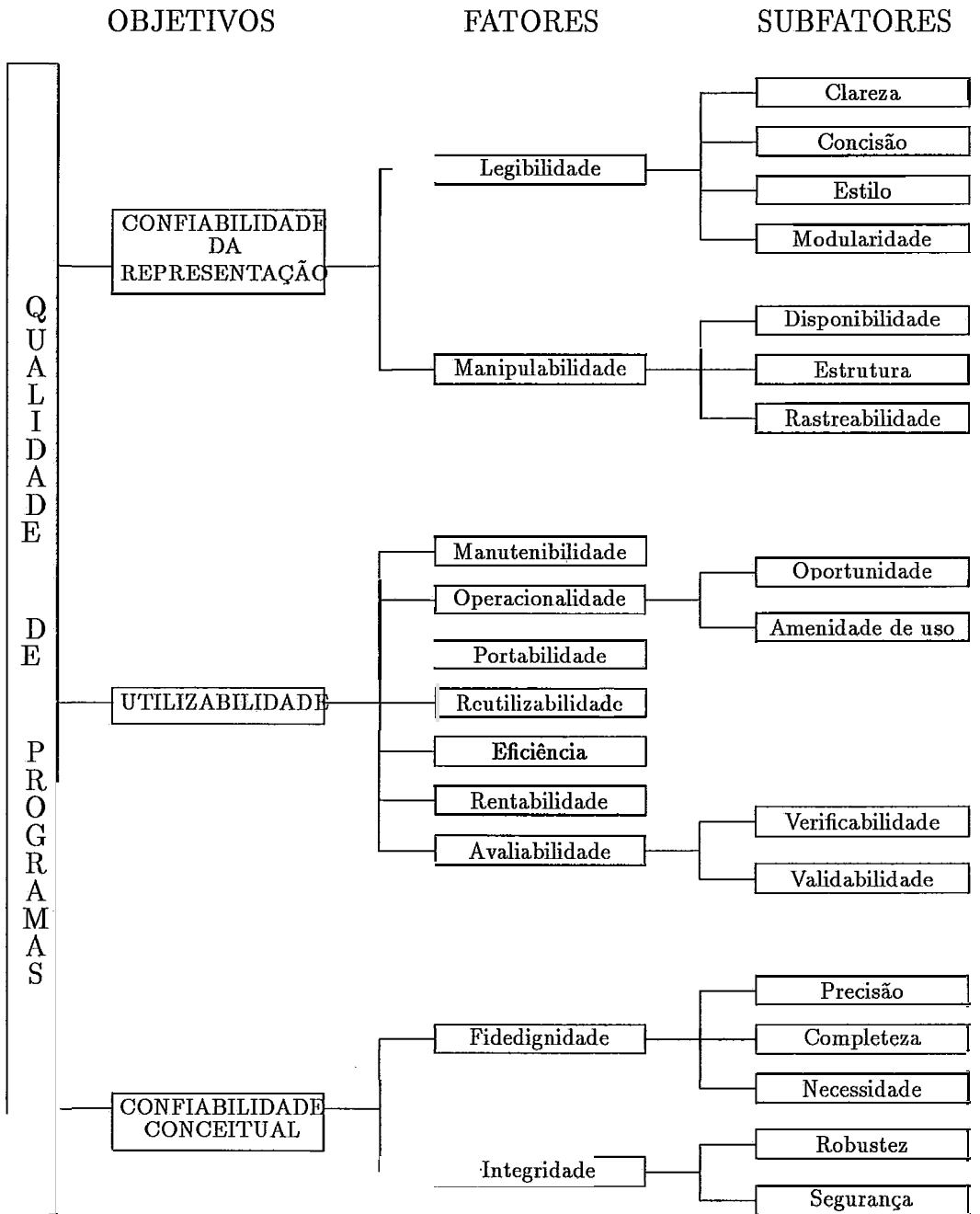


Figura II.2: Qualidade de Programas — Fonte: [ROCHA 87]

Para cada *fator* são definidos *subfatores*. Para cada *subfator* são definidos *critérios*. Os *critérios* são atributos de qualidade primitivos avaliáveis.

Objetiva-se determinar a qualidade de um determinado programa — *científico, educacional, financeiro*, etc. Existem de 1 a o objetivos de qualidade para o programa em questão. Cada objetivo contem de 1 a f fatores. Cada fator contem de 0 a s subfatores. Cada subfator contem de 0 a c critérios.

Nomeamos os programas por P ; os objetivos por O ; os fatores por F ; os subfatores por S ; os critérios por C . Usaremos a seguinte notação, para identificarmos um atributo de qualidade, dentro do conjunto de atributos de qualidade de programa:

$$PpOoFfSsCc \quad (II.1)$$

Como, normalmente, estamos interessados somente na determinação da qualidade de um determinado software — *financeiro*, por exemplo — a notação anterior pode ficar reduzida a:

$$OoFfSsCc \quad (II.2)$$

Assim, se tivermos a notação: $O1F1S1C2$ ou $O1.F1.S1.C2$ significaria, por exemplo, como veremos a seguir, $O1$ (*objetivo da confiabilidade da representação*), $F1$ (*fator: legibilidade*), $S1$ (*subfator: clareza*), $C2$ (*critério: uso de terminologia uniforme*).

II.6.2 Objetivo Confiabilidade da Representação

Os analistas de sistemas têm uma preocupação adicional em relação aos usuários: é a representação do software em linguagem inteligível pelo computador. A disponibilidade do código fonte, traduzida na facilidade de acessá-lo e, sobretudo, de estar

sempre atualizado é indispensável. O ideal seria ter, também, a documentação sempre atualizada e organizada de uma maneira tal, que houvesse facilidade de acesso à mesma.

Durante a fase de codificação, os projetistas, de uma maneira geral, produzem seu trabalho baseado em suas experiências e no estilo próprio de programação. Isto poderia ser até aceitável se, na fase de manutenção, o projetista estivesse sempre disponível para fazer as alterações necessárias no programa, lembrando-se daquilo que fez, e porque fez. Isto sem levar em consideração outros fatores agravantes, tais como a insatisfação profissional, os acidentes, as doenças, as mudanças nas atividades profissionais, que poderiam afetar, seriamente, a qualidade dos programas desenvolvidos [CARDOSO 90].

Portanto, é de fundamental importância, que os programas sejam legíveis e de fácil manipulação, sendo produzidos de acordo com as normas de elaboração de código fonte da empresa. Isto evitaria que apenas seus próprios desenvolvedores pudessem manipulá-los com destreza — chamados *os donos do programa*. A qualidade do programa gerado é muito importante, não apenas para garantir sua funcionalidade e sua eficiência, mas, também, para facilitar seu entendimento por outros desenvolvedores, que precisem utilizá-lo. Sabe-se, por exemplo, que a maioria das alterações em sistemas financeiros exigem uma tempestividade acentuada, seja para atender a determinações governamentais ou pela própria competitividade do mercado.

A figura **II.3** mostra os atributos selecionados para o objetivo *confiabilidade da representação* de programa.

O1. Objetivo – Confiabilidade da representação: refere-se às características de representação do programa, que afetam sua compreensão e manipulação por pessoas [FREITAS 85]. É atingido através dos *fatores: legibilidade e manipulabilidade*.

F1. Fator – Legibilidade: o programa deve ser escrito de tal forma, que permita o seu entendimento, de maneira fácil, por quem não participou

OBJETIVO FATORES SUBFATORES CRITÉRIOS

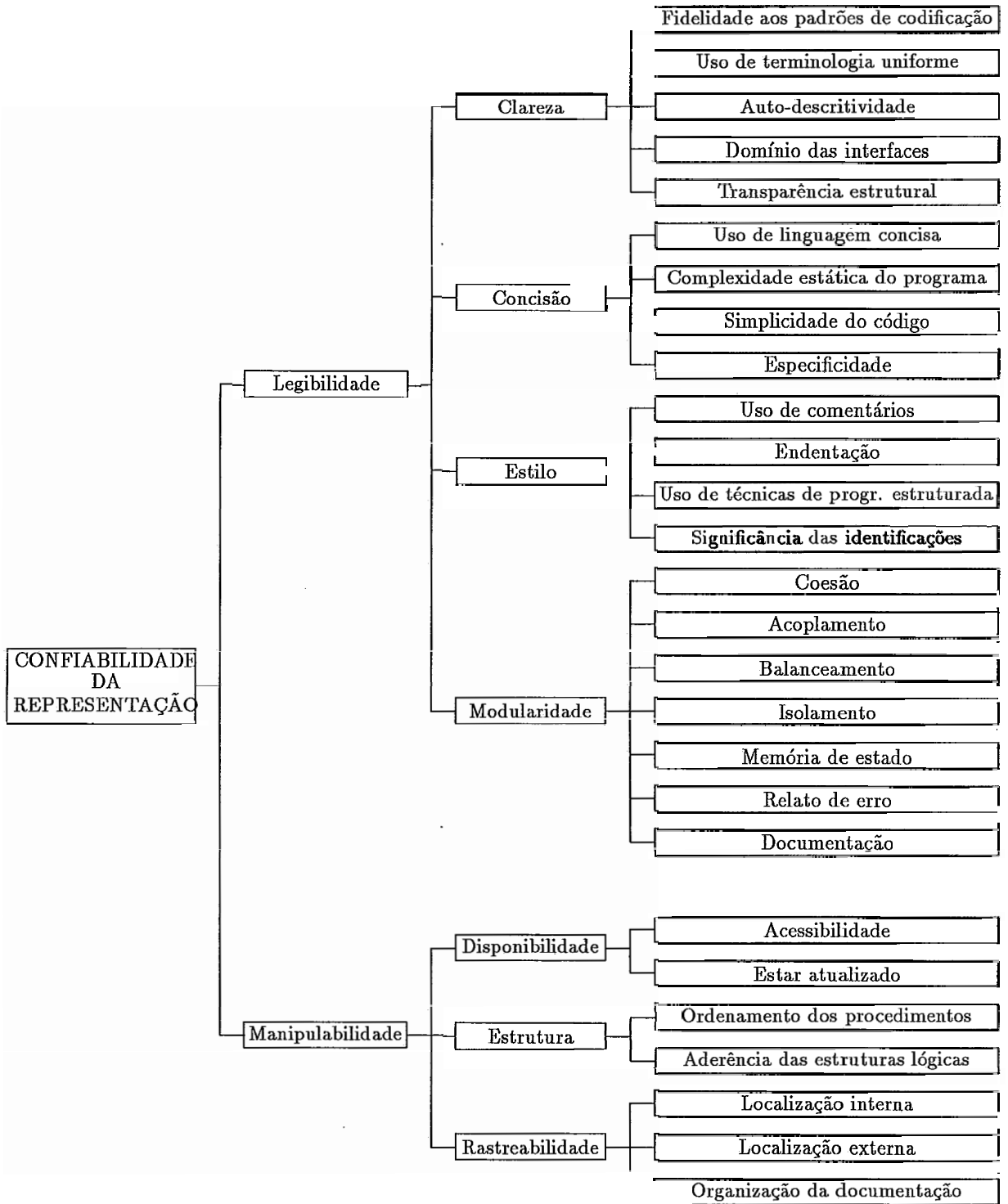


Figura II.3: Confiabilidade da Representação de Programas

de seu desenvolvimento [ROCHA 87]. É atingido pelos pelos *subfatores: clareza, concisão, estilo e modularidade.*

S1. Subfator – Clareza: é a característica de um programa ter suas funções codificadas na forma mais clara possível, e isento de práticas, que o tornem complexo e de difícil entendimento [RADC-TR-85-37]. É avaliado através dos *critérios: fidelidade aos padrões de codificação, uso de terminologia uniforme, auto-descritividade, domínios das interfaces, transparência estrutural.*

C1. Critério – Fidelidade aos padrões de codificação: o programa está codificado de acordo com o padrão da organização desenvolvedora.

- *Processo de Avaliação – certifica-se que:*
 - *o programa está codificado dentro dos padrões de codificação da organização.*

C2. Critério – Uso de terminologia uniforme: nome dado às entidades do programa, tais como variáveis, blocos, arquivos, e outras que devem ter terminologia uniforme [PALERMO 88].

- *Processo de Avaliação – certifica-se que:*
 - *o programa possui terminologia uniforme, para referenciar suas entidades.*

C3. Critério – Auto-descritividade: característica que um programa deve ter de tal forma que o leitor do código possa determinar seus objetivos, restrições, entrada/saída, procedimentos e controle de versões [ROCHA 90].

- *Processo de Avaliação – certifica-se que:*
 - *o programa apresenta identificação e comentários dos objetivos e restrições dos módulos, sub-rotinas e procedimentos, no início de cada um deles, bem como de suas entradas/saídas;*
 - *o programa possui controle de alterações de cada módulo, anexada na documentação e no código fonte.*

C4. Critério – Domínios das interfaces: corresponde a presença

de um número conveniente de parâmetros passados ou recebidos [CARDOSO 90].

- *Processo de Avaliação – certifica-se que:*

- *o programa não possui parâmetros desnecessários;*
- *o programa utiliza a quantidade de parâmetros especificada em suas funções.*

C5. Critério – Transparência estrutural: o programa não contém referências a declarações, que não sejam utilizadas e nem usa campos auxiliares redundantes ou desnecessários.

- *Processo de Avaliação – certifica-se que:*

- *o programa não faz referências a declarações, que não sejam executadas;*
- *o programa não possui variáveis de trabalho redundantes ou desnecessárias.*

S2. Subfator – Concisão: característica de um programa que tem suas funções implementadas com a quantidade mínima de código. É avaliado através dos *critérios: uso de linguagem concisa, complexidade estática do programa, simplicidade do código e especificidade.*

C1. Critério – Uso de linguagem concisa: deve ser escolhida a linguagem de programação mais concisa para a aplicação em foco, pois o número de declarações ou comandos é o próprio código fonte do programa. Segundo [ARTHUR 85] as linguagens de 2ª geração, como Assembler, são menos concisas que linguagens de 3ª geração, como Fortran ou Cobol e, que por sua vez, são menos concisas que as de 4ª geração.

- *Processo de Avaliação – certifica-se que:*

- *o programa encontra-se codificado na linguagem mais adequada (mais concisa) à aplicação.*

C2. Critério – Complexidade estática do programa: baseia-se no número total de operadores e operandos, gerados ou estimados, quando o projeto está completo [HALSTEAD 77]. Um voca-

bulário extenso significa que o programador necessitará aprender mais palavras para poder entender, codificar e manter o sistema [ARTHUR 85].

- *Processo de Avaliação:*

- *feito pelas equações abaixo [HALSTEAD 77]:*

- . *Vocabulário* = $n_1 + n_2$;

- . *Dificuldade* = $n_1.n_2/2.N_2$.

- *definição das variáveis:*

- . n_1 = número de operadores distintos do programa;

- . n_2 = número de operandos distintos do programa;

- . N_1 = número total das ocorrências do operadores;

- . N_2 = número total das ocorrências dos operandos.

C3. Critério – Simplicidade do código: consiste em uma determinada função conter uma quantidade mínima necessária de comandos e de variáveis.

- *Processo de Avaliação – certifica-se que:*

- *o programa não possui trechos, que nunca serão executados, gerando dificuldades no entendimento do código.*

C4. Critério – Especificidade: condição em que a escolha de determinadas opções do programa não o conduza a estados indeterminados e que não tenham sido previstos [CARDOSO 90].

- *Processo de Avaliação – certifica-se que:*

- *os caminhos básicos dos módulos de um programa, é menor ou igual a 10. A equação, abaixo, avalia a complexidade ciclomática de um módulo do programa ou seja, o número de caminhos básicos deste módulo, através de seu grafo de controle [McCABE 76]:*

- . $V(G) = e - n + 2p$.

- *definição das variáveis:*

- . $V(G)$ = complexidade de um módulo.

- . e = número de ligações no grafo.

- . n = número de nós no grafo.
- . p = número de componentes conexos no grafo.

S3. Subfator – Estilo: refere-se à necessidade de o programa conter elementos de estilo, de forma que expresse seus objetivos e especificações de maneira clara, simples, elegante, organizada, direta, consistente e considerando as padronizações ou recomendações estabelecidas [PRESSMAN 87]. É avaliado através dos *critérios: uso de comentários, endentação, uso de técnicas de programação estruturada, significância das identificações.*

C1. Critério – Uso de comentários: comentários explicativos são inseridos no programa, para explicarem aquilo que não ficar óbvio apenas com a leitura do código — os comentários apresentam o propósito de um grupo de declarações do programa ou o que está acontecendo na lógica do mesmo [STAHL 88].

- *Processo de Avaliação – certifica-se que:*

- *os comentários estão visualmente destacados;*
- *os comentários foram atualizados a cada alteração do código;*
- *os comentários auto-documentam o programa, efetivamente;*
- *há um prólogo descrevendo o programa;*
- *há um prólogo descrevendo cada componente do programa.*

C2. Critério – Endentação: refere-se ao formato de codificação, que facilita, esteticamente, o entendimento da estrutura lógica do mesmo [STAHL 88].

- *Processo de Avaliação – certifica-se que:*

- *os itens similares estão alinhados em uma mesma coluna;*
- *as declarações encontram-se endentadas em relação às palavras-chaves introdutórias;*
- *os níveis de aninhamento estão, claramente, identificados;*
- *a endentação é consistente.*

C3. Critério – Uso de técnicas de programação estruturada: é a característica de o programa está escrito de acordo com as

normas e as técnicas da programação estruturada, que são utilizadas para construção de programas, através de estruturas lógicas de controle e de suas combinações [HEHL 86].

- *Processo de Avaliação – certifica-se que:*

- *o programa não fere as regras da programação estruturada.*

C4. Critério – Significância das identificações: os nomes dados às constantes, às variáveis, às funções, aos procedimentos e às sub-rotinas traduzem o seu significado de forma apropriada e mnemônica, e seguem algum tipo de padronização estabelecida pela empresa [BARGUT 86].

- *Processo de Avaliação – certifica-se que:*

- *o nome dado às constantes, às variáveis, às funções, aos procedimentos e às sub-rotinas, traduzem seu significado facilmente;*

- *os nomes dados aos identificadores do programa seguem a padronização de nomenclatura estabelecida pela organização [MARTIN 85].*

S4. Subfator – Modularidade: característica de um programa ter implementadas suas funções, através de uma estrutura de módulos altamente independente, e particionados logicamente [NASCIMENTO 86]. É avaliado pelos *critérios: coesão, acoplamento, balanceamento, isolamento, memória de estado, relato de erro, e documentação.*

C1. Critério – Coesão: é a força, que mantém unidos os elementos de um módulo. O objetivo desejado é o mais alto grau de coesão entre os módulos. A coesão se processa de acordo com a seguinte escala ascendente de níveis [PAGE-JONES 80]:

- *coesão por coincidência:* os elementos de um módulo não têm razão aparente para estarem juntos;
- *coesão lógica:* existe alguma relação lógica entre os elementos dos módulos;
- *coesão temporal:* o relacionamento mais forte entre os elemen-

tos do módulo é que eles são executados ao mesmo tempo;

- *coesão procedural*: os elementos estão envolvidos em atividades diferentes e, possivelmente, não relacionadas;
- *coesão comunicacional*: os elementos referenciam os mesmos dados de entrada e/ou saída;
- *coesão seqüencial*: a saída de um elemento é a entrada para o próximo;
- *coesão funcional*: todos os elementos estão relacionados ao desempenho de uma função.

Processo de Avaliação:

- *identificar o tipo de coesão em cada programa: coesão funcional, sequencial ou comunicacional dão aos programas boa coesão; coesão procedural ou temporal dão aos programas média coesão; coesão coincidental ou lógica dão aos programas baixa coesão (devem ser evitadas);*

C2. Critério – Acoplamento: é o grau de interdependência entre os módulos. O baixo acoplamento entre módulos indica um sistema bem particionado. Os tipos de acoplamento são descritos em ordem ascendente de níveis [PAGE-JONES 80]:

- *acoplamento de dados*: a comunicação entre os módulos é feita por parâmetros (dado elementar ou uma tabela homogênea);
- *acoplamento por dados estruturados*: a comunicação entre os módulos é feita através de estruturas de dados;
- *acoplamento por controle*: envolve a passagem de variáveis de controle, para o gerenciamento da lógica interna do outro módulo;
- *acoplamento por área comum de dados*: os módulos referem-se à mesma área global de dados;
- *acoplamento por conteúdo*: um módulo faz referências ao interior de outro módulo.

Processo de Avaliação:

- *identificar os tipos de acoplamento entre os programas chamado e chamador: são considerados bons programas, aqueles com acoplamento de dados, por área comum de dados, por dados estruturados, e por controle; deve ser evitado o acoplamento por conteúdo.*

C3. Critério – Balanceamento: uma estrutura é balanceada, se os módulos, de nível mais alto, tratam os dados de uma forma lógica, de modo que estes não sejam dependentes das características físicas [PAGE-JONES 80].

- *Processo de Avaliação – certifica-se que:*

- *os módulos de nível mais elevado do programa tratam os dados de uma forma lógica, independentemente das características físicas.*

C4. Critério – Isolamento: avalia tudo o que se precisa saber com relação à implementação do módulo, sem ser necessário conhecer suas características internas, para que seja possível utilizá-lo corretamente [PAGE-JONES 80].

- *Processo de Avaliação – certifica-se que:*

- *o que o módulo propõe-se a fazer é identificado, sem a necessidade de ser conhecida sua a estrutura interna.*

C5. Critério – Memória de estado: situação em que um módulo, todas as vezes que invocado, executa suas funções como se fosse a primeira vez, não possuindo memória de sua prévia existência [PAGE-JONES 80].

- *Processo de Avaliação – certifica-se que:*

- *um módulo executa sempre suas funções, como se fosse a primeira vez, independentemente de quando ou de quantas vezes seja processado.*

C6. Critério – Relato de erro: os erros são relatados pelo módulo, que os detecta e os identifica.

- *Processo de Avaliação – certifica-se que:*

– o módulo identifica e relata os erros.

C7. Critério – Documentação: a documentação do módulo deve estar disponível e possuir bom padrão de qualidade [PRESSMAN 87].

- *Processo de Avaliação – certifica-se que:*

– o módulo possui documentação disponível e de boa qualidade.

F2. Fator – Manipulabilidade: é a característica de um programa poder ser facilmente manipulado por diferentes pessoas [ROCHA 87]. É atingido pelos subfatores: *disponibilidade, estrutura e reatibilidade*.

S1. Subfator – Disponibilidade: característica de um programa e sua documentação estarem atualizados e prontos para uso, quando necessário. É avaliado através dos critérios: *acessibilidade e estar atualizado* [FREITAS 86].

C1. Critério – Acessibilidade: os programas fonte e suas respectivas documentações internas devem estar disponíveis e serem, facilmente, acessados por qualquer usuário leitor autorizado [RADC-TR-85-37].

- *Processo de Avaliação – certifica-se que:*

– o código fonte e a documentação são acessáveis com facilidade.

C2. Critério – Estar atualizado: o conteúdo do programa fonte deve retratar sua versão mais atualizada ou seja, conter as últimas alterações realizadas.

- *Processo de Avaliação – certifica-se que:*

– o programa fonte contém as últimas alterações realizadas;

– as atualizações realizadas estão, claramente, indicadas.

S2. Subfator – Estrutura: característica de um programa possuir um padrão definido de composição de suas partes, formando uma organização hierárquica [FREITAS 86]. É avaliado através dos critérios: *ordenamento dos procedimentos e aderência das estruturas lógicas*.

C1. Critério – Ordenamento dos procedimentos: os procedimentos devem estar dispostos de maneira ordenada, onde se

identifique, facilmente, as ações de entrada, de saída e de processamento lógico do programa.

- *Processo de Avaliação – certifica-se que:*

- *os procedimentos de entrada, de saída e de processamento lógico são, facilmente, identificados no programa.*

C2. Critério – Aderência das estruturas lógicas: a construção das estruturas, na linguagem de implementação, correspondem às estruturas lógicas do problema. (Evita-se, por exemplo, utilizar um comando de seleção de uma linguagem, para se implementar uma estrutura lógica de repetição.)

- *Processo de Avaliação – certifica-se que:*

- *as estruturas, na linguagem de implementação, correspondem às estruturas lógicas do problema.*

S3. Subfator – Rastreabilidade: característica de um programa e sua documentação permitirem a procura de uma informação através da seqüência de agregação de detalhes de um determinado aspecto, desde a sua visão mais geral até a mais detalhada, e vice-versa [ROCHA 87]. É avaliado através dos *critérios: localização interna, localização externa e organização da documentação.*

C1. Critério – Localização interna: todos os componentes do código fonte são, rapidamente, localizados através da existência de facilidades internas.

- *Processo de Avaliação – certifica-se que:*

- *existem facilidades dentro do programa fonte, que permitem localizar, com destreza, seus componentes;*

- *o programa possui referências cruzadas [BROWN 87].*

C2. Critério – Localização externa: deve ser feita pela verificação da existência de dicionário de dados, fazendo o interrelacionamentos entre os componentes do mesmo [ANDRADE 91].

- *Processo de Avaliação – certifica-se que:*

- *existe dicionário de dados ou outros mecanismos, que facilitam*

tem a localização dos relacionamentos entre os componentes do programa.

C3. Critério – Organização da documentação: há facilidades de manipulação dos programas fonte, e suas respectivas documentações, resultante da forma de como foram organizados.

- *Processo de Avaliação – certifica-se que:*
 - *há uma forma sistemática de organização dos programas fonte e de suas documentações.*

II.6.3 Objetivo Utilizabilidade

Para um software financeiro, o volume de manutenção varia, de uma maneira geral, com a dinâmica do mercado financeiro e com os ajustes na economia realizados pelo governo federal. No entanto, quando surgem estas modificações evolutivas ou adaptativas, normalmente, dispõe-se de pouco tempo para realizá-las e possuem, costumeiramente, um caráter emergencial. Daí a relevância da manutenibilidade nesta área.

Existem variadas formas de reutilização de módulos na área das finanças, como por exemplo: *cálculo de dígitos verificadores, cálculo de intervalos de datas, cálculo de imposto de renda* e outros. Vários programas fazem uso dessas informações, sendo conveniente transformá-las em módulos independentes e acessíveis. No concepção de [BIGGERSTAFF 87], o reuso de componentes do software amplifica a capacidade de desenvolvimento do mesmo, resultando em um total de símbolos menor na produção do sistema, e menor tempo gasto na organização desses símbolos.

As grandes empresas, que se relacionam intensamente com as várias entidades do mercado financeiro, devem preocupar-se com a portatibilidade de seus sistemas, pois o intercâmbio de informações entre elas, se já não o for, tende a ser computarizado.

A figura II.4 mostra os atributos selecionados para o objetivo *utilizabilidade* de programa.

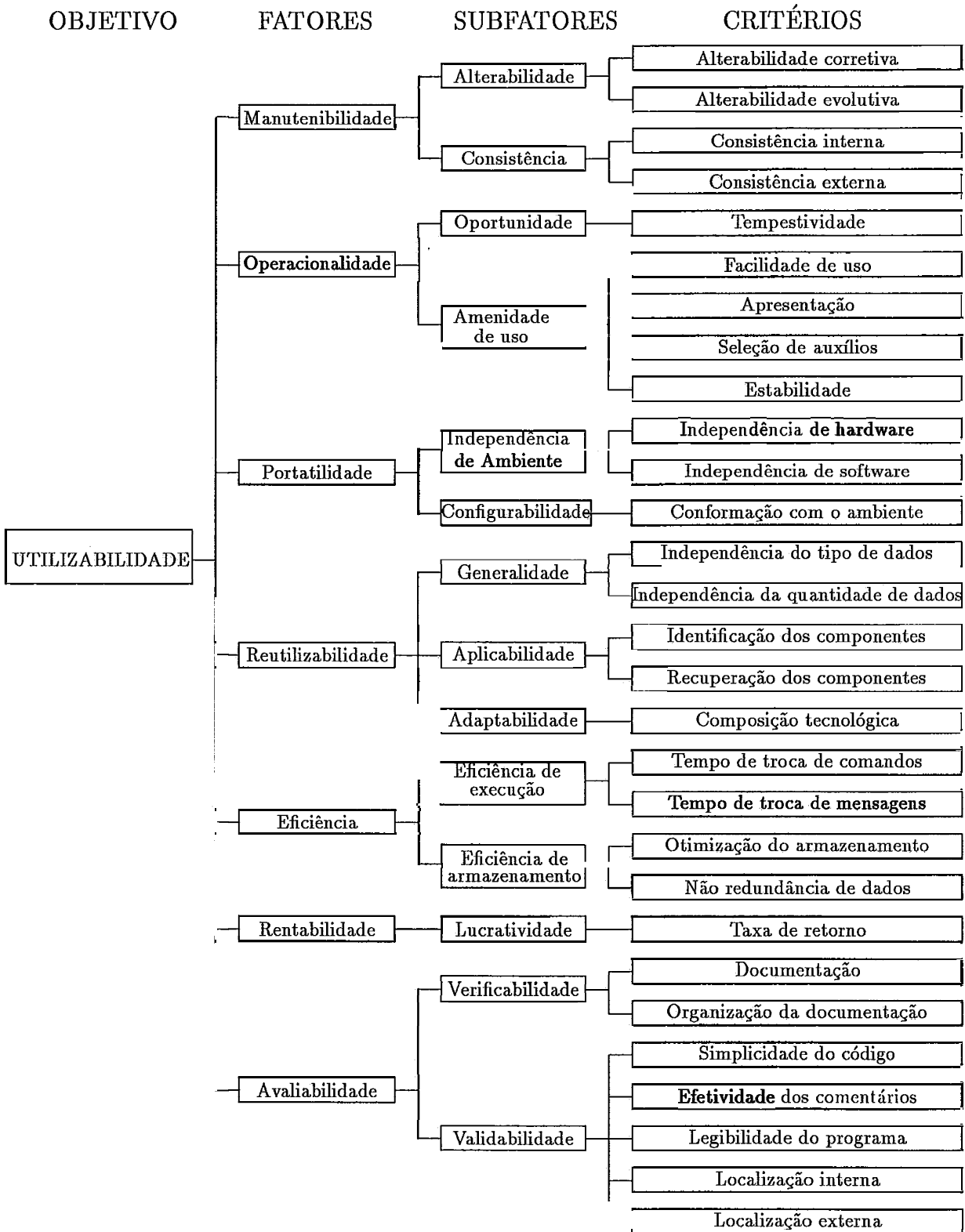


Figura II.4: Utilizabilidade de Programas

O2. Objetivo – Utilizabilidade: deve ser avaliado considerando-se todas as suas possíveis formas de utilização: uso durante a fase de operação, nas manutenções realizadas, possíveis reutilizações de módulos em outros programas, uso em equipamentos diferentes do previsto no desenvolvimento. É atingido através dos *fatores: manutenibilidade, operacionalidade, portatibilidade, reutilizabilidade, eficiência, rentabilidade e avaliabilidade* [ROCHA 90].

F1. Fator – Manutenibilidade: é a característica de um programa permitir a introdução de alterações, após ter sido inicialmente definido, desenvolvido e aceito como operacional. É atingido pelos *subfatores: alterabilidade, e consistência* [BARGUT 86].

S1. Subfator – Alterabilidade: refere-se à característica que um programa deve ter de forma a facilitar a incorporação de modificações. É avaliado através dos *critérios: alterabilidade corretiva e alterabilidade evolutiva* [CARDOSO 90].

C1. Critério – Alterabilidade corretiva: condição para que o programa seja implementado, levando em conta possíveis correções de erro ou alterações funcionais, de tal forma que sejam, facilmente, executadas.

- *Processo de Avaliação – certifica-se que:*

- *o programa encontra-se implementado de maneira a facilitar possíveis correções de erro ou alterações funcionais.*

C2. Critério – Alterabilidade evolutiva: condição para que o programa seja implementado, levando em conta possíveis melhorias e evoluções, de tal forma que, quando realizadas, sejam mencionadas na documentação e no código fonte.

- *Processo de Avaliação – certifica-se que:*

- *o programa encontra-se implementado de maneira a facilitar possíveis melhorias e evoluções.*

S2. Subfator – Consistência: refere-se à característica que um programa deve ter de forma a apresentar uma notação uniforme, para

os símbolos utilizados. É avaliado através dos *critérios: consistência interna e consistência externa* [CARDOSO 90].

C1. Critério – Consistência interna: ocorre na medida em que não haja conflitos entre os nomes de referências, de variáveis e de representações de dados, dentro do módulo [RADC-TR-85-37].

- *Processo de Avaliação – certifica-se que:*

- *não há conflito entre os nomes de referências, de variáveis e de representações de dados, internamente, no módulo.*

C2. Critério – Consistência externa: ocorre quando não há conflitos entre a forma como são implementados os módulos, os protocolos utilizados para as chamadas de função, e os procedimentos padronizados para este fim [RADC-TR-85-37].

- *Processo de Avaliação – certifica-se que:*

- *o programa não contém conflitos na maneira de implementar os módulos, os protocolos utilizados para as chamadas de função e os procedimentos padronizados.*

F2. Fator – Operacionalidade: é a característica de um programa ser oportuno e ameno ao uso, facilitando a comunicação com o usuário, durante todo o tempo em que este o utilizar. É atingido através dos *subfatores: oportunidade e amenidade de uso* [ROCHA 87].

S1. Subfator – Oportunidade: característica de um programa produzir resultados em tempo hábil. É avaliado através do *critério: tempestividade*.

C1. Critério – Tempestividade: é a característica do programa produzir resultados satisfatórios, no tempo oportuno.

- *Processo de Avaliação – certifica-se que:*

- *os resultados do programa atendem ao usuário, no tempo oportuno.*

S2. Subfator – Amenidade de uso: refere-se à característica de um programa apresentar os resultados esperados, oferecendo sentimento de satisfação e de confiança ao usuário, que o utiliza. É avaliado

através dos *critérios: facilidade de uso, apresentação, seleção de auxílios e estabilidade* [ROCHA 90].

C1. Critério – Facilidade de uso: a implementação do programa é realizada levando-se em conta requisitos, que facilitem sua manipulação [ROCHA 90].

- *Processo de Avaliação – certifica-se que:*
 - *o programa possui requisitos, que facilitem sua manipulação pelo usuário.*

C2. Critério – Apresentação: uso de interfaces gráficas e de outras características, que facilitem a apresentação de suas informações [CARDOSO 90].

- *Processo de Avaliação – certifica-se que:*
 - *o programa tem interfaces gráficas ou outras características, que facilitem a apresentação das informações.*

C3. Critério – Seleção de auxílios: disponibilidade, em tempo hábil, de características, da apresentação de procedimentos e de informações, que possam servir de auxílio para o usuário.

- *Processo de Avaliação – certifica-se que:*
 - *o programa possui meios de apresentar seus procedimentos e suas características, no instante em que o usuário desejar.*

C4. Critério – Estabilidade: o programa deve ser confortável e permanecer compreensível e familiar ao invés de sofrer alterações aleatoriamente [APPLE 87].

- *Processo de Avaliação – certifica-se que:*
 - *o programa é confortável, compreensível e familiar, durante todo o tempo de execução.*

F3. Fator – Portatibilidade: é a característica de um programa poder ser operado de maneira fácil e adequada em configurações de equipamentos diferentes da original. É atingido através dos *subfatores: independência de ambientes e configurabilidade* [ROCHA 87].

S1. Subfator – Independência de ambiente: refere-se à característica de um programa não conter restrição, quanto a aspectos de software e de hardware. É avaliado através dos *critérios: independência de hardware e independência de software.*

C1. Critério – Independência de hardware: possibilidade de execução do programa em diferentes ambientes de hardware.

- *Processo de Avaliação – certifica-se que:*
 - *o programa pode ser executado em diferentes ambientes de hardware.*

C2. Critério – Independência de software: possibilidade da execução do programa em diferentes ambientes de software.

- *Processo de Avaliação – certifica-se que:*
 - *o programa pode ser executado em diferentes ambientes de software.*

S2. Subfator – Configurabilidade: refere-se à característica que um programa deve ter, de forma a facilitar a alteração de suas características dentro de um mesmo ambiente. É avaliado através dos *critérios: conformação com o ambiente [CARDOSO 90].*

C1. Critério – Conformação com o ambiente: o programa faz uso de facilidades do ambiente, para agilizar suas possíveis alterações sem, no entanto, comprometer sua independência.

- *Processo de Avaliação – certifica-se que:*
 - *o programa faz uso de facilidades do ambiente onde executa, não comprometendo sua independência.*

F4. Fator – Reutilizabilidade: é a característica de um programa ter suas funções desenvolvidas, de maneira que o esforço requerido, para a sua reutilização parcial ou total em outras aplicações, é menor que para criar novo código. É atingido através dos *subfatores: generalidade, aplicabilidade e adaptabilidade [ROCHA 87].*

S1. Subfator – Generalidade: refere-se à característica que um programa deve ter, para que seus módulos tenham o mínimo de restrições

de tipos e de quantidade de dados. É avaliado através dos *critérios: independência do tipo de dados e independência da quantidade de dados* [CARDOSO 90].

C1. Critério – Independência do tipo de dados: aptidão de um programa operar com os vários tipos de dados da linguagem utilizada.

- *Processo de Avaliação – certifica-se que:*

- *o programa pode operar com os diversos tipos de dados da linguagem utilizada.*

C2. Critério – Independência da quantidade de dados: característica de um programa não possuir restrições, para a utilização de qualquer volume de dados.

- *Processo de Avaliação – certifica-se que:*

- *o programa não possui limitações, para a utilização de qualquer volume de dados.*

S2. Subfator – Aplicabilidade: é a característica de um programa poder ser reutilizado ao se especificar outra aplicação, na mesma área do domínio do problema. É avaliado através dos *critérios: identificação dos componentes e recuperação dos componentes*.

C1. Critério – Identificação dos componentes: consiste na padronização dos nomes dados aos componentes de software existentes, com o objetivo de facilitar a sua reutilização.

- *Processo de Avaliação – certifica-se que:*

- *as estruturas de código têm nomes padronizados, visando a sua reutilização.*

C2. Critério – Recuperação dos componentes: é a facilidade de se recuperar componentes de software existentes [JONES 88].

- *Processo de Avaliação – certifica-se que:*

- *o código é armazenado de acordo com um esquema de classificação padronizado;*

- *há facilidades na recuperação de componentes do software.*

S3. Subfator – Adaptabilidade: é a habilidade em se trocar, facilmente, componentes de código, envolvendo-os com as modificações requisitadas pelo meio. É avaliado através do *critério: composição tecnológica*.

C1. Critério – Composição tecnológica: a composição dos componentes de código deve ser de tal maneira que, em geral, não necessitem ser modificados, para serem reusados.

- *Processo de Avaliação – certifica-se que:*
 - *o código fonte pode ser facilmente reusado, sem alterações ou modificações significativas.*

F5. Fator – Eficiência: é a característica de um programa realizar suas funções sem desperdício de recursos. É avaliado através dos *subfatores: eficiência de execução e eficiência de armazenamento* [RADC-TR-85-37].

S1. Subfator – Eficiência de execução: refere-se à característica de um programa executar suas funções dentro do menor tempo possível. É avaliado através dos *critérios: tempo de troca de comandos e tempo de troca de mensagens* [CARDOSO 90].

C1. Critério – Tempo de troca de comandos: condição em que é baixo o tempo gasto para a troca de comandos no programa.

- *Processo de Avaliação – certifica-se que:*
 - *o programa executa a troca de comandos com rapidez.*

C2. Critério – Tempo de troca de mensagens: condição em que é baixo o tempo gasto na troca de mensagens (comunicação), entre os módulos de um programa.

- *Processo de Avaliação – certifica-se que:*
 - *o tempo gasto na troca de mensagens é baixo.*

S2. Subfator – Eficiência de armazenamento: refere-se à característica de um programa utilizar, adequadamente, a memória disponível. É avaliado através dos *critérios: otimização do armazenamento e não redundância de dados* [CARDOSO 90].

C1. Critério – Otimização do armazenamento: utilização das características de otimização de armazenamento, disponíveis na linguagem de programação.

- *Processo de Avaliação – certifica-se que:*
 - *o programa utiliza características de otimização de armazenamento, disponíveis na linguagem de programação.*

C2. Critério – Não redundância de dados: inexistência de duplicidade de dados.

- *Processo de Avaliação – certifica-se que:*
 - *os dados não são armazenados em duplicidade.*

F6. Fator – Rentabilidade: é a característica de um programa possuir relação custo/benefício aceitável. É avaliado através do *subfator: lucratividade* [ROCHA 87].

S1. Subfator – Lucratividade: refere-se à característica do programa em relação ao retorno financeiro, às vantagens e aos benefícios advindos de seu uso ou de sua comercialização. É avaliado através do *critério: taxa de retorno.*

C1. Critério – Taxa de retorno: situação em que a taxa de retorno da utilização do programa é superior ao investimento feito em sua confecção, ou em sua aquisição.

- *Processo de Avaliação – certifica-se que:*
 - *a taxa de retorno da utilização do programa é superior ao investimento feito em sua confecção ou em sua aquisição.*

F7. Fator – Avaliabilidade: é a característica de um programa, que retrata a facilidade em verificá-lo e validá-lo de modo a assegurar a execução da função, que lhe cabe. É avaliado através dos *subfatores: verificabilidade e validabilidade* [ROCHA 87].

S1. Subfator – Verificabilidade: é a característica de um programa poder ser avaliado facilmente, com relação à forma de sua apresentação. É avaliado através dos *critérios: documentação e organização da documentação.*

C1. Critério – Documentação: a documentação do programa deve estar disponível e com bom padrão de qualidade (semelhante ao mesmo critério do subfator modularidade).

- *Processo de Avaliação – certifica-se que:*

- *o programa possui documentação disponível e de boa qualidade.*

C2. Critério – Organização da documentação: facilidade na manipulação dos fontes dos programas, com respectivas documentações, resultantes da forma com que foram organizados (semelhante ao mesmo critério do subfator rastreabilidade).

- *Processo de Avaliação – certifica-se que:*

- *existe uma forma sistemática de organização dos códigos fonte e de suas documentações, agilizando o acesso aos mesmos.*

S2. Subfator – Validabilidade: é a característica de um programa poder ser, facilmente, avaliado dentro de um processo finito, e economicamente viável. É avaliado através dos *critérios: simplicidade do código, efetividade dos comentários, legibilidade do programa, localização interna e localização externa* [RADC-TR-85-37].

C1. Critério – Simplicidade do código: consiste em uma determinada função contem uma quantidade mínima necessária de comandos e variáveis (semelhante ao mesmo critério do subfator concisão).

- *Processo de Avaliação – certifica-se que:*

- *o programa não possui trechos, que nunca serão executados, gerando dificuldades no entendimento do código.*

C2. Critério – Efetividade dos comentários: condição em que os comentários usados, explicam, de maneira clara e objetiva, o item, que está sendo apurado.

- *Processo de Avaliação – certifica-se que:*

- *existem comentários simples, claros e concisos, efetivamente,*

auto-documentando o programa.

C3. Critério – Legibilidade do programa: não utilização de identificadores, que dificultem o entendimento do programa.

- *Processo de Avaliação – certifica-se que:*

- *o programa não possui identificadores, que dificultem seu entendimento.*

C4. Critério – Localização interna: todos os componentes do programa fonte são, rapidamente, localizados através da existência de facilidades internas (semelhante ao mesmo critério do subfator rastreabilidade).

- *Processo de Avaliação – certifica-se que:*

- *existem facilidades dentro do programa fonte, que permitem localizar, com destreza, seus componentes;*

- *o programa possui referências cruzadas [BROWN 87].*

C5. Critério – Localização externa: deve ser feita pela verificação da existência de dicionário de dados fazendo os interrelacionamentos entre os componentes do mesmo (semelhante ao mesmo critério do subfator rastreabilidade) [ANDRADE 91].

- *Processo de Avaliação – certifica-se que:*

- *existe dicionário de dados ou outros mecanismos, que facilitam a localização dos relacionamentos entre os componentes do programa.*

II.6.4 Objetivo Confiabilidade Conceitual

Este objetivo é de grande monta para o software financeiro, pois decisões importantes da empresa são tomadas, via de regra, mediante os resultados auferidos do processamento de seus dados.

Na área financeira, não se pode correr o risco de gerar programas, que não representem, realmente, os objetivos do usuário. Pode estar envolvido aí, uma grande transação financeira ou ser subsídio para uma decisão importante, que

mudará até os rumos da própria instituição.

Portanto, é indispensável que o programa represente, fidedignamente, o que foi especificado e projetado. Isto pode representar a própria saúde financeira da empresa e, por seguinte, sua própria competitividade no mercado, ou até sua própria existência.

De uma maneira geral, os sistemas financeiros envolvem somas pecuniárias vultosas (às vezes, a grande maioria do capital dos clientes). É visivelmente notório, que a integridade e a segurança das informações manipuladas assumem uma importância vital.

A figura II.5 mostra os atributos selecionados para o objetivo *confiabilidade conceitual* de programa.

O3. Objetivo – Confiabilidade conceitual: um programa atinge este objetivo, quando implementa, satisfatoriamente, o que foi especificado e projetado. Portanto, é fundamental que esse objetivo seja plenamente atingido, para que o programa satisfaça às necessidades e aos requisitos, que motivaram sua construção. É atingido através dos *fatores: fidedignidade e integridade* [ROCHA 87].

F1. Fator – Fidedignidade: é a característica de um programa corresponder ao que foi especificado e projetado. É atingido pelos pelos *subfatores: precisão, completude e necessidade* [ROCHA 87].

S1. Subfator – Precisão: é a característica de um programa proporcionar a suficiente exatidão nos cálculos e nos resultados de modo a satisfazer a utilização pretendida pelo usuário, descrita nos requisitos de desempenho. É avaliado através dos *critérios: acurácia e correção* [ROCHA 87].

C1. Critério – Acurácia: os métodos numéricos são consistentes com os resultados da aplicação.

- *Processo de Avaliação – certifica-se que:*

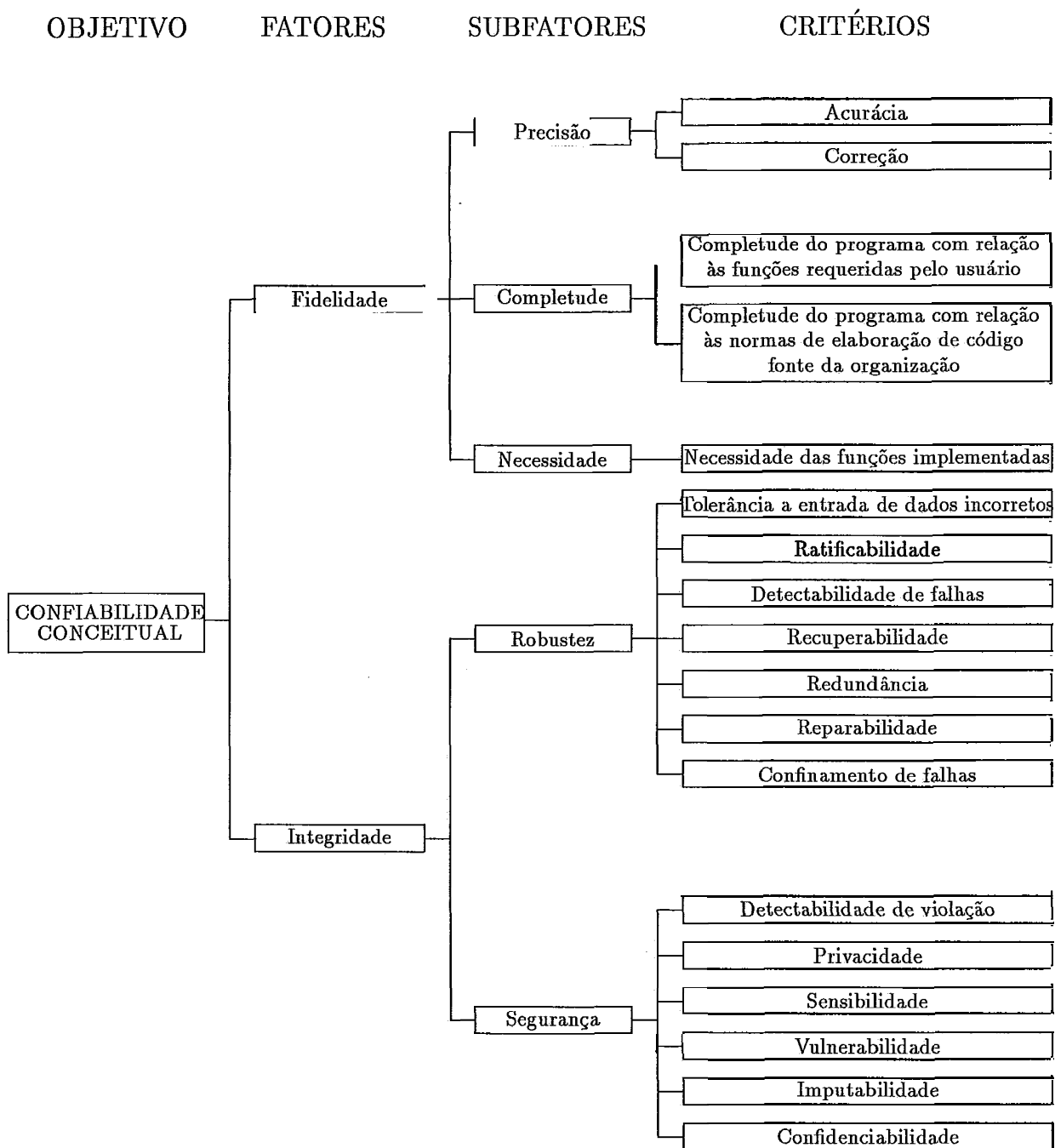


Figura II.5: Confiabilidade Conceitual de Programas

– *os resultados da aplicação estão consistentes com os métodos numéricos empregados.*

C2. Critério – Correção: os resultados obtidos são coerentes com o que é esperado pelo usuário.

- *Processo de Avaliação – certifica-se que:*

– *os resultados obtidos do programa correspondem ao esperado pelo usuário.*

S2. Subfator – Completude: é a característica de um programa ter implementadas todas as funções especificadas. É avaliado através dos *critérios: completude do programa com relação às funções requeridas pelo usuário, e completude do programa com relação às normas de elaboração de códigos fonte da organização [RADC-TR-85-37].*

C1. Critério – Completude do programa com relação às funções requeridas pelo usuário: condição em que todas as funções pretendidas pelo usuário estão implementadas no software.

- *Processo de Avaliação – certifica-se que:*

– *todas as funções pretendidas pelo usuário estão implementadas, através de processos de checagens no projeto ou na especificação do sistema.*

C2. Critério – Completude do programa com relação às normas de elaboração de códigos fonte da organização: o programa deve estar implementado, tendo por base as normas de elaboração de códigos fonte da organização.

- *Processo de Avaliação – certifica-se que:*

– *o programa foi codificado de acordo com as normas de elaboração de códigos fonte da organização.*

S3. Subfator – Necessidade: é a característica de um programa ter implementadas somente as funções, que foram especificadas. É avaliado através do *critério: necessidade das funções implementadas [ROCHA 87].*

C1. Critério – Necessidade das funções implementadas: situação na qual todas as funções imprescindíveis ao alcance dos objetivos do projeto estejam implementadas.

- *Processo de Avaliação – certifica-se que:*

- *as funções imprescindíveis ao alcance dos objetivos do projeto estão implementadas.*

F2. Fator – Integridade: é a característica de um programa enfrentar situações hostis — *dados errados ou agressões*. É atingido através dos *subfatores: robustez e segurança* [ROCHA 87].

S1. Subfator – Robustez: refere-se à característica do programa possuir meios apropriados, para enfrentar situações hostis, reagindo a elas, sem perda do controle. É avaliado através dos *critérios: tolerância a entrada de dados incorretos, ratificabilidade, detectabilidade de falhas, recuperabilidade, redundância, reparabilidade e confinamento de falhas* [ROCHA 87].

C1. Critério – Tolerância a entrada de dados incorretos: situação em que o programa é capaz de suportar um certo grau de variação na entrada de dados, sem mau funcionamento ou rejeição [GILB 77].

- *Processo de Avaliação – certifica-se que:*

- *o programa é capaz de tolerar um certo grau de variação na entrada de dados, sem mau funcionamento ou rejeição.*

C2. Critério – Ratificabilidade: o programa admite, pelo operador, a realimentação dos dados de entrada, permitindo que esses dados possam ser validados posteriormente [SEI 86].

- *Processo de Avaliação – certifica-se que:*

- *é permitido a realimentação no programa, pelo operador dos dados de entrada, para que possam ser validados posteriormente.*

C3. Critério – Detectabilidade de falhas: o programa detecta desvios de comportamentos esperado para o software, hardware

e operador [GILB 77].

- *Processo de Avaliação – certifica-se que:*

- *o programa é capaz de detectar desvios de comportamento esperado para o software, hardware e operador.*

C4. Critério – Recuperabilidade: existem mecanismos, que possibilitem a correção de erros pelo próprio programa [GILB 77].

- *Processo de Avaliação – certifica-se que:*

- *o programa possui mecanismos próprios de correção de erros;*
- *o programa é capaz de reconstruir, na forma e no conteúdo desejados, suas informações.*

C5. Critério – Redundância: existem cópias dos módulos mais críticos para serem utilizadas, em casos de falhas dos módulos em operação [MEREY 85].

- *Processo de Avaliação – certifica-se que:*

- *existem cópias dos módulos considerados mais críticos.*

C6. Critério – Reparabilidade: existem mecanismos, que permitam ao software ser reparado, e retorne às condições de operação, em tempo hábil [GILB 77].

- *Processo de Avaliação – certifica-se que:*

- *há mecanismos de reparação do software em tempo hábil.*

C7. Critério – Confinamento de falhas: condição em que os erros, que, porventura, ocorram em uma determinada parte do programa, não migrem para outras partes do mesmo programa [MYERS 76].

- *Processo de Avaliação – certifica-se que:*

- *os erros, que venham a ocorrer em uma determinada parte do programa, não migram para outras regiões do mesmo programa.*

S2. Subfator – Segurança: é a característica de um programa possuir a habilidade de evitar falhas, que possam provocar danos materiais ou humanos. É avaliado através dos *critérios: detectabilidade de*

violação, privacidade, sensibilidade, vulnerabilidade, imputabilidade e confiabilidade [ROCHA 87].

C1. Critério – Detectabilidade de violação: há mecanismos, que evitem que o dano objetivado pelo acesso indevido, venha a ocorrer [GILB 77].

- *Processo de Avaliação – certifica-se que:*

- *o programa possui mecanismos, que evitem danos causados por acessos indevidos.*

C2. Critério – Privacidade: há mecanismos de proteção e de controle das operações realizadas por um determinado indivíduo [SEI 86].

- *Processo de Avaliação – certifica-se que:*

- *há mecanismos de proteção e de controle das operações, realizadas por um determinado indivíduo.*

C3. Critério – Sensibilidade: há indicações da importância das informações processadas [SEI 86].

- *Processo de Avaliação – certifica-se que:*

- *existem indicações da importância dos dados processados.*

C4. Critério – Vulnerabilidade: há mecanismos, que tornem o programa seguro a determinados tipos de ataque [SEI 86].

- *Processo de Avaliação – certifica-se que:*

- *existem mecanismos, que tornem o programa seguro a determinados tipos de ataque.*

C5. Critério – Imputabilidade: há mecanismos, que responsabilizem uma entidade ou um indivíduo, por suas ações no manuseio do software [SEI 86].

- *Processo de Avaliação – certifica-se que:*

- *existem mecanismos, que responsabilizem uma entidade ou um indivíduo, por suas ações no manuseio do software.*

C6. Critério – Confiabilidade: há mecanismos, que garantam que a informação será protegida contra revelações ou co-

locada à disposição de indivíduos ou de órgãos não autorizados [SEI 86].

- *Processo de Avaliação – certifica-se que:*

- há mecanismos, que garantam a proteção das informações contra acessos não autorizados.*

II.7 Conclusão

Neste capítulo, foi dada uma visão geral sobre qualidade de software. Baseando-se em um método para a avaliação da qualidade de software, apresentou-se atributos de qualidade de programas, encontrados na literatura técnica atual, que se adequassem aos sistemas financeiros. No capítulo seguinte, comenta-se, brevemente, sobre as características das aplicações financeiras, com o objetivo de conhecê-las melhor e a seus usuários.

Capítulo III

APLICAÇÕES FINANCEIRAS

Faz-se, aqui, algumas considerações básicas sobre aplicações financeiras. Todos nós, de uma maneira ou de outra, somos diretamente afetados pelas oscilações do mercado financeiro, principalmente, as empresas na administração de suas finanças. As aplicações na área financeira, notadamente, quando inseridas em economias amplamente indexadas, como a brasileira, necessitam de um instrumento confiável de apoio ao processo empresarial de tomada de decisão: o *software financeiro*.

Neste contexto, é que se justifica o estudo da avaliação da qualidade de sistemas financeiros, como um meio de auxiliar os projetistas destes, no controle da qualidade do produto gerado. Os usuários, por sua vez, conhecendo o perfil de qualidade destes sistemas (objetivo do trabalho), podem cooperar com os projetistas, para que as questões, no desenvolvimento do produto de software, sejam resolvidas em uma ordem de prioridades, dentro de um planejamento previsível e viável.

III.1 Aplicações Administrativas

A administração moderna é uma integração de várias escolas clássicas, com seus objetivos direcionados para quatro funções: *o planejamento, a organização, a direção e o controle*. No *planejamento*, procura-se estabelecer, antecipadamente, um esquema de previsão relativo a ações futuras. A *organização* trata da maneira de se agrupar, logicamente, diversas tarefas. A *direção* coordena atividades, e o *controle* verifica a execução do plano, para corrigí-lo e realimentá-lo [FERREIRA 87].

Há um consenso sobre as atividades administrativas, resumindo-as em quatro áreas: *a administração de material, de pessoal, de produtos de vendas e de finanças*. Na administração financeira, há cinco atividades importantes: [FLINK 77]

- *análise financeira dos registros e demonstrativos contábeis;*
- *estimativas de fluxos de caixa;*
- *análise e escolha de investimentos;*
- *fornecimento de informações financeiras atualizadas, para apoiarem decisões;*
- *elaboração de planos financeiros;*

III.2 Administração Financeira

As informações sobre administração financeira, que se seguem, foram pesquisadas em [HORNE 79], [CHERRY 82], [GILTMAN 87], e [MARTINS 90].

As finanças de empresas constituem uma das áreas especializadas, dentro do campo das finanças, que é um dos campos funcionais da administração de empresas. As finanças podem ser definidas, simplesmente, como a administração do dinheiro e das reivindicações monetárias.

A análise das demonstrações contábeis constitui-se em um dos mais importantes estudos da administração financeira. O planejamento financeiro de uma empresa é desenvolvido, fundamentalmente, através da projeção de suas demonstrações contábeis, como estimativa mais aproximada possível da posição econômico-financeira esperada.

A Administração Financeira é uma forma de Economia aplicada. A Microeconomia oferece vários guias de maximização do lucro, baseados na teoria de firma. O administrador financeiro baseia-se nessas teorias econômicas, para que as empresas operem eficiente e lucrativamente.

III.2.1 O Significado das Finanças nas Empresas

Há várias especializações na área das finanças, naturalmente, cada uma com seus problemas e objetivos: *finanças públicas*, *finanças internacionais* e *finanças das instituições*.

A área das finanças das instituições será aquela sobre a qual incide nosso interesse. As finanças das instituições interessam-se pela aquisição e uso eficiente dos fundos exigidos pela empresa. A função das finanças é verificar se existem fundos disponíveis, para serem empregados em projetos, que ofereçam lucro. A segunda grande tarefa da administração financeira é cuidar, para que os fundos estejam distribuídos entre os vários usos concorrentes, na combinação mais eficiente possível.

Estreitamente associado com a aquisição de fundos e seu uso eficiente está o planejamento da função financeira. É essencial, um planejamento cuidadoso, em qualquer atividade administrativa, e a importância das decisões financeiras, para o sucesso geral da empresa, faz do planejamento uma fase decisiva no processo de aquisição e uso eficaz de fundos.

III.2.2 O Papel da Administração Financeira

Toda decisão financeira deve ser avaliada em termos de algum objetivo, meta, ou critério. Uma decisão financeira seria boa, quando contribuísse para se atingir esse objetivo. São citados alguns objetivos da tomada de decisão financeira, sugeridos por estudiosos e líderes empresariais:

- *maximização do lucro*: a empresa, dados vários cursos de ação, escolheria a possibilidade, que oferecesse a expectativa de maior lucro;
- *lucros satisfatórios*: a empresa deve realizar um retorno sobre suas operações, que a torne capaz de compensar seus acionistas, razoavelmente, pelo capital fornecido, não procurando, necessariamente, o lucro máximo;
- *equilíbrio de interesses*: a empresa opera, de maneira tal, a alcançar um equilíbrio de todos os elementos em sociedade;

- *interesses administrativos*: decisões-chaves são muitas vezes tomadas, para beneficiarem os administradores profissionais.

O objetivo mais amplamente aceito, para a tomada de decisão, é a maximização do valor atual do patrimônio dos proprietários na empresa, isto é, a maximização do valor de mercado de uma participação da empresa. Este valor é uma função de duas variáveis: *o nível de lucros previstos e o risco ou incerteza associados à realização real desse nível de lucros.*

III.2.3 Funções do Gerente Financeiro

O lugar do gerente financeiro, na estrutura organizacional da empresa, fica, normalmente, no nível da superior administração. A razão disto é que as decisões financeiras afetam toda a empresa. Por isto é que pode ser solicitado a opinar sobre problemas como *fusão, consolidação ou falência.*

O gerente financeiro, ao adquirir e distribuir os fundos de que a empresa necessita, precisa usar e confiar nos dados financeiros fornecidos. Sua função é administrar o fluxo de fundos da empresa, de maneira a contribuir para a conquista de máximos valores justificáveis para os proprietários da empresa. O software financeiro, neste contexto, é um alinhado importante.

O papel exercido pelo gerente financeiro, em uma empresa moderna, está sujeito a um processo de permanente transformação. Suas responsabilidades têm-se ampliado e, paralelamente, adquirido uma grande importância no desenvolvimento da empresa. Atualmente, podemos citar como responsabilidades de um gerente financeiro:

- *determinar, na empresa, o volume total de capital a ser empregado;*
- *alocar esse capital, eficientemente, nos vários ativos;*
- *obter a estrutura financeira ótima, em relação à avaliação geral da empresa.*

O gerente financeiro consome a maior parte de seu tempo com a

administração do capital de giro, planejamento, distribuição e controle. A administração do capital a longo prazo também é importante, devido aos grandes montantes de fundos envolvidos e as conseqüências de erros sobre projetos, que comprometerão a empresa durante muitos anos. Também é atribuída ao gerente financeiro uma grande quantidade de funções rotineiras, como: *política de crédito, relações bancárias, custódias de caixa e de títulos, administração de riscos, transações imobiliárias e relações financeiras.*

III.2.4 Capital de Curto Prazo na Estrutura Financeira

Antes que o gerente financeiro possa começar a negociar fundos para satisfazerem a demanda a curto prazo, precisa saber *de quanto precisará, quando os fundos serão necessários, e quando e como os fundos podem ser devolvidos.* Por isso, ele se interessa, principalmente, pelo orçamento financeiro, já que é este, que lhe dará essas informações básicas necessárias. Em outras palavras, é preciso um planejamento financeiro cuidadoso.

O gerente financeiro é o responsável pela previsão das necessidades de fundos de capital de giro (*ativos correntes*) e pela execução de planos, para satisfazerem a essas necessidades. Os principais componentes do capital de giro são:

- *caixa*, exceto o restrito para outras operações, que não sejam correntes;
- *títulos negociáveis*;
- *estoques*;
- *contas a receber*;
- *despesas antecipadas*, quando não constituírem ativos correntes, criando fundos dentro do ciclo operacional corrente.

Ativo disponível é o que a empresa pode reunir, sem grande demora, para satisfazer a demanda de *caixa*. Esta categoria de aplicações de fundos inclui a *caixa* — moeda corrente e saldos de conta bancária de movimentos — e os *investimentos temporários*. A empresa compromete fundos com o ativo monetário para

comprar liquidez ou seja, a capacidade de satisfazer a demanda de *caixa*. Com a finalidade de reduzir a incerteza no orçamento de caixa, pode-se fazer uso da análise de sensibilidade ou de simulação, fornecida por sistemas financeiros.

O *estoque* são as mercadorias mantidas, fisicamente, disponíveis pela empresa, na expectativa de ingressarem no ciclo de produção, de seguir o seu curso produtivo normal, ou de serem comercializadas. O interesse do gerente financeiro no estoque está em dois níveis:

- *como funcionário da alta administração, ele participa do estabelecimento da política básica de estoque;*
- *sua responsabilidade pelo fornecimento de fundos significa que ele precisa estar ciente dos acontecimentos, que afetarão as necessidades de estoque da firma, durante o próximo período.*

Contas a receber são os montantes devidos à empresa, provenientes da venda de mercadorias ou serviços no curso ordinário dos negócios. O propósito do compromisso de fundos com os valores a receber é conseguir um lucro líquido, aumentando o fluxo de receita operacional. O gerente financeiro também é responsável por este planejamento.

III.2.5 Capital de Longo Prazo na Estrutura Financeira

Despesa de capital é aquela cujos benefícios virão durante um período superior a um ano. Essas despesas são semelhantes às do capital de giro, embora sejam mais refinadas.

A avaliação dos projetos de capital precisa incluir a consideração da *incerteza* ou *risco* associados com os projetos determinados. O *risco* refere-se a uma situação, em que o resultado específico de uma tentativa é desconhecido, todavia a distribuição probabilística aplicável aos resultados é conhecida. Na *incerteza*, não conhecemos nem a magnitude, nem a distribuição probabilística dos resultados possíveis.

A principal questão da administração financeira, relacionada com os fundos das operações, é a disposição de lucros: a decisão quanto ao montante, que será pago em dividendos, e o montante, que será retido dentro da empresa.

III.2.6 Aplicações Financeiras

As operações do Sistema Financeiro Nacional podem ser feitas através dos seguintes segmentos:

- *mercado de crédito;*
- *mercado monetário;*
- *mercado cambial;*
- *mercado de capitais.*

As operações financeiras disponíveis para suprir as necessidades circulantes, ou atender aos excessos momentâneos de caixa das empresas, são desenvolvidas, basicamente, no mercado de crédito e mercado monetário.

O mercado de crédito tem como objetivo suprir a demanda por recursos de curto e de médio prazo da economia, sendo constituído por todas as instituições financeiras bancárias ou monetárias (bancos comerciais).

No mercado monetário, onde se realizam operações que envolvem títulos e valores mobiliários de renda fixa, as empresas dispõem de condições para aplicar seus eventuais excessos de liquidez com vista a auferir determinada remuneração financeira.

O mercado cambial envolve negociações de moedas estrangeiras conversíveis, entre instituições financeiras bancárias e não bancárias autorizadas.

O mercado de capitais engloba as operações financeiras de médio e de longo prazos.

III.2.7 Análise Financeira

A análise financeira é o exame e comparação dos dados financeiros, para tirar conclusões sobre a posição corrente e em perspectiva da empresa. Ao executar a análise dos dados financeiros, o analista deve usar toda a sua compreensão, experiência e habilidade sobre questões básicas da administração financeira.

Para que o administrador financeiro possa tomar decisões racionais em relação a medidas condizentes com as metas da empresa, deverá ter, à sua disposição, determinados instrumentos analíticos. O instrumento mais amplamente usado na análise financeira é a análise de índices, que envolve o exame das relações estratégicas entre os itens sobre as demonstrações da empresa.

Os índices financeiros mais utilizados dividem-se em quatro grupos: *medidas de liquidez, de atividades, de endividamento e de lucratividade*. A utilização desses índices permite a avaliação de certos aspectos do desempenho de uma empresa, sejam os acionistas, os mutuantes reais e potenciais, bem como a própria administração da empresa.

A análise e a interpretação dos vários índices financeiros, respaldadas pelas informações gerenciais geradas por sistemas financeiros, proporcionarão ao analista capaz e experiente, uma visão mais objetiva e real da situação financeira e do desempenho da empresa.

III.3 O Software Financeiro Atual

O software financeiro atual é uma ferramenta indispensável na área de finanças de uma empresa, melhorando a qualidade e reduzindo o tempo de tomada de decisões de seus usuários. Além disso, estes produtos de software têm exercido influências diretamente no grau de competitividade das empresas, adequando-as às preferências dos clientes ou amoldando-as à dinâmica do mercado.

Já existem alguns sistemas da área financeira disponíveis e comercializados no mercado. Algumas empresas optam por adquirir tais produtos, analisando

o custo/benefício dos mesmos. Isto se reflete, mais diretamente, no tempo que seria gasto no desenvolvimento de um outro produto similar, no grau de complexidade do mesmo, na disponibilidade da futura equipe desenvolvedora do produto, no custo de aquisição, no processo de implantação e na manutenção do sistema adquirido, entre outros fatores.

Todavia, o controle de qualidade desses produtos de software ainda se processa com o atrelamento de uma equipe cativa de analistas aos mesmos. Os membros da equipe de manutenção, costumeiramente, conhecem profundamente o código fonte. O sistema está na memória do analista ou, usualmente, o analista é parte integrante da memória do sistema.

III.4 Conclusão

Neste capítulo, apresentou-se algumas características de aplicações da área de finanças e de seus usuários, no contexto da administração financeira de uma empresa. No capítulo seguinte, desenvolve-se uma metodologia de pesquisa, para avaliar a utilizabilidade dos atributos de qualidade de programas, apresentados no capítulo anterior, para software financeiro.

Capítulo IV

METODOLOGIA DE PESQUISA

A metodologia diz respeito à resolução de problemas em todas as suas fases, não sendo limitada, somente, ao levantamento de dados ou ao uso de técnicas particulares. Uma investigação bem conduzida deve satisfazer às exigências tanto teóricas como observacionais.

Apresenta-se uma metodologia de trabalho, para avaliar a aplicabilidade dos atributos de qualidade de programa, descritos anteriormente, utilizando-se de instrumentos metodológicos indispensáveis ao alcance de resultados confiáveis. Serviu-se de uma metodologia na orientação do processo de investigação, na seleção das hipóteses, na tomada de decisão e nas técnicas de coletas de dados (por exemplo, a obtenção de pesos para os atributos de qualidade). Os dispositivos de investigação foram relacionados com a realidade observada. Esta metodologia foi baseada em [COURA 86].

IV.1 Pesquisa de Campo

O objetivo de uma pesquisa de campo é realizar um estudo explanatório, com funções básicas de gerar dados para uma técnica de classificação dos aspectos propostos em ordem de prioridade, e para uma técnica de avaliação de sistemas interativos [ESTEVAM 90].

A pesquisa de campo realizada percorreu as seguintes etapas na consecução de seus objetivos:

- *definição das hipóteses e dos objetivos da pesquisa;*
- *definição dos elementos da população;*
- *elaboração de questionários;*
- *definição de técnicas de aplicação do questionário;*
- *seleção das Empresas a serem pesquisadas;*
- *aplicação de questionários;*
- *tratamento estatístico dos dados coletados.*

IV.1.1 Hipóteses e Objetivos da Pesquisa de Campo

As seguintes hipóteses foram levantadas, para serem comprovadas através da pesquisa de campo:

1. *Os atributos de qualidade de um software financeiro pertencem, em sua maioria, a intervalos de pesos elevados;*
2. *Os usuários são mais exigentes que os analistas de sistema, em relação aos requisitos de qualidade de um software financeiro;*
3. *Sistemas de um mesmo tipo de aplicação financeira têm características de qualidade semelhantes, tendendo a um perfil de qualidade mais definido, que o perfil de qualidade das aplicações financeiras em geral.*

Partindo-se das premissas anteriores, persegue-se os seguintes objetivos como resultados da pesquisa:

- *verificar se os atributos de qualidade de programa selecionados, adequam-se aos sistemas financeiros;*
- *traçar um perfil de qualidade para software da área financeira, para a avaliação dos mesmos.*

IV.1.2 Definição dos Elementos da População

As características das aplicações financeiras computadorizadas determinaram a escolha dos elementos da população, abaixo especificados, sobre o qual a pesquisa foi efetuada:

- *analista de sistema*: por ser o elemento indispensável e atuante, desde a concepção e em toda a vida útil do software;
- *usuário*: por ser aquele, que de fato faz uso efetivo do sistema e para o qual o sistema foi construído, devendo serem satisfeitas, plenamente, suas necessidades.

O analista de sistema deve satisfazer às necessidades de seus usuários sem, no entanto, prometer-lhes resultados inatingíveis — o computador não possui discernimento e somente faz o que lhe é mandado fazer. É necessário desenvolver a habilidade de saber perguntar e perguntar — nunca tomar como certo e definitivo aquilo que não foi ainda totalmente esclarecido. O usuário, por sua vez, precisa aprender a dizer e a definir o que realmente quer ou necessita. Todo o esforço no esclarecimento e na elucidação desde os pontos mais críticos do sistema, até os considerados de menor relevância, principalmente em seu bojo conceitual, minimizarão e até eliminarão transtornos e aborrecimentos na fase de implantação e de operação do sistema.

Resumindo, analistas de sistema e usuários devem falar a mesma linguagem ou, no mínimo, fazerem mútuos esforços neste sentido, perseguirem, solidariamente, os mesmos objetivos. Não existem fórmulas mágicas, nem o computador é um mago, nem faz milagres. É preciso dialogar mesmo, é necessário cooperação mútua, para que se alcancem resultados satisfatórios, traduzidos na elaboração de sistemas eficientes e eficazes.

IV.1.3 Processo de Investigação

O dispositivo de investigação consistiu na elaboração de dois *questionários* distintos para *analistas de sistema* e para *usuários*, sendo que este último é um subconjunto do primeiro. O que motivou a aplicação desses questionários foi a relevante importância, tanto dos analistas de sistema, quanto dos usuários, durante todo o ciclo de vida do produto de software. O introsamento saudável entre ambos é de vital monta, irremediavelmente indispensável — um não existe sem o outro.

As informações coletadas pelos questionários comprovarão ou não as hipóteses levantadas. Os requisitos de qualidade propostos para um software financeiro perfazem um total de 100 ítems pesquisados para analistas de sistema, e 50 ítems pesquisados para usuários, sob a ótica do método escolhido para avaliação da qualidade de software, além de informações cadastrais (anexo B e D). Para os analistas de sistema, tem-se 74 critérios, que se agrupam em 26 subfatores. Para os usuários, tem-se 34 critérios, que se agrupam em 16 subfatores.

Estas diferentes quantidades de ítems, para os dois questionários, deve-se ao fato de que, no desenvolvimento e manutenção de um sistema, o analista de sistema envolve-se com um número maior de questões, tais como a representação do produto de software, de maneira inteligível pelo computador. Todos esses ítems foram dispostos em quadros, de acordo com seu objetivo de qualidade, de forma que poderiam servir, para avaliar até quatro sistemas por entrevistado.

Na obtenção dos dados, uma vez que não foi um processo imediato entre a realidade observada e o observador, foram utilizadas as seguintes técnicas de aplicação dos questionários:

- *preenchimento de informações cadastrais de acordo com os dados solicitados;*
- *obtenção pesos (0 a 5) para os requisitos de qualidade de acordo com o seu grau de importância, para o(s) sistema(s) financeiro(s) avaliados, de acordo com a seguinte escala de valores:*

– peso 5,0 – E — Extremamente Alto;

- peso 4,0 – A — Alto;
- peso 3,0 – ME – Médio;
- peso 2,0 – B — Baixo;
- peso 1,0 – MB – Muito Baixo;
- peso 0,0 – NE – Nenhum ou Ausente.

IV.1.4 Técnica de Coleta de Dados

A aplicação do questionário seguiu as seguintes etapas de coleta de dados:

- *as entrevistas foram iniciadas fazendo-se uma breve introdução dos objetivos da pesquisa, e orientações gerais sobre o preenchimento do questionário;*
- *o questionário foi acompanhado de um Manual Explicativo dos Requisitos de Qualidade de Software Financeiro (anexo C ou E), contendo uma descrição sucinta desses requisitos;*
- *foram entregues aos entrevistados seus respectivos questionários (anexo B ou D);*
- *foram selecionados sistemas importantes da área financeira, no máximo, até quatro por entrevistado (limitação do questionário), em que o mesmo estivesse trabalhando ou tivesse trabalhado neles, na empresa;*
- *obteve-se dos entrevistados um peso quantitativo, de acordo com a escala de valores, descrita acima, para cada item avaliado, através de explicações verbais ou contidas em seu respectivo Manual Explicativo;*
- *salientou-se que não se estava fazendo avaliação do sistema cadastrado no questionário, mas avaliando-se o grau de importância de cada item pesquisado, tomando-se, como base, o sistema em questão. Isto significava dizer que o peso atribuído retrataria como o sistema deveria estar, e não o estado em que o mesmo se apresentava;*

- *alertou-se, também, para que os pesos atribuídos tivessem um grau de importância relativa entre si, para que não se tendesse a dar sempre uma gradação máxima a todos os itens, na busca de um sistema perfeito e longe de uma realidade tangível.*

A metodologia permite-nos, ao nível da observação e da produção de dados, de um lado, indagar as limitações das técnicas de pesquisa empírica comumente utilizadas e, de outro lado, abrir a discussão acerca de métodos e procedimentos de natureza ativa e coletiva.

Inicialmente, foram realizados pré-testes pilotos com cinco pessoas com o mesmo perfil dos elementos da população, sendo três analistas de sistema e dois usuários. O questionário foi, então, empregado, para ser verificada sua aplicabilidade como ferramenta. Os pontos de dúvidas, em relação aos itens dos questionários ou em relação aos Manuais Explicativos, foram modificados no sentido de elucidá-los. Sugestões foram acatadas.

Devido a extensão dos questionários, optou-se pela aplicação dos mesmos sempre em forma de entrevista, onde o entrevistado ficava somente com seu questionário (anexo A ou C) e o entrevistando com o respectivo *Manual Explicativo dos Requisitos de Qualidade de Software Financeiro* (anexo B ou D). Cada item dos quadros do questionário era explanado com a ajuda do Manual Explicativo ou através de explicações verbais, de acordo com o interesse ou solicitações do entrevistado.

IV.1.5 Seleção das Empresas

As propriedades desejadas, em uma análise estatística, podem depender, diretamente, do método empregado. O processo de seleção de uma amostra é complexo, para que se possa tirar conclusões válidas sobre a mesma.

As empresas selecionadas pertencem ao grupo das maiores instituições do setor financeiro, agrupadas nas ramificações abaixo, que constam na edição da revista *Exame Melhores e Maiores* de 1991 [EXAME 91]:

- *os 50 maiores bancos, por empréstimo;*
- *maiores bancos de desenvolvimento, por empréstimo;*
- *maiores bancos de investimento, por empréstimo;*
- *maiores financeiras, por financiamento;*
- *maiores empresas de crédito imobiliário, por aplicação;*
- *maiores corretoras, por patrimônio líquido;*
- *maiores distribuidoras, por patrimônio líquido;*
- *maiores seguradoras, por prêmio;*
- *maiores empresas de leasing, por recursos.*

Para as empresas selecionadas foram enviadas carta de apresentação e ou feitos contatos telefônicos com os departamentos financeiro e de informática, marcando-se a entrevista. As instituições financeiras foram entrevistadas no segundo semestre de 1991, no Rio de Janeiro. Apenas uma empresa foi entrevistada em Fortaleza, no primeiro semestre de 1991.

Concluída a etapa de pesquisa de campo foi realizada a análise estatística dos resultados, através do SPSS/PC [SPSS 84].

IV.2 Tratamento Estatístico dos Dados

Os métodos estatísticos referem-se a dados obtidos da observação na forma de medidas ou de contagem, a partir da fonte de tais observações. Os problemas, em sua maioria, agrupam-se em duas categorias: referem-se ou à estimativa de alguma propriedade ou à testes de hipóteses sobre a população.

O tratamento estatístico dos dados coletados vem nos propiciar a verificação das hipóteses levantadas por um método experimental. A análise estatística dos resultados foi realizada através do pacote estatístico SPSS/PC [SPSS 84]. Segue-se a seguinte estratégia:

- *Definição dos dados de acordo com as definições do SPSS/PC;*
- *Construção da massa de dados;*
- *Elaboração de programas, na linguagem do SPSS/PC, segundo os objetivos traçados, para a apresentação dos resultados;*
- *A análise descritiva de cada atributo foi realizada, pelo uso da função “ FREQUENCE ” do SPSS/PC:*
 - *esta função elabora dados estatísticos dos atributos, mostrando as distribuições de freqüências, diagramas de barras e histogramas de freqüências;*
 - *a análise das amostras foi realizada através de medidas de tendência central e de dispersão — média (\bar{x}) e desvio padrão (σ):*
 - * *a média é baseada em todas as observações de uma série, e é menos afetada por flutuações causadas pela amostragem. No entanto, a média tem duas desvantagens principais [PARANTHAMAN 90]:*
 1. *dá mais importância a valores maiores dentro do conjunto que a menores;*
 2. *pode conduzir a conclusões falsas (por exemplo, duas séries, inteiramente diferentes entre si, podem ter a mesma média.*
 - * *o desvio padrão é a medida do grau de dispersão, definida como a raiz quadrada da média dos quadrados dos desvios dos valores observados com relação à média. É a medida mais confiável e mais freqüentemente usada.*
 - *a estratégia de desempate utilizada, na classificação dos itens das tabelas geradas, foi através do menor valor do desvio padrão dos mesmos, como se pode observar no anexo F.*
 - *Com esta amostra foram geradas as tabelas do capítulo V, utilizando-se, aproximadamente, o intervalo de $\bar{x} - 2\sigma$ a $\bar{x} + 2\sigma$.*

A geração da massa de dados foi realizada mediante os resultados da pesquisa de campo, procurando-se garantir que a amostra do universo pesquisado fosse significativa. Foram geradas três massas de dados, assim distribuídas:

- *analistas de sistema*: tomou-se, como base, somente as informações fornecidas pelos analistas de sistema;
- *usuários*: tomou-se, como base, somente as informações fornecida pelos usuários;
- *consolidado*: corresponde a composição de todas as informações colhidas de analistas de sistema e de usuários.

IV.3 Conclusão

Neste capítulo, foram descritos a metodologia de pesquisa, as hipóteses e os objetivos da pesquisa, as técnicas de coleta de dados, o processo de seleção da amostra e o tratamento estatístico desses dados. No próximo capítulo, serão apresentados os resultados obtidos, com a análise estatística da amostra coletada.

Capítulo V

CARACTERÍSTICAS DE UM SOFTWARE FINANCEIRO DE QUALIDADE

Neste capítulo, descrevem-se os resultados da pesquisa de campo, realizada com o objetivo de se identificar os principais atributos de qualidade de um software financeiro. Essa pesquisa foi realizada em 12 instituições financeiras. Os sistemas pesquisados foram desenvolvidos ou mantidos, em sua grande maioria, pelas próprias empresas consultadas. Apenas alguns deles foram adquiridos de outras empresas, mas mantidos pela própria equipe de analistas da empresa adquirente.

V.1 Instituições Financeiras Pesquisadas

A tabela V.1 fornece a distribuição da amostra, com relação ao número de instituições financeiras pesquisadas, enquadrando-as de acordo com a classificação estabelecida na edição da revista Exame Melhores e Maiores de 1991 [EXAME 91]. Vale ressaltar que a maioria das instituições pesquisadas enquadraram-se em mais de uma das classificações, descritas na tabela.

Tabela V.1: Instituições Financeiras Pesquisadas

INSTITUIÇÕES FINANCEIRAS	Quantidade
50 maiores bancos, por empréstimo	07
Maiores bancos de desenvolvimento, por empréstimo	02
Maiores bancos de investimento, por empréstimo	02
Maiores Financeiras, por financiamento	03
Maiores empresas de crédito imobiliário, por aplicação	01
Maiores corretoras, por patrimônio líquido	03
Maiores distribuidoras, por patrimônio líquido	02
Maiores seguradoras, por prêmio	02
Maiores empresas de leasing, por recursos	05
T O T A L	12

V.2 Coleta dos Dados

A coleta de dados da pesquisa de campo estendeu-se a 46 entrevistados, com um total de 99 sistemas pesquisados. Ressalta-se que cada entrevistado podia responder, de acordo com o questionário, até por quatro sistemas. A média geral de sistemas, por entrevistado, foi de 2. Notou-se que os entrevistados responsáveis por mais de um sistema tendiam a repetir seus escores em mais de 60 % dos itens do questionário.

Observou-se as características de implementação dos sistemas financeiros pesquisados: porte do sistema e equipamento em que foi implementado. A tabela V.2 fornece esta distribuição amostral.

Tabela V.2: Características da Implementação dos Sistemas Pesquisados

CARACTERÍSTICAS DE IMPLEMENTAÇÃO	TOTAL
PORTE DO SISTEMA	100,00 %
Pequeno	12,12 %
Médio	37,37 %
Grande	50,51 %
EQUIPAMENTO	100,00 %
Microcomputador	7,07 %
Minicomputador	8,08 %
Mainframe	84,85 %

Os sistemas financeiros pesquisados são listados na tabela V.3, a partir das informações cadastrais constantes nos questionários (anexos *BeD*). Vários destes sistemas, que eram semelhantes em funções nas diversas empresas pesquisa-

Tabela V.3: Tabela dos Sistemas Financeiros Pesquisados

SISTEMAS PESQUISADOS	Qde Analista	Qde Usuário	Qde Sist.
Avaliação Financeira Gerencial	03	02	05
Cobrança	04	02	06
Contabilidade	03	04	07
Contas a Pagar	02	01	03
Controle de Aplic. de Clientes	03	04	07
Controle da Carteira de Ações	02	04	06
Controle de Contratos Passivos	01	01	02
Controle de Corretoras	02	02	04
Controle de Garantia de Penhor	01	–	01
Controle Orçamentário	02	02	04
Crédito a Clientes	05	03	08
Depósito a Prazo (CDB/RDB)	01	02	03
Fluxo de Caixa	05	02	07
Fundos de Investimento	05	08	13
Gerência de Taxas Financeiras	02	02	04
Letras Hipotecárias	–	01	01
Operações de Open Market	04	08	12
Poupança	01	01	02
Seguros	02	–	02
Sistema Nacional de Debêntures	01	01	02
T O T A L	22	24	99

das, e que tinham nomes diferentes, foram agrupados em um nome padrão.

V.3 Análise dos Dados Coletados

A análise estatística descrita no capítulo anterior, foi realizada para cada atributo de qualidade dos sistemas financeiros pesquisados. *Crêterios* e *subfatores* foram analisados em separado e em conjunto, tanto para os dados coletados dos *usuários*, quanto para os dados colhidos dos *analistas*. Analisou-se todos os elementos da população, em conjunto, que foi denominado de *consolidado*.

Nas tabelas, que serão analisadas a seguir, os requisitos de qualidade de um software financeiro são agrupados em intervalos de pesos, baseados na descrição da subseção IV.1.3. Devido a grande incidência de pesos elevados foi incluído, nesta escala de valores, o seguinte peso: 4,5 – MA – Muito Alto. Serão analisados, apenas, os atributos de qualidade considerados mais importantes, para

Tabela V.4: Critérios do Objetivo Confiabilidade da Representação

Requisitos de Qualidade de Software Financeiro

CRITÉRIOS	Analista Sistema	Usuário	Consolidado
Acessibilidade	A – MA	–	A – MA
Acoplamento	ME – MA	–	ME – MA
Aderência das estruturas	ME	–	ME
Auto-descritividade	ME – A	–	ME – A
Balanceamento	ME – A	–	ME – A
Coesão	ME – MA	–	ME – MA
Complexidade estática do programa	B – ME	–	B – ME
Documentação	ME – A	–	ME – A
Domínio das interfaces	B – A	–	B – A
Endentação	ME	–	ME
Especificidade	A – MA	–	A – MA
Estar atualizado	A – E	–	A – E
Fidelidade padrões codificação	ME – A	–	ME – A
Isolamento	ME – MA	–	ME – MA
Localização externa	ME	–	ME
Localização interna	ME – A	–	ME – A
Memória de estado	A – E	–	A – E
Ordenamento dos procedimentos	ME – A	–	ME – A
Organização da documentação	ME	–	ME
Relato de erro	ME – A	–	ME – A
Significância das identificações	ME – A	–	ME – A
Simplicidade do código	ME	–	ME
Transparência estrutural	ME	–	ME
Uso de comentários	ME	–	ME
Uso de linguagem concisa	A	–	A
Uso de técn. de prog. estruturada	ME – A	–	ME – A
Uso de terminologia uniforme	ME – A	–	ME – A

cada intervalo. A análise de todos esses atributos, em todas as tabelas, tornar-se-ia um processo repetitivo, uma vez que muitos destes são comuns a várias tabelas. Além disto, o significado e o processo de avaliação dos mesmos estão descritos no capítulo *II*.

V.3.1 Critérios do Objetivo Confiabilidade da Representação

A tabela V.4 apresenta os resultados, classificados em ordem alfabética dos critérios, obtidos para o objetivo *Confiabilidade da Representação*. Como não é atribuição dos usuários a implementação, em computador, de seus sistemas, então o conjunto de

requisitos de qualidade desse objetivo foi aplicado, somente, aos analistas de sistemas. Isto explica porque os valores, no *consolidado*, são os mesmos obtidos dos analistas de sistema. Chama-se a atenção para o fato de que todos os atributos de qualidade, dessa tabela, têm pesos variando de médio a extremamente alto. Os analistas preocupam-se, realmente, com a qualidade na produção de sistemas financeiros. Os dados estatísticos, que serviram de base para a elaboração dessa tabela, estão no anexo *F*, na tabela F.1.

Essa tabela, por listar um conjunto significativo de atributos de qualidade, e seus respectivos graus de importância, pode orientar os analistas de sistemas financeiros, de que maneira devem ser dispendidos os esforços, na codificação de seus programas. Pode-se perceber que os critérios *memória de estado* e *estar atualizado*, têm pesos, que variam de alto a extremamente alto. Isto nos leva a inferir que o desenvolvedor espera que os módulos do sistema estejam sempre atualizados, e executem suas funções, quando chamados, como se fosse a primeira vez, não possuindo memória prévia de sua existência.

Os atributos de qualidade *acessibilidade* e *especificidade* têm peso variando de alto a muito alto. As alterações realizadas nos sistemas financeiros sempre exigem pressa, logo, é muito importante que os códigos fonte e suas documentações estejam disponíveis e que possam ser reproduzidas facilmente. Além disto, o programa deve estar testado o suficiente, para que a escolha de certas condições, no processamento deste, não o leve a estados indeterminados e que não tenham sido previstos, deixando o sistema vulnerável, e à mercê do acaso.

Os critérios *acoplamento*, *coesão* e *isolamento* variam de um grau de importância de médio a muito alto. Todos esses critérios pertencem ao subfator modularidade. Uma estrutura de módulos, altamente independente, é perseguida pelos desenvolvedores. É a estratégia do dividir, para conquistar. Dividir-se o programa em módulos funcionais, possuindo estes um baixo acoplamento entre si, e não sendo necessário o conhecimento das características internas dos mesmos, para que seja possível utilizá-los, corretamente.

Os requisitos *auto-descritividade*, *balanceamento*, *documentação*, *fi-*

delidade aos padrões de codificação, localização interna, ordenamento dos procedimentos, relato de erro, significância das identificações, uso de técnicas de programação estruturada e uso de terminologia uniforme têm peso variando de médio a alto. Os analistas demonstraram grande aceitação no que se refere à elaboração de normas, pela empresa, que padronizassem a codificação dos programas, inclusive a uniformização do uso de nomes de programas, de arquivos, de telas e de relatórios. Eles usam, largamente, as referências cruzadas e as consideram altamente desejáveis nos programas, para a *localização interna* de seus componentes. O uso de *técnicas de programação estruturada* já faz parte da cultura geral das instituições. No entanto, ainda existem desenvolvedores que possuem uma técnica própria de codificação de seus programas e de elaboração de seus sistemas, e parecem resistir a qualquer mudança neste sentido, que possa vir a por em risco seu domínio da situação.

Os atributos de qualidade *aderência das estruturas, endentação, localização externa, organização da documentação, simplicidade de código, transparência estrutural e uso de comentários*, segundo os resultados obtidos, têm importância média na codificação de um software financeiro. O uso de dicionário de dados, que fornece substancial ajuda na localização dos relacionamentos entre os componentes do programa, vem-se difundindo. O grande problema encontrado, nas empresas pesquisadas, é a falta de um ambiente, que possa dar suporte automático a todas as etapas do ciclo de vida do produto. Existe, por exemplo, uma dicotomia entre especificação e projeto; projeto e codificação. Como as alterações no código são mais imediatas e urgentes, não existindo algo que as ligue às especificações e ao projeto, estes tendem a ficar desatualizados.

V.3.2 Critérios do Objetivo Utilizabilidade

A tabela V.5 apresenta os resultados obtidos para o objetivo *Utilizabilidade* de todas os elementos da população, em ordem alfabética de critérios. Alguns destes requisitos de qualidade dizem respeito somente aos analistas de sistema, pois representam a maneira de como o código fonte é utilizado. Neste caso, os valores obtidos dos analistas são os mesmos do consolidado. Os dados estatísticos, que serviram de base para a construção dessa tabela, estão no anexo F, na tabela F.2.

Tabela V.5: Critérios do Objetivo Utilizabilidade

Requisitos de Qualidade de Software Financeiro

CRITÉRIOS	Analista Sistema	Usuário	Consolidado
Alterabilidade corretiva	A – MA	A – E	A – MA
Alterabilidade evolutiva	ME – A	ME – MA	ME – MA
Apresentação	ME – A	ME	ME
Conformação com o ambiente	B – ME	B	B – ME
Composição tecnológica	ME – A	ME – A	ME – A
Consistência externa	ME – MA	–	ME – MA
Consistência interna	ME – MA	–	ME – MA
Documentação	ME	ME – A	ME
Efetividade dos comentários	ME	–	ME
Estabilidade	ME – A	–	ME – A
Facilidade de uso	ME – MA	ME – A	ME – A
Identif. dos componentes	ME – A	–	ME – A
Independência de hardware	MB – B	MB – B	MB – B
Independência de software	MB – B	MB – B	MB – B
Independ. do tipo de dados	B – ME	–	B – ME
Independ. quantidade de dados	ME – MA	ME – MA	ME – MA
Legibilidade do programa	ME – A	–	ME – A
Localização externa	ME – A	–	ME – A
Localização interna	ME – A	–	ME – A
Não redundância de dados	ME – A	–	ME – A
Organiz. da documentação	ME	ME	ME
Otimiz. do armazenamento	ME	–	ME
Recuper. dos componentes	ME	–	ME
Seleção de auxílios	B – A	ME	B – A
Simplicidade do código	ME	–	ME
Taxa de retorno	B	B – ME	B – ME
Tempestividade	A – E	A – E	A – E
Tempo de troca de comandos	MB – B	–	MB – B
Tempo de troca de mensagens	B – ME	–	B – ME

Essa tabela demonstra que o grau de exigência dos usuários, em relação aos analistas, varia de acordo com o atributo de qualidade observado. Os usuários são mais exigentes nos requisitos de qualidade: *alterabilidade corretiva*, *alterabilidade evolutiva*, *documentação* e *taxa de retorno*. A alterabilidade do sistema favorece os usuários, no sentido de receberem os resultados de suas solicitações mais rapidamente. A *documentação* serve como instrumento de verificabilidade do sistema. Apesar de os custos/benefícios do sistema não possuírem peso significativo tanto para analistas, quanto para usuários, estes últimos tendem a considerar melhor este aspecto. Isto se evidencia, quando um sistema começa a dar muitos problemas de manutenção, comprometendo o seu desempenho. Esse sistema tende a ser descontinuado ou substituído por um outro, onde se exerça menor esforço de utilização com maiores e melhores resultados.

Os desenvolvedores são mais exigentes do que os usuários nos atributos de qualidade *apresentação*, *conformação com o ambiente* e *facilidade de uso*. A *apresentação* do sistema, traduzida no uso de interfaces gráficas ou de outras características, facilita o entendimento com os usuários. Os analistas tendem a fazer uso de facilidades do ambiente, que venham a oferecer alguma vantagem seja esta, por exemplo, na eficiência de execução ou na otimização do armazenamento do código ou dos dados. O critério *facilidade de uso* tem peso variando de médio a muito alto para os analistas, uma vez que estes precisam, usualmente, operacionalizar o sistema de uma maneira rápida e eficaz. Para os usuários, é desejável que o software seja de fácil utilização.

Os atributos de qualidade *composição tecnológica*, *independência de hardware*, *independência de software*, *independência da quantidade de dados*, *organização da documentação* e *tempestividade* apresentam um consenso entre analistas e usuários. Não é relevante se o sistema é processado em um determinado tipo equipamento ou é executado debaixo de um certo sistema operacional. O que importa, para ambos, é que esse ambiente permita que o software produza resultados satisfatórios, no tempo oportuno.

Computando-se os dados de todos os elementos da população, nota-se que os resultados do consolidado ora favorecem as opiniões dos analista, ora

Tabela V.6: Critérios do Objetivo Confiabilidade Conceitual

Requisitos de Qualidade de Software Financeiro

CRITÉRIOS	Analista Sistema	Usuário	Consolidado
Acurácia	E	E	E
Complectude prog. com relação às funções requer. pelo usuário	ME – MA	ME – MA	ME – MA
Complectude prog. com relação às normas elabor. cód. fonte organiz.	ME – A	ME – A	ME – A
Confidenciabilidade	A – E	MA – E	A – E
Confinamento de falhas	A – E	A – E	A – E
Correção	E	E	E
Detectabilidade de falhas	A – E	A – E	A – E
Detectabilidade de violação	A – E	A – E	A – E
Imputabilidade	A – MA	A – E	A – E
Necessidade funções implementadas	A – E	MA – E	A – E
Privacidade	A – MA	A – MA	A – MA
Ratificabilidade	A – MA	ME – MA	ME – MA
Recuperabilidade	ME – MA	A – E	A – MA
Redundância	A – E	A – E	A – E
Reparabilidade	A – MA	A – E	A – E
Sensibilidade	ME – MA	ME – MA	ME – MA
Tolerância entrada dados incorretos	ME – A	ME – MA	ME – MA
Vulnerabilidade	A – E	MA – E	A – E

favorecem os pareceres dos usuários, compondo um equilíbrio entre ambos. É esse equilíbrio, que deve ser atingido pela equipe envolvida na geração de um produto de software. Ítens como *alterabilidade corretiva*, *conformação com o ambiente e documentação* refletem melhor o posicionamento dos analistas, no consolidado. A correção de um erro, em um programa, por exemplo, inquieta muito mais os analistas, que são pressionados tanto por seus usuários, quanto por seus superiores imediatos. A *alterabilidade evolutiva*, por exemplo, tem peso muito alto para os usuários e, portanto, para o consolidado. Deve ser dada, portanto, importância muito alta a esse requisito de qualidade, uma vez que os usuários dos sistemas financeiros possuem a visão crítica do mercado, e são sensíveis à dinâmica do mesmo, buscando, na utilização de seus produtos de software, alcançarem a melhor produtividade e competitividade para a empresa.

V.3.3 Critérios do Objetivo Confiabilidade Conceitual

A tabela V.6 apresenta os resultados obtidos de todos os elementos da população, para o objetivo *confiabilidade conceitual*, em ordem alfabética de critérios. Aqui, todos os requisitos de qualidade selecionados foram pesquisados tanto para analistas, quanto para usuários. Os dados estatísticos, que serviram de base para a construção dessa tabela, estão no anexo F, na tabela F.3.

Os usuários atribuíram pesos iguais ou maiores que os analistas, a todos os critérios desse objetivo, a exceção da *ratificabilidade*. Isto demonstra que os usuários de software financeiro esperam que seus sistemas, quando implementados, reproduzam o que foi especificado e projetado. Um outro fato, que merece destaque, é que quase todos esses atributos de qualidade possuem, em sua maioria, pesos muito elevados. Isto exige dos analistas maior empenho e cuidados adicionais, no desenvolvimento destes produtos.

Os itens *acurácia* e *correção* são vitais e devem ser perseguidos a todo custo. São considerados de extrema importância tanto para analistas, quanto para usuários. Não se admite, portanto, que um software financeiro não seja suficientemente exato nos resultados gerados, e que contenha fórmulas de cálculo erradas ou desatualizadas. Será apresentada, até o final desta subseção, a análise dos requisitos de qualidade de software financeiro, tendo-se em conta, somente, o posicionamento do consolidado.

Os atributos de qualidade *confidenciabilidade*, *confinamento de falhas*, *detectabilidade de falhas*, *detectabilidade de violação*, *imputabilidade*, *necessidade das funções implementadas*, *redundância*, *reparabilidade* e *vulnerabilidade* têm grau de importância que varia de alto a extremamente alto. Isto significa que um software financeiro deve ser robusto e atender às normas de segurança pré-estabelecidas. Entenda-se por um programa robusto aquele, que é capaz de enfrentar situações hostis, sem que isto afete sua integridade. Como uma extensão da robustez e da segurança de que devam ser dotados os sistemas financeiros, está o processo sistemático de copiar os módulos mais importantes do sistema e seus dados. Deste modo, gera-se uma *redundância* benéfica de informações, atuando como agente de prevenção de possíveis falhas operacionais. É indispensável, que as funções imprescindíveis ao alcance dos objetivos do projeto sejam implementadas oportunamente,

para que o sistema, adquirindo sua própria identidade, possa assegurar ao usuário que o produto de software, que está sendo gerado, é o que foi solicitado.

Os critérios em privacidade *recuperabilidade* têm pesos variando de alto a muito alto. É altamente desejável, que haja mecanismos de proteção, de controle e de correção de cada operação realizada no sistema. Para as situações previsíveis de erro é desejável, que a correção dos mesmos seja feita de maneira automática, pelo próprio programa.

Os atributos de qualidade *completude do programa com relação às funções requeridas pelo usuário, ratificabilidade, sensibilidade e tolerância à entrada de dados incorretos* possuem pesos variando de médio a muito alto. Os usuários esperam que todas as suas solicitações sejam implementadas, tanto na época do desenvolvimento do produto, quanto na fase de operação do mesmo. A *ratificabilidade*, nos programas, permite a realimentação e posterior validação dos dados de entrada pelo operador. Neste caso, os analistas são favorecidos, pois é diminuída ou até eliminada as intervenções dos mesmos, em possíveis correções dos dados de entrada.

A *completude do programa com relação às normas de elaboração de códigos fonte da organização* tem grau de importância variando de médio a alto. Há uma tendência crescente em se padronizar, na empresa, a codificação de programas e até todo o processo de desenvolvimento de sistemas. Ressente-se, ainda, da falta de uma efetiva e eficaz monitoração em relação ao cumprimento dessas normas.

V.3.4 Classificação dos Critérios obtidos dos Analistas de Sistema

A tabela V.7 mostra a classificação dos critérios, em ordem descendente de valores obtidos dos analistas, independentemente dos objetivos de qualidade a que pertençam. Embora muitos destes requisitos de qualidade já tenham sido analisados anteriormente, faz-se, aqui, uma nova avaliação dos aspectos mais significativos dos mesmos, na concepção dos analistas de sistema. Os dados estatísticos, que serviram de base para a geração dessa tabela, estão no anexo *F*, na tabela F.4.

Tabela V.7: Classificação dos critérios obtidos dos Analistas de Sistema

Requisitos de Qualidade de Software Financeiro	
Acurácia	E
Correção	E
Confidenciabilidade	A – E
Necessidade funções implementadas	A – E
Vulnerabilidade	A – E
Redundância	A – E
Estar atualizado	A – E
Detectabilidade de violação	A – E
Memória de estado	A – E
Confinamento de falhas	A – E
Detectabilidade de falhas	A – E
Tempestividade	A – E
Imputabilidade	A – MA
Acessibilidade	A – MA
Especificidade	A – MA
Reparabilidade	A – MA
Ratificabilidade	A – MA
Privacidade	A – MA
Alterabilidade corretiva	A – MA
Consistência interna	ME – MA
Uso de linguagem concisa	A
Facilidade de uso	ME – MA
Recuperabilidade	ME – MA
Coesão	ME – MA
Sensibilidade	ME – MA
Consistência externa	ME – MA
Isolamento	ME – MA
Acoplamento	ME – MA
Compleitude prog. com relação às funções requer. pelo usuário	ME – MA
Independ. quantidade de dados	ME – MA
Tolerância entrada dados incorretos	ME – A
Estabilidade	ME – A
Documentação (R)	ME – A
Uso de terminologia uniforme	ME – A
Ordenamento dos procedimentos	ME – A
Fidelidade padrões codificação	ME – A
Relato de erro	ME – A
Alterabilidade evolutiva	ME – A
Auto-descritividade	ME – A
Significância das identificações	ME – A

Tabela V.7: Continuação

Requisitos de Qualidade de Software Financeiro	
Não redundância de dados	ME – A
Localização interna (R)	ME – A
Balanceamento	ME – A
Identif. dos componentes	ME – A
Composição tecnológica	ME – A
Legibilidade do programa	ME – A
Compleitude prog. com relação às normas elabor. códigos fonte organiz.	ME – A
Uso de técn. de prog. estruturada	ME – A
Apresentação	ME – A
Localização externa (U)	ME – A
Localização interna (U)	ME – A
Recuper. dos componentes	ME
Organização da documentação (R)	ME
Aderência das estruturas	ME
Localização externa (R)	ME
Transparência estrutural	ME
Uso de comentários	ME
Simplicidade de código (R)	ME
Otimiz. do armazenamento	ME
Efetividade dos comentários	ME
Simplicidade do código (U)	ME
Seleção de auxílios	B – A
Organiz. da documentação (U)	ME
Documentação (U)	ME
Domínio das interfaces	B – A
Endentação	ME
Complexidade estática do programa	B – ME
Conformação com o ambiente	B – ME
Independ. do tipo de dados	B – ME
Tempo de troca de mensagens	B – ME
Taxa de retorno	B
Tempo de troca de comandos	MB – B
Independência de software	MB – B
Independência de hardware	MB – B

Embora um produto de software precise atingir os objetivos de qualidade *confiabilidade da representação*, *utilizabilidade*, e *confiabilidade conceitual*, estes não são alcançados em momentos distintos, existindo um inter-relacionamento entre eles. Alguns critérios são comuns tanto ao objetivo *utilizabilidade*, quanto ao objetivo *confiabilidade da representação*, sendo distinguidos por (U) ou (R), respectivamente. A grande maioria desses critérios tem peso elevado, demonstrando o grande interesse dos analistas em atenderem a seus usuários em suas necessidades, e em tempo hábil. Isto não somente para evitar desgastes entre ambos, mas, sobretudo, por terem consciência do valor desempenhado por um software financeiro, dentro da instituição.

A *acurácia* e a *correção* são os atributos de qualidade considerados de extrema importância. Os sistemas financeiros devem fornecer informações precisas, pois é em cima destes dados, que os usuários se apoiam para tomarem suas decisões. Esta é uma questão vital, que não pode ser relegada a segundo plano, ou ter a sua importância diminuída.

Os critérios, a seguir, têm peso variando de alto a extremamente alto: *confidenciabilidade*, *necessidade das funções implementadas*, *vulnerabilidade*, *redundância*, *estar atualizado*, *detectabilidade de violação*, *memória de estado*, *confinamento de falhas*, *detectabilidade de falhas* e *tempestividade*. O item *estar atualizado* nos diz que o código fonte deve estar em sua versão mais atualizada, incluindo as últimas alterações efetuadas. Isto é tão importante quando a robustez e a segurança do programa. Por conseguinte, o esforço de gerar resultados satisfatórios e dentro de um padrão de segurança, seria em vão, se o código fonte estivesse desatualizado. O item *redundância* seja, talvez, um dos mais, efetivamente, utilizados. É consensual, entre analistas, que se deva ter cópias dos módulos mais críticos do sistema, para que sejam utilizados, em casos de falhas dos módulos em operação. Na verdade, costuma-se ter cópias de todos os módulos do sistema e da totalidade de seus dados. Também, é muito aceita pelos analistas a implementação das funções de um programa, em uma estrutura de módulos. Um módulo, por exemplo, que possua *memória de estado* é, via de regra, imprevisível. Embora este seja chamado com os mesmos dados de entrada, fornece resultados diferentes a cada chamada,

acarretando problemas de operação e de manutenção.

Os atributos de qualidade *imputabilidade, acessibilidade, especificidade, reparabilidade, ratificabilidade, privacidade e alterabilidade corretiva* têm grau de importância, que varia de alto a muito alto. Mais uma vez, os ítems de qualidade relacionados com a segurança e com a robustez do programa têm seu valor evidenciado. O fácil acesso aos códigos fonte e às suas respectivas documentações é altamente desejável, pois permitirá que estes sejam alterados com rapidez. O programa deverá ser manuseado por várias pessoas ao longo de sua vida útil, portanto, é necessário que esteja sempre disponível, possa ser reproduzido facilmente, após cada alteração e quando se necessitar cópias do mesmo.

O critério *uso de linguagem concisa* tem importância alta. Vale salientar que a maioria dos sistemas pesquisados utilizam linguagem estruturada. A linguagem de programação em que está codificado o programa é importante, pois o número de declarações ou de comandos é o próprio código fonte do programa, implementando suas funções. Quando um único comando representa o equivalente a muitas linhas de código, fica fácil compreender seu significado e seus objetivos.

Os seguintes critérios possuem seus pesos variando de médio a muito alto: *facilidade de uso, recuperabilidade, coesão, sensibilidade, consistência externa, isolamento, acoplamento, completude do programa com relação às funções requeridas pelo usuário e independência da quantidade de dados*. Neste conjunto, os analistas ressaltam, mais uma vez, seu interesse pela modularidade. O intuito de se buscar a modularidade é a necessidade de se procurar vencer a complexidade do programa, tornando-o legível, verificável, manutenível e reutilizável. Diz-se que um programa está bem particionado, quando há um baixo acoplamento entre seus módulos. O que se busca no módulo é uma forte coesão entre seus elementos, e que apenas o conhecimento das definições deste, baste para que ele seja bem utilizado. Portanto, as funções do sistema podem ser implementadas em módulos, facilitando o controle quanto à completude do mesmo em relação às solicitações do usuário.

Os critérios, a seguir, têm peso variando de médio a alto: *tolerância a entrada de dados incorretos, estabilidade, documentação (R), uso de terminolo-*

gia uniforme, ordenamento dos procedimentos, fidelidade aos padrões de codificação, relato de erro, alterabilidade evolutiva, auto-descritividade, significância das identificações, não redundância de dados, localização interna (R), balanceamento, identificação dos componentes, composição tecnológica, legibilidade do programa, completude do programa com relação às normas de elaboração dos códigos fonte da organização, uso de técnicas de programação estruturada, apresentação, localização externa (U) e localização interna (U). Nota-se, neste grupo, a preocupação dos analistas com a legibilidade do programa. Um programa claro, simples, com comentários explícitos e adequados de seus componentes e de seu funcionamento e seguindo a uma padronização estabelecida pela empresa, torna mais fácil e rápida sua leitura. Informações repetidas, no código fonte, desviam a atenção dos responsáveis pela manutenção para questões menos importantes, dificultando o entendimento do mesmo. Portanto, um programa precisa ser legível e, facilmente, compreendido ao longo de toda sua vida útil.

Os seguintes requisitos de qualidade têm peso médio: *recuperação dos componentes, organização da documentação (R), aderência das estruturas, localização externa (R), transparência estrutural, uso de comentários, simplicidade de código (R), otimização do armazenamento, efetividade dos comentários, simplicidade do código (U), organização da documentação (U), documentação (U) e documentação.* Neste conjunto, destaca-se a importância da *documentação* e da *organização da documentação*. Todos comentam que a documentação de um sistema é necessária, porém a classificam como de importância mediana. Ainda existe o mal costume de analistas de sistema e de usuários de fazerem de suas memórias, partes dos arquivos de documentação. Não que isto seja um mal, por si mesmo. O problema se apresenta, quando a memória humana é o único ou o mais importante meio de documentação existente no sistema. Documentar, organizar a documentação consome tempo. No entanto, não é um tempo gasto inutilmente — é um investimento indispensável, para garantir a qualidade do software gerado.

Os itens *complexidade estática do programa, conformação com o ambiente, independência do tipo de dados e tempo de troca de mensagens* possuem pesos, que variam de baixo a médio. De uma maneira geral, os analistas de software

financeiro dão pouca importância ao tipo ou à quantidade de operadores e de operandos, que possa ter um programa. Sabe-se, no entanto, que um código fonte com vocabulário extenso dificulta o entendimento e a manutenção do mesmo.

Os critérios *taxa de retorno*, *tempo de troca de comandos*, *independência de software* e *independência de hardware* estão no intervalo de pesos de muito baixo a baixo. O custo dispendido com o processamento de dados, no desenvolvimento e operação do sistema, não aflige os analistas. O modelo do computador ou da versão do sistema operacional utilizados, na empresa, não causam inquietações, desde que não haja mudanças bruscas de equipamentos ou substituições de software básico, em curto espaço de tempo.

V.3.5 Classificação dos Critérios obtidos dos Usuários

A tabela V.8 mostra a classificação dos critérios, em ordem descendente de valores, obtidos dos usuários, independentemente dos objetivos de qualidade a que pertençam. Vejamos, então, os pontos de vista dos usuários em relação a seus atributos de qualidade. Os dados estatísticos, que serviram de base para a construção dessa tabela, estão no anexo F, na tabela F.5.

Os usuários esperam que o produto de software atenda às suas reais necessidades, reproduzindo seus anseios e seus objetivos, tempestivamente. Isto justifica que a grande maioria desses critérios tem peso elevado. Eles compartilham com os analistas de sistema que o principal atributo de qualidade de um software financeiro é a precisão, traduzida nos critérios *acurácia* e *correção*, recebendo de ambos o peso máximo estabelecido.

Os critérios *vulnerabilidade*, *confidenciabilidade* e *necessidade das funções implementadas* têm seus pesos variando no intervalo de muito alto a extremamente alto. Isto traduz a preocupação dos usuários com a existência de mecanismos de segurança, que tornem o sistema confiável, garantido a proteção das informações geradas, não sendo estas acessadas por órgãos ou por indivíduos não autorizados. Para os usuários, existem certas funções imprescindíveis, sendo estas consideradas vitais para o bom desempenho do sistema. É necessário, portanto,

Tabela V.8: Classificação dos critérios obtidos dos Usuários

Requisitos de Qualidade de Software Financeiro	
Acurácia	E
Correção	E
Vulnerabilidade	MA - E
Confidenciabilidade	MA - E
Necessidade funções implementadas	MA - E
Detectabilidade de violação	A - E
Confinamento de falhas	A - E
Detectabilidade de falhas	A - E
Tempestividade	A - E
Reparabilidade	A - E
Imputabilidade	A - E
Recuperabilidade	A - E
Redundância	A - E
Alterabilidade corretiva	A - E
Privacidade	A - MA
Sensibilidade	ME - MA
Ratificabilidade	ME - MA
Independ. quantidade de dados	ME - MA
Alterabilidade evolutiva	ME - MA
Tolerância entrada dados incorretos	ME - MA
Completo prog. com relação às funções requer. pelo usuário	ME - MA
Facilidade de uso	ME - A
Composição tecnológica	ME - A
Completo prog. com relação às normas elabor. códigos fonte organiz.	ME - A
Documentação	ME - A
Apresentação	ME
Organiz. da documentação	ME
Seleção de auxílios	ME
Taxa de retorno	B - ME
Conformação com o ambiente	B
Independência de hardware	MB - B
Independência de software	MB - B

que os analistas percebam, desde cedo, essas funções e as implementem em tempo hábil, para estreitar o bom relacionamento entre eles e os usuários. No entanto, é fundamental que, desde o princípio, os usuários definam, clara e objetivamente, suas necessidades.

Os atributos de qualidade, a seguir, têm pesos variando de alto a extremamente alto: *detectabilidade de violação, confinamento de falhas, detectabilidade de falhas, tempestividade, reparabilidade, imputabilidade, recuperabilidade, redundância e alterabilidade corretiva*. Além da robustez e da segurança, o grau de oportunidade de um software financeiro é altíssimo. Os resultados do sistema deverão estar disponíveis para o usuário com *tempestividade* ou então poderão perder seu grau de competitividade, e tornarem-se extemporâneos. O mercado financeiro é dinâmico e volátil, exigindo rapidez e flexibilidade na utilização das informações geradas, influenciando, assim, no volume de manutenção de seus produtos de software. Por conseguinte, quando surgem alterações, sejam adaptativas ou corretivas, são, na maioria das vezes, de caráter emergencial – daí a pressa exigida, sempre, pelos usuários destes produtos.

O critério *privacidade* tem peso variando de alto a muito alto. Os usuários necessitam de um mecanismo eficaz de proteção e de controle das operações realizadas por determinado indivíduo, no manuseio do software. Os sistemas possuem um conjunto de funções, que podem ser agrupadas em subconjuntos menores. Cada um desses subconjuntos pode ser vinculado a um nível hierárquico de responsabilidade, na empresa. Por exemplo, as operações realizadas por um gerente não poderão ser executadas por alguém de uma hierarquia inferior e assim por diante, assegurando, desta forma, a *privacidade* na utilização do software.

Os seguintes critérios têm peso variando de médio a muito alto: *sensibilidade, ratificabilidade, independência da quantidade de dados, alterabilidade evolutiva, tolerância à entrada de dados incorretos e completude do programa com relação às funções requeridas pelo usuário*. Os usuários desejam que seus produtos tenham indicações da importância das informações processadas. Isto significa que, quando o usuário for executar uma determinada função do sistema, ele tenha a noção exata daquilo que está processando, através de alguma indicação explícita no sistema.

Sabe-se que uma empresa objetiva obter, progressivamente, uma fatia maior do mercado. Mais clientes significa, teoricamente, melhor aceitação do produto pelo consolidado ou maiores vantagens oferecidas pelo produto. A princípio, não deve haver limites em relação ao montante de clientes, que operem em uma determinada aplicação. Se, por algum motivo, esta quantidade de clientes vier a incrementada, o sistema deve, facilmente, absorver essa nova massa de dados.

Os atributos de qualidade *facilidade de uso, composição tecnológica, completude do programa com relação às normas de elaboração de códigos fonte da organização e documentação* têm grau de importância variando de médio a alto. Quando o usuário necessita de alguma informação parecida com as já fornecidas, ele espera que o sistema seja adaptável, o suficiente, para conseguir seu intento com rapidez. Eles demonstraram-se, também, receptivos às padronizações de nomes e da diagramação de telas e de relatórios e manifestaram grande interesse que seus produtos tivessem interface amigável.

Os seguintes requisitos de qualidade *apresentação, organização da documentação e seleção de auxílios* têm importância média para os usuários. É desejável que a documentação do sistema esteja bem organizada, de modo a facilitar o manuseio de suas informações. Alguns usuários consideraram, na pesquisa, a tarefa de documentar, como sendo exclusiva dos analistas de sistema, embora não se furtem de auxiliá-los neste trabalho, quando solicitados. A *documentação* do sistema deveria ser um trabalho, efetivamente, realizado por ambos e iniciado desde as especificações. Outro item bem aceito pelos usuários é a *seleção de auxílios*, principalmente, em sistemas, que utilizam telas, facilitando a navegação no mesmo.

Os critérios *taxa de retorno, conformação com o ambiente, independência de hardware e independência de software* têm pouca importância para os usuários. Eles não se importam se as informações vêm de microcomputador ou de *mainframe*, mas se elas são precisas e estão disponíveis no tempo oportuno. O que, realmente, interessa é a confiabilidade das informações geradas.

V.3.6 Classificação dos Critérios obtidos no Consolidado

Tabela V.9: Classificação dos critérios obtidos do Consolidado

Requisitos de Qualidade de Software Financeiro	
Acurácia	E
Correção	E
Confidenciabilidade	A – E
Necessidade funções implementadas	A – E
Vulnerabilidade	A – E
Detectabilidade de violação	A – E
Confinamento de falhas	A – E
Estar atualizado	A – E
Detectabilidade de falhas	A – E
Memória de estado	A – E
Redundância	A – E
Tempestividade	A – E
Imputabilidade	A – E
Reparabilidade	A – E
Recuperabilidade	A – MA
Acessibilidade	A – MA
Especificidade	A – MA
Alterabilidade corretiva	A – MA
Privacidade	A – MA
Ratificabilidade	ME – MA
Consistência interna	ME – MA
Sensibilidade	ME – MA
Uso de linguagem concisa	A
Independ. quantidade de dados	ME – MA
Coesão	ME – MA
Consistência externa	ME – MA
Isolamento	ME – MA
Acoplamento	ME – MA
Tolerância entrada dados incorretos	ME – MA
Completude prog. com relação às funções requer. pelo usuário	ME – MA
Alterabilidade evolutiva	ME – MA
Facilidade de uso	ME – A
Estabilidade	ME – A
Documentação (R)	ME – A
Uso de terminologia uniforme	ME – A
Ordenamento dos procedimentos	ME – A
Fidelidade padrões codificação	ME – A
Relato de erro	ME – A
Significância das identificações	ME – A
Auto-descritividade	ME – A

Tabela V.9: Continuação

Requisitos de Qualidade de Software Financeiro	
Não redundância de dados	ME – A
Localização interna (R)	ME – A
Balanceamento	ME – A
Identif. dos componentes	ME – A
Composição tecnológica	ME – A
Legibilidade do programa	ME – A
Uso de técn. de prog. estruturada	ME – A
Compleitude prog. com relação às normas elabor. códigos fonte organiz.	ME – A
Localização externa (U)	ME – A
Localização interna (U)	ME – A
Recuper. dos componentes	ME
Organização da documentação (R)	ME
Aderência das estruturas	ME
Documentação (U)	ME
Localização externa (R)	ME
Apresentação	ME
Transparência estrutural	ME
Uso de comentários	ME
Simplicidade do código (R)	ME
Otimiz. do armazenamento	ME
Efetividade dos comentários	ME
Simplicidade do código (U)	ME
Domínio das interfaces	B – A
Organiz. da documentação (U)	ME
Seleção de auxílios	B – A
Endentação	ME
Complexidade estática do programa	B – ME
Independ. do tipo de dados	B – ME
Conformação com o ambiente	B – ME
Tempo de troca de mensagens	B – ME
Taxa de retorno	B – ME
Tempo de troca de comandos	MB – B
Independência de software	MB – B
Independência de hardware	MB – B

A tabela V.9 mostra a classificação dos critérios, em ordem descendente de pesos, obtidos de todos os elementos da população, independentemente dos objetivos de qualidade a que pertençam. Alguns critérios são comuns tanto ao objetivo *utilizabilidade*, quanto ao objetivo *confiabilidade da representação*, sendo distinguidos por (U) ou (R), respectivamente. A grande maioria destes atributos de qualidade possuem altos pesos, como não poderia ser diferente, pois representam a composição dos dados obtidos de analistas e de usuários, examinados anteriormente. Aqui, faz-se uma nova análise de todo o conjunto dessas informações. Os dados estatísticos, que serviram de base para a construção dessa tabela, estão no anexo F, na tabela F.6.

Confirmando o consenso entre analistas e usuários, a *acurácia* e a *correção* são os requisitos de qualidade mais importantes de um software financeiro. Portanto, um sistema, que não seja preciso, isto é, seus métodos numéricos não são consistentes com os resultados da aplicação, nem os resultados obtidos são coerentes com o que é esperado pelo usuário, é de uso quase que proibitivo, na área financeira. No mínimo, gera muitos conflitos entre analistas e usuários.

Os seguintes atributos de qualidade possuem pesos que variam de alto a extremamente alto: *confidenciabilidade*, *necessidade das funções implementadas*, *vulnerabilidade*, *detectabilidade de violação*, *confinamento de falhas*, *estar atualizado*, *detectabilidade de falhas*, *memória de estado*, *redundância*, *tempestividade*, *imputabilidade* e *reparabilidade*. No consolidado, refletem-se os pesos atribuídos, em tabelas anteriores, à segurança e à robustez de um software financeiro, além da necessidade de que as funções imprescindíveis ao sistema sejam implementadas. O grau de oportunidade de um software financeiro é, sem sombra de dúvidas, uma característica marcante. Os resultados do sistema devem ser tempestivos, sob pena de tornarem-se tardios, ou causarem conflitos entre analistas de sistema e usuários, ou até prejuízos para a empresa. Há uma consciência clara de que negligências podem causar danos ao patrimônio da instituição, e afetar a própria saúde financeira da empresa, uma vez que, normalmente, são envolvidas vultosas quantias pecuniárias.

Os critérios *recuperabilidade*, *acessibilidade*, *especificidade*, *alterabilidade corretiva* e *privacidade* têm peso variando de alto a muito alto. Sabe-se que

é fundamental que um software financeiro seja flexível e acessável facilmente, permitindo que alterações sejam realizadas, com facilidade, não somente por quem o desenvolveu, mas, também, por qualquer outro analista.

Os seguintes atributos de qualidade possuem pesos variando de médio a muito alto: *ratificabilidade, consistência interna, sensibilidade, independência da quantidade de dados, coesão, consistência externa, isolamento, acoplamento, tolerância a entrada de dados incorretos, completude do programa com relação às funções requeridas pelo usuário e alterabilidade evolutiva*. Um software financeiro deve ser projetado não somente para as suas necessidades atuais, mas, sempre que possível, deve-se atentar para sua possível evolução tanto da base de dados, quanto da própria estrutura de código. Todavia para que a manutenção ocorra, naturalmente, é necessário que o programa apresente uma notação uniforme, sem conflitos entre os nomes de seus componentes e esteja, de preferência, implementado em uma estrutura de módulos funcionais. Um sistema desenvolvido em de uma estrutura modular tem suas manutenções futuras, consideravelmente, agilizadas.

Os critérios, a seguir, possuem pesos, que variam no intervalo de médio a alto: *facilidade de uso, estabilidade, documentação (R), uso de terminologia uniforme, fidelidade aos padrões de codificação, relato de erro, significância das identificações, auto-descritividade, não redundância de dados, localização interna (R), balanceamento, identificação dos componentes, composição tecnológica, legibilidade do programa, uso de técnicas de programação estruturada, completude do programa com relação às normas de elaboração dos códigos fonte da organização, localização externa (U) e localização interna (U)*. Aqui neste grupo, destaca-se o uso de técnicas de programação estruturada, que são utilizadas para a construção de programas, através de estruturas lógicas de controle e de suas combinações. Essas técnicas são bem aceitas, pois se constata que elas minimizam a quantidade de erros ocorridos, durante a elaboração do programa, diminuem os custos de manutenção, além de gerarem códigos claros, legíveis e bem estruturados. A manutenção é agilizada, também, com o uso de referências cruzadas, facilitando a localização interna de seus componentes.

Alguns critérios são considerados de média importância, para um

software financeiro. São eles: *recuperação dos componentes, organização da documentação (R), aderência das estruturas, documentação (U), localização externa (R), apresentação, transparência estrutural, uso de comentários, simplicidade de código (R), otimização do armazenamento, efetividade dos comentários, simplicidade do código (U), organização da documentação (R) e endentação*. É desejável que um programa não contenha referências a declarações que não sejam, efetivamente, utilizadas e que haja comentários explicativos, para o que não ficar claro, apenas com a leitura do código. É desejável que as estruturas, na linguagem de programação, correspondam às estruturas lógicas do problema. É desejável, que elementos do código fonte, como módulos e sub-rotinas sejam, facilmente, reutilizáveis. É desejável, também, que haja otimização na armazenagem dos dados na empresa, uma vez que os dispositivos de armazenamento são dispendiosos.

Os demais itens de qualidade relacionados, abaixo, pertencem a um intervalo de pesos baixos. São eles: *complexidade estática do programa, independência do tipo de dados, conformação com o ambiente, tempo de troca de mensagens, taxa de retorno, tempo de troca de comandos, independência de software e independência de hardware*. Embora não haja preocupações, quanto aos custos do sistema, este não deve ser menosprezado. É importante, que o software seja bem projetado, desde as suas especificações. Não adianta adiar os problemas para mais tarde. Eles voltarão mais fortes e exigirão mais energia. Corre-se o risco de se passar a vida inteira remendando-se sistemas mal projetados, e implementados de maneira sofrível. Um sistema bem produzido, em tese, não depende do equipamento, onde foi instalado. Obviamente, um bom *hardware* ajuda no desempenho do mesmo — é sempre uma força que soma, e nunca uma muleta necessária e indispensável para que este funcione. Tornar um programa muito dependente do ambiente, onde executa, pode ser um mal evitável, e não, somente, um incômodo vício.

V.3.7 Classificação dos Atributos de Qualidade por Subfator

A tabela V.10 apresenta os resultados obtidos de todos os elementos da população, em ordem alfabética de subfatores. Existem alguns destes atributos de qualidade,

Tabela V.10: Classificação dos Atributos de Qualidade por Subfator

Requisitos de Qualidade de Software Financeiro			
SUBFATOR	Analista Sistema	Usuário	Consolidado
Adaptabilidade	ME – A	ME – A	ME – A
Alterabilidade	ME – A	A – MA	A – MA
Aplicabilidade	ME – A	–	ME – A
Amenidade de Uso	ME – A	ME	ME
Clareza	ME – A	–	ME – A
Compleitude	ME – A	ME – A	ME – A
Concisão	ME – A	–	ME – A
Configurabilidade	B – ME	B	B – ME
Consistência	ME – MA	–	ME – MA
Disponibilidade	A – MA	–	A – MA
Eficiência de Armazenamento	ME	–	ME
Eficiência de Execução	B – ME	B – ME	B – ME
Estilo	ME	–	ME
Estrutura	ME – A	–	ME – A
Generalidade	ME	ME – MA	ME – A
Independência do Ambiente	MB – B	MB – B	MB – B
Lucratividade	B	B – ME	B – ME
Modularidade	ME – A	–	ME – A
Necessidade	A – E	MA – E	A – E
Oportunidade	ME – E	A – E	A – E
Precisão	E	E	E
Rastreabilidade	ME	–	ME
Robustez	A – MA	A – MA	A – MA
Segurança	A – MA	MA	A – MA
Validabilidade	ME	ME – MA	ME – A
Verificabilidade	ME	ME	ME

que não dizem respeito aos usuários, tendo sido, por conseguinte, pesquisados somente para os analistas de sistema. São aqueles relacionados com a codificação e a utilização dos programas por analistas. Essa tabela nos dá uma visão macroscópica dos atributos de qualidade e seus respectivos pesos obtidos dos analistas, dos usuários e do consolidado. Pode-se identificar em que pontos os analistas e usuários falam a mesma linguagem ou em que aspectos são mais ou menos exigentes. Os dados estatísticos, que serviram de base para a construção dessa tabela, estão no anexo F', na tabela F.7.

Para os itens *adaptabilidade, completude, eficiência de execução, independência de ambiente, precisão, robustez e verificabilidade* os usuários têm o mesmo sentimento que os analistas, em relação à qualidade. Quando o software é, facilmente, adaptável, os analistas dispõem, de uma maneira geral, um menor esforço de codificação, atendendo, em tempo hábil, às solicitações de seus usuários. Se todas as funções especificadas no sistema estão implementadas, isto gera a satisfação do usuário, por ter recebido, integralmente, seu produto de software e, para o analista, por ter concluído aquele projeto. Os computadores estão cada vez mais rápidos no processamento das informações, e vêm suprindo as expectativas com relação à *eficiência de execução* de programas. Os pesos deste atributo variando de baixo a médio, equivale dizer que os usuários e os analistas de sistemas financeiros não mais se inquietam com a *eficiência de execução* do ambiente onde o sistema é processado, salvo quando há pane no computador. No entanto, alguns usuários de microcomputadores ressaltaram a importância deste critério em alguns de seus sistemas. Usuários e desenvolvedores compartilham, também, do mesmo sentimento em relação à precisão e à robustez de seus produtos de software, impelidos por razões já descritas, anteriormente. Ambos concordam que a documentação e a organização da mesma muito auxiliam a verificabilidade do sistema, como é constatado em subseções anteriores. No entanto, eles não consideram seus padrões de documentação satisfatórios, julgando a tarefa de documentar e organizar a documentação enfadonha e postergável.

Para os subfatores *alterabilidade, generalidade, lucratividade, necessidade, oportunidade, segurança e validabilidade* os usuários são mais exigentes que

os analistas de sistema. Na maioria das vezes, as alterações de programas são urgentes e de implementação imediata. Há um tempo máximo, pré-estabelecido, que precisa ser cumprido. O subfator *generalidade* está mais, intimamente, ligado com a independência da quantidade de dados do sistema. Se, de repente, o número de clientes de uma empresa for incrementado consideravelmente, o sistema deve suportar esta demanda de mercado, no tempo oportuno. Zelar pela segurança dos sistemas financeiros é cuidar, literalmente, do próprio patrimônio da empresa. É isto os gerentes financeiros procuram fazer, justificando, portanto, a grande exigência destes com relação à segurança de seus produtos de software. A *validabilidade*, para os usuários, corresponde ao processo de avaliação dos resultados do sistema. Eles, via de regra, não utilizam os resultados de seus produtos antes de estes passarem por seu crivo. Seja, por exemplo, a validação do conteúdo das informações geradas em relatórios ou em telas.

Apenas os subfatores *amenidade de uso* e *configurabilidade* têm menor importância para os usuários, que para os analistas. Os desenvolvedores de software financeiro esforçam-se para atenderem a seus usuários, procurando obter destes um sentimento de satisfação e de confiança no produto. Para isto, são levados em consideração requisitos de qualidade, que facilitem a manipulação do produto e a apresentação de suas informações de maneira compreensível e familiar. Ainda tentando atender, mais rápido e eficientemente, as solicitações de seus usuários, os desenvolvedores, às vezes, dotam seus programas de características próprias do ambiente onde executam, conforme descrito anteriormente.

Os demais subfatores dizem respeito à codificação, à reutilização e à manutenção dos programas, portanto, tarefas próprias dos analistas de sistema. São eles: *aplicabilidade, clareza, concisão, consistência, disponibilidade, eficiência de armazenamento, estilo, estrutura, modularidade e rastreabilidade*. Dentre estes atributos de qualidade a *disponibilidade* tem o peso mais elevado, variando de alto a muito alto. Portanto, é fundamental, que o programa e sua documentação estejam sempre atualizados e prontos para o uso, quando for preciso. É desejável, também, que o mesmo esteja armazenado em meios físicos de fácil apresentação. O subfator *consistência* tem peso variado de médio a muito alto. É recomendável, portanto, que

os programas tenham notação uniforme para seus símbolos, não havendo conflitos entre eles, agilizando assim, sua manutenção.

Os atributos de qualidade *aplicabilidade, clareza, concisão, estrutura e modularidade* possuem grau de importância variando de médio a alto. Embora ainda não haja uma forma sistemática de reutilização de código, os desenvolvedores procuram dotar, intuitivamente, parte do código, de características de reutilização, visando o seu emprego em outras aplicações, na mesma área do domínio do problema. Eles buscam, também, alcançar uma maior homogeneidade em seus programas, tornando-os mais claros e concisos, e de fácil entendimento. Com isto surge a necessidade de se ter as funções do programa implementadas através de módulos funcionais e serem bem projetados. É desejável, também, que os módulos do sistema estejam inseridos em uma organização hierárquica, dentro de uma estrutura balanceada e definida, permitindo aos manipuladores do mesmo, uma rápida compreensão, no nível de detalhe desejado.

Os atributos de qualidade *eficiência de armazenamento, estilo e rastreabilidade* possuem importância mediana. A preocupação com a *eficiência de armazenamento*, conforme observou-se na pesquisa, só se manifesta em sistemas de grande porte. Notou-se, também, que muitos dos sistemas pesquisados ainda são projetados de uma forma unilateral ou seja, eles geram todas os dados dos quais necessitam. Muitas vezes, estas informações já são geradas por outros sistemas, dentro da empresa, embora não estejam no formato ideal para aquela aplicação. No contexto geral, existem muitos dados armazenados de forma redundante, embora várias instituições utilizem sistemas de bancos de dados. A *rastreabilidade* é vista como agilizadora da manutenção, facilitando a localização de cada elemento do programa em suas documentações, e o inter-relacionamento desses elementos, dentro do sistema.

V.3.8 Classificação dos Subfatores obtidos dos Analistas de Sistema

Tabela V.11: Classificação dos Subfatores obtidos dos Analistas de Sistema

Requisitos de Qualidade de Software Financeiro	
Precisão	E
Necessidade	A – E
Segurança	A – MA
Disponibilidade	A – MA
Oportunidade	ME – E
Robustez	A – MA
Consistência	ME – MA
Modularidade	ME – A
Alterabilidade	ME – A
Compleitude	ME – A
Concisão	ME – A
Amenidade de Uso	ME – A
Estrutura	ME – A
Adaptabilidade	ME – A
Clareza	ME – A
Aplicabilidade	ME – A
Eficiência de Armazenamento	ME
Rastreabilidade	ME
Estilo	ME
Validabilidade	ME
Verificabilidade	ME
Generalidade	ME
Configurabilidade	B – ME
Eficiência de Execução	B – ME
Lucratividade	B
Independência do Ambiente	MB – B

Tabela V.12: Classificação dos Subfatores obtidos dos Usuários

Requisitos de Qualidade de Software Financeiro	
Precisão	E
Necessidade	MA – E
Oportunidade	A – E
Segurança	MA
Robustez	A – MA
Alterabilidade	A – MA
Generalidade	ME – MA
Validabilidade	ME – MA
Completude	ME – A
Adaptabilidade	ME – A
Amenidade de Uso	ME
Verificabilidade	ME
Eficiência de Execução	B – ME
Lucratividade	B – ME
Configurabilidade	B
Independência do Ambiente	MB – B

A tabela V.11 apresenta a classificação dos subfatores, em ordem descendente de importância, obtida dos analistas de sistema. Essa tabela fornece uma visão geral das características de qualidade de um software financeiro, no entendimento de analistas de sistema. Os dados estatísticos, que serviram de base para a construção dessa tabela, estão no anexo *F*, na tabela F.8.

Através destes atributos de qualidade, podemos verificar o esforço que deve ser dispendido, ao longo do desenvolvimento e da vida útil de um sistema da área financeira. Essa tabela, confirma-nos, novamente, que a maioria dos requisitos de qualidade de um software financeiro tem pesos elevados. Isto evidencia a enorme responsabilidade dos analistas, no sentido de que seus produtos de software tenham o desempenho esperado. Análises mais detalhadas a respeito desses subfatores podem ser vistas na subseção V.3.7.

V.3.9 Classificação dos Subfatores obtidos dos Usuários

A tabela V.12 apresenta a classificação dos subfatores obtida dos usuários, em ordem descendente de importância. Essa tabela, a exemplo da anterior, fornece-nos os principais atributos de qualidade a serem observados, no desenvolvimento de um

Tabela V.13: Classificação dos Subfatores obtidos do Consolidado

Requisitos de Qualidade de Software Financeiro	
Precisão	E
Necessidade	A – E
Oportunidade	A – E
Segurança	A – MA
Disponibilidade	A – MA
Robustez	A – MA
Alterabilidade	A – MA
Consistência	ME – MA
Modularidade	ME – A
Completude	ME – A
Validabilidade	ME – A
Generalidade	ME – A
Concisão	ME – A
Estrutura	ME – A
Clareza	ME – A
Adaptabilidade	ME – A
Aplicabilidade	ME – A
Amenidade de Uso	ME
Eficiência de Armazenamento	ME
Rastreabilidade	ME
Estilo	ME
Verificabilidade	ME
Configurabilidade	B – ME
Eficiência de Execução	B – ME
Lucratividade	B – ME
Independência do Ambiente	MB – B

software financeiro, na concepção de usuários. Os dados estatísticos, que serviram de base para a construção dessa tabela, estão no anexo *F*, na tabela F.9.

Percebe-se que os usuários da área financeira são mais exigentes, em relação à qualidade de seus produtos de software, que os analistas. Isto se justifica pela importância que tem o software financeiro, na estrutura interna de uma instituição. Análises mais detalhadas a respeito desses subfatores foram feitas na subseção V.3.7.

V.3.10 Classificação dos Subfatores obtidos do Consolidado

A tabela V.13 apresenta a classificação dos subfatores obtida do consolidado, em ordem descendente de importância. Essa tabela, mostra-nos os pontos de vista de analistas e de usuários, formando uma composição dos valores obtidos de ambos, no que diz respeito à qualidade de um software financeiro. Os dados estatísticos, que serviram de base para a construção dessa tabela, estão no anexo *F*, na tabela F.10.

Alguns pontos refletem, basicamente, o sentimento dos usuários; outros, o entendimento dos analistas de sistema. Pode-se perceber, também, como se evidenciou nas subseções anteriores, que a maioria dos requisitos de qualidade dos sistemas financeiros possuem alta importância. Na verdade, para que a qualidade deste produto seja atingida é necessário grande empenho da equipe desenvolvedora, além do apoio efetivo da gerência do departamento de computação da empresa. Mais ainda, os usuários devem estar dispostos e preparados para colaborar, efetivamente, na construção deste produto. Sem um trabalho de equipe e de cooperação mútua, o desenvolvimento de um produto de software de qualidade pode ser penoso e desgastante, e até não vir a realizar-se.

V.3.11 Classificação dos Subfatores/critérios obtidos dos Analistas de Sistema

A tabela V.14 é uma composição das tabelas V.7 e V.11. Ela nos apresenta a classificação dos subfatores/critérios obtidos dos analistas de sistema, em ordem descendente de pesos. Os dados estatísticos, que serviram de base para a construção dessa tabela, estão no anexo *F*, na tabela F.11.

Essa tabela nos dá uma noção completa de todos os atributos de qualidade selecionados para um software financeiro, no entendimento dos analistas de sistema. Considerações mais detalhadas a respeito desses atributos de qualidade podem ser observadas nas subseções V.3.4 e V.3.7.

V.3.12 Classificação dos Subfatores/critérios obtidos dos Usuários

Tabela V.14: Classificação dos subfatores/critérios obtidos dos Analistas de Sistema

Requisitos de Qualidade de Software Financeiro	
PRECISÃO	E
- Acurácia	E
- Correção	E
NECESSIDADE	A - E
- Necessidade funções implementadas	A - E
SEGURANÇA	A - MA
- Confidenciabilidade	A - E
- Vulnerabilidade	A - E
- Detectabilidade de violação	A - E
- Imputabilidade	A - MA
- Privacidade	A - MA
- Sensibilidade	ME - MA
DISPONIBILIDADE	A - MA
- Estar atualizado	A - E
- Acessibilidade	A - MA
OPORTUNIDADE	A - E
- Tempestividade	A - E
ROBUSTEZ	A - MA
- Redundância	A - E
- Confinamento de falhas	A - E
- Detectabilidade de falhas	A - E
- Reparabilidade	A - MA
- Ratificabilidade	A - MA
- Recuperabilidade	ME - MA
- Tolerância entrada dados incorretos	ME - A
CONSISTÊNCIA	ME - MA
- Consistência interna	ME - MA
- Consistência externa	ME - MA
MODULARIDADE	ME - A
- Memória de estado	A - E
- Coesão	ME - MA
- Isolamento	ME - MA
- Acoplamento	ME - MA
- Documentação	ME - A
- Relato de erro	ME - A
- Balanceamento	ME - A
ALTERABILIDADE	ME - A
- Alterabilidade corretiva	A - MA
- Alterabilidade evolutiva	ME - A

Tabela V.14: Continuação

Requisitos de Qualidade de Software Financeiro	
COMPLETUDE	ME – A
– Completude prog. com relação às funções requer. pelo usuário	ME – MA
– Completude prog. com relação às normas elabor. códigos fonte organiz.	ME – A
CONCISÃO	ME – A
– Especificidade	A – MA
– Uso de linguagem concisa	A
– Simplicidade de código	ME
– Complexidade estática do programa	B – ME
AMEDIDADE DE USO	ME – A
– Facilidade de uso	ME – MA
– Estabilidade	ME – A
– Apresentação	ME – A
– Seleção de auxílios	B – A
ESTRUTURA	ME – A
– Ordenamento dos procedimentos	ME – A
– Aderência das estruturas	ME
ADAPTABILIDADE	ME – A
– Composição tecnológica	ME – A
CLAREZA	ME – A
– Uso de terminologia uniforme	ME – A
– Fidelidade padrões codificação	ME – A
– Auto-descritividade	ME – A
– Transparência estrutural	ME
– Domínio das interfaces	B – A
APLICABILIDADE	ME – A
– Identif. dos componentes	ME – A
– Recuper. dos componentes	ME
EFICIÊNCIA DE ARMAZENAMENTO	ME
– Não redundância de dados	ME – A
– Otimiz. do armazenamento	ME
RASTREABILIDADE	ME
– Localização interna	ME – A
– Organização da documentação	ME
– Localização externa	ME
ESTILO	ME
– Significância das identificações	ME – A
– Uso de técn. de prog. estruturada	ME – A
– Uso de comentários	ME
– Endentação	ME

Tabela V.14: Continuação
Requisitos de Qualidade de Software Financeiro

VALIDABILIDADE	ME
– Legibilidade do programa	ME – A
– Localização externa	ME – A
– Localização interna	ME – A
– Efetividade dos comentários	ME
– Simplicidade do código	ME
VERIFICABILIDADE	ME
– Organiz. da documentação	ME
– Documentação	ME
GENERALIDADE	ME
– Independ. quantidade de dados	ME – MA
– Independ. do tipo de dados	B – ME
CONFIGURABILIDADE	B – ME
– Conformação com o ambiente	B – ME
EFICIÊNCIA DE EXECUÇÃO	B – ME
– Tempo de troca de mensagens	B – ME
– Tempo de troca de comandos	MB – B
LUCRATIVIDADE	B
– Taxa de retorno	B
INDEPENDÊNCIA DO AMBIENTE	MB – B
– Independência de software	MB – B
– Independência de hardware	MB – B

Tabela V.15: Classificação dos subfatores/critérios obtidos dos Usuários

Requisitos de Qualidade de Software Financeiro	
PRECISÃO	E
- Acurácia	E
- Correção	E
NECESSIDADE	MA - E
- Necessidade funções implementadas	MA - E
OPORTUNIDADE	A - E
- Tempestividade	A - E
SEGURANÇA	MA
- Vulnerabilidade	MA - E
- Confidenciabilidade	MA - E
- Detectabilidade de violação	A - E
- Privacidade	A - MA
- Imputabilidade	A - E
- Sensibilidade	ME - MA
ROBUSTEZ	A - MA
- Confinamento de falhas	A - E
- Detectabilidade de falhas	A - E
- Reparabilidade	A - E
- Recuperabilidade	A - E
- Redundância	A - E
- Ratificabilidade	ME - MA
- Tolerância entrada dados incorretos	ME - MA
ALTERABILIDADE	A - MA
- Alterabilidade corretiva	A - E
- Alterabilidade evolutiva	ME - MA
GENERALIDADE	ME - MA
- Independ. quantidade de dados	ME - MA
VALIDABILIDADE	ME - MA
COMPLETUDE	ME - A
- Completude prog. com relação às funções requer. pelo usuário	ME - MA
- Completude prog. com relação às normas elabor. códigos fonte organiz.	ME - A
ADAPTABILIDADE	ME - A
- Composição tecnológica	ME - A
AMENIDADE DE USO	ME
- Facilidade de uso	ME - A
- Apresentação	ME
- Seleção de auxílios	ME
VERIFICABILIDADE	ME
- Documentação	ME - A
- Organiz. da documentação	ME

Tabela V.15: Continuação

Requisitos de Qualidade de Software Financeiro	
EFICIÊNCIA DE EXECUÇÃO	B – ME
LUCRATIVIDADE	B – ME
– Taxa de retorno	B – ME
CONFIGURABILIDADE	B
– Conformação com o ambiente	B
INDEPENDÊNCIA DO AMBIENTE	MB – B
– Independência de hardware	MB – B
– Independência de software	MB – B

A tabela V.15 é uma composição das tabelas V.8 e V.12. Ela nos apresenta a classificação dos subfatores/critérios obtidos dos usuários, em ordem descendente de pesos. Os dados estatísticos, que serviram de base para a construção dessa tabela, estão no anexo *F*, na tabela F.12.

Essa tabela nos mostra uma visão completa de todos os atributos de qualidade selecionados para um software financeiro, no ponto de vista dos usuários. Dados mais detalhados a respeito desses atributos de qualidade podem ser observados nas subseções V.3.5 e V.3.7.

V.3.13 Classificação dos Subfatores/critérios obtidos do Consolidado

A tabela V.16 é uma composição das tabelas V.9 e V.13. Ela nos apresenta a classificação dos subfatores/critérios obtidos do consolidado, em ordem descendente de pesos. Os dados estatísticos, que serviram de base para a construção dessa tabela, estão no anexo *F*, na tabela F.13.

Essa tabela nos apresenta o conjunto de todos os requisitos de qualidade selecionados para um software financeiro, na concepção de analistas de sistema e de usuários. Informações minuciosas a respeito desses atributos de qualidade podem ser vistas nas subseções V.3.6 e V.3.7.

Tabela V.16: Classificação dos subfatores/critérios obtidos do Consolidado

Requisitos de Qualidade de Software Financeiro	
PRECISÃO	E
– Acurácia	E
– Correção	E
NECESSIDADE	A – E
– Necessidade funções implementadas	A – E
OPORTUNIDADE	A – E
– Tempestividade	A – E
SEGURANÇA	A – MA
– Confidenciabilidade	A – E
– Vulnerabilidade	A – E
– Detectabilidade de violação	A – E
– Imputabilidade	A – E
– Privacidade	A – MA
– Sensibilidade	ME – MA
DISPONIBILIDADE	A – MA
– Estar atualizado	A – E
– Acessibilidade	A – MA
ROBUSTEZ	A – MA
– Confinamento de falhas	A – E
– Detectabilidade de falhas	A – E
– Redundância	A – E
– Reparabilidade	A – E
– Recuperabilidade	A – MA
– Ratificabilidade	ME – MA
– Tolerância entrada dados incorretos	ME – MA
ALTERABILIDADE	A – MA
– Alterabilidade corretiva	A – MA
– Alterabilidade evolutiva	ME – MA
CONSISTÊNCIA	ME – MA
– Consistência interna	ME – MA
– Consistência externa	ME – MA
MODULARIDADE	ME – A
– Memória de estado	A – E
– Coesão	ME – MA
– Isolamento	ME – MA
– Acoplamento	ME – MA
– Documentação	ME – A
– Relato de erro	ME – A
– Balanceamento	ME – A

Tabela V.16: Continuação

Requisitos de Qualidade de Software Financeiro	
COMPLETEUDE	ME – A
– Completude prog. com relação às funções requer. pelo usuário	ME – MA
– Completude prog. com relação às normas elabor. códigos fonte organiz.	ME – A
VALIDABILIDADE	ME – A
– Legibilidade do programa	ME – A
– Localização externa	ME – A
– Localização interna	ME – A
– Efetividade dos comentários	ME
– Simplicidade do código	ME
GENERALIDADE	ME – A
– Independ. quantidade de dados	ME – MA
– Independ. do tipo de dados	B – ME
CONCISÃO	ME – A
– Especificidade	A – MA
– Uso de linguagem concisa	A
– Simplicidade do código	ME
– Complexidade estática do programa	B – ME
ESTRUTURA	ME – A
– Ordenamento dos procedimentos	ME – A
– Aderência das estruturas	ME
CLAREZA	ME – A
– Uso de terminologia uniforme	ME – A
– Fidelidade padrões codificação	ME – A
– Auto-descritividade	ME – A
– Transparência estrutural	ME
– Domínio das interfaces	B – A
ADAPTABILIDADE	ME – A
– Composição tecnológica	ME – A
APLICABILIDADE	ME – A
– Identif. dos componentes	ME – A
– Recuper. dos componentes	ME
AMENIDADE DE USO	ME
– Facilidade de uso	ME – A
– Estabilidade	ME – A
– Apresentação	ME
– Seleção de auxílios	B – A
EFICIÊNCIA DE ARMAZENAMENTO	ME
– Não redundância de dados	ME – A
– Otimiz. do armazenamento	ME

Tabela V.16: Continuação

Requisitos de Qualidade de Software Financeiro	
RASTREABILIDADE	ME
- Localização interna	ME
- Organização da documentação	ME
- Localização externa	ME
ESTILO	ME
- Significância das identificações	ME - A
- Uso de técn. de prog. estruturada	ME - A
- Uso de comentários	ME
- Endentação	ME
VERIFICABILIDADE	ME
- Documentação	ME
- Organiz. da documentação	ME
CONFIGURABILIDADE	B - ME
- Conformação com o ambiente	B - ME
EFICIÊNCIA DE EXECUÇÃO	B - ME
- Tempo de troca de mensagens	B - ME
- Tempo de troca de comandos	MB - B
LUCRATIVIDADE	B - ME
- Taxa de retorno	B - ME
INDEPENDÊNCIA DO AMBIENTE	MB - B
- Independência de software	MB - B
- Independência de hardware	MB - B

V.4 Avaliação das Hipóteses

No capítulo anterior, foram levantadas hipóteses para serem comprovadas ou refutadas, após a análise estatística dos resultados. Faz-se, aqui, a avaliação dessas hipóteses, fundamentando-se nos resultados obtidos. Todas as hipóteses dizem respeito à proporção populacional. Para se realizar induções sobre uma proporção populacional p , deve-se basear na proporção observada na amostra p' . Sejam, por exemplo, as hipóteses [BANKS 89]:

$$H_0 : p = p_0$$

$$H_1 : p \neq p_0$$

Contanto que $0,10 \leq p \leq 0,90$ e n seja grande o procedimento descrito é aceitável. Satisfeitas estas condições, a distribuição da frequência relativa p' será, aproximadamente, normal. Para se testar $p = p_0$, calcula-se:

$$z_0 = \frac{p' - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \quad (\text{V.1})$$

e rejeita-se H_0 , se $|z_0| > z_\alpha$.

1. Hipótese H_1 : *os atributos de qualidade de um software financeiro pertencem, em sua maioria, a intervalos de pesos elevados.*

- De acordo com a tabela V.9, 67,56 % dos atributos de qualidade selecionados para um software financeiro, pertencem a um intervalo de pesos elevados.

- Considerando-se $p = 0,50$ (limite inferior da *maioria*), a um nível de 5% de significância, através de teste unilateral, constrói-se a hipótese:

$$H_0 : p = 0,50$$

$$H_1 : p > 0,50$$

- O teste estatístico, para $n = 99$, é dado por:

$$z_0 = \frac{0,6756 - 0,50}{\sqrt{\frac{0,50(1-0,50)}{99}}} = 3,49$$

– O valor crítico de $z_{0,05} = 1,645$ [BANKS 89]. Uma vez que $z_0 = 3,49 > 1,645 = z_{0,05}$, H_0 é rejeitada. Logo, ao nível de $\alpha = 5\%$, não se pode rejeitar a hipótese H_1 .

- Pela tabela V.6, 100 % dos requisitos de qualidade selecionados para o objetivo da confiabilidade conceitual pertencem a intervalos de pesos elevados; 70,58 % desses requisitos estão no intervalo de pesos variando de alto a extremamente alto.

– Tomando-se $p = 0,50$, a um nível de 5% de significância, através de teste unilateral, constrói-se a hipótese:

$$H_0 : p = 0,50$$

$$H_1 : p > 0,50$$

– O teste estatístico, para $n = 99$, é dado por:

$$z_0 = \frac{1,00 - 0,50}{\sqrt{\frac{0,50(1-0,50)}{99}}} = 9,94$$

– O valor crítico de $z_{0,05} = 1,645$. Uma vez que $z_0 = 9,94 > 1,645 = z_{0,05}$, H_0 é rejeitada. Logo, ao nível de $\alpha = 5\%$, não se pode rejeitar, novamente, a hipótese H_1 .

2. Hipótese H_2 : *os usuários são mais exigentes que os analistas de sistema, em relação aos requisitos de qualidade de um software financeiro.*

- Analisando-se a tabela F.7 encontra-se que em 62,50 % dos atributos de qualidade, os usuários têm um grau de exigência igual ou superior em relação aos analistas.

– Tomando-se $p = 0,50$, a um nível de 5% de significância, através de teste unilateral, constrói-se a hipótese:

$$H_0 : p = 0,50$$

$$H_2 : p > 0,50$$

– O teste estatístico, para $n = 99$, é dado por:

$$z_0 = \frac{0,625 - 0,50}{\sqrt{\frac{0,50(1-0,50)}{99}}} = 2,48$$

– O valor crítico de $z_{0,05} = 1,645$. Uma vez que $z_0 = 2,48 > 1,645 = z_{0,05}$, H_0 é rejeitada. Logo, ao nível de $\alpha = 5\%$, não se pode rejeitar a hipótese H_2 .

- Os índices percentuais, acima, são os mesmos, quando se analisa as tabelas F.2 e F.3 dos critérios pesquisados para ambas.

3. Hipótese H_3 : *sistemas de um mesmo tipo de aplicação financeira têm características de qualidade semelhantes, tendendo a um perfil de qualidade mais definido, que o perfil de qualidade das aplicações financeiras em geral.*

- Tomou-se dois grupos de aplicações financeiras, pertencentes aos elementos da população, que tivessem o maior número de sistemas pesquisados por aplicação, de acordo com a tabela V.3, gerando-se as tabelas V.17 e V.18.
- Com base nessas tabelas, tomemos σ , como o desvio padrão da população (*consolidado*), representando, aqui, o perfil de qualidade das aplicações financeiras em geral; s , como o desvio padrão da amostra ou seja, cada um dos conjuntos de aplicações financeiras selecionados; μ , como a média da população; \bar{x} , como a média da amostra.
- Consideremos os seguintes resultados, para os dois grupos selecionados das aplicações financeiras:
- tabela V.17 — para os 13 Sistemas de Fundos de Investimento processados, 76,92 % dos s são menores ou iguais a σ .

. Tomando-se $p = 0,50$, a um nível de 5% de significância, através de teste unilateral, constrói-se a hipótese:

$$H_0 : p = 0,50$$

$$H_3 : p > 0,50$$

. O teste estatístico, para $n = 13$, é dado por:

$$z_0 = \frac{0,7692 - 0,50}{\sqrt{\frac{0,50(1-0,50)}{13}}} = 1,94$$

. O valor crítico de $z_{0,05} = 1,645$. Uma vez que $z_0 = 1,94 > 1,645 = z_{0,05}$,

H_0 é rejeitada. Logo, ao nível de $\alpha = 5\%$, não se pode rejeitar a hipótese H_3 .

- tabela V.18 — para os 12 Sistemas de Open Market processados, 80,76 % dos s são menores ou iguais a σ .

. Tomando-se $p = 0,50$, a um nível de 5% de significância, através de teste unilateral, constrói-se a hipótese:

$$H_0 : p = 0,50$$

$$H_3 : p > 0,50$$

. O teste estatístico, para $n = 12$, é dado por:

$$z_0 = \frac{0,8076 - 0,50}{\sqrt{\frac{0,50(1-0,50)}{12}}} = 2,13$$

. O valor crítico de $z_{0,05} = 1,645$. Uma vez que $z_0 = 2,13 > 1,645 = z_{0,05}$, H_0 é rejeitada. Logo, ao nível de $\alpha = 5\%$, não se pode rejeitar a hipótese H_3 .

- Portanto, os sistemas de aplicações financeiras semelhantes tendem a um perfil de qualidade mais definido ou seja e seus atributos de qualidade possuem menores desvios padrões, quando comparados com os mesmos atributos de qualidade selecionados, para as aplicações em geral.

Tabela V.17: Comparação dos valores do desvio padrão, por subfator, dos *Sistemas de Fundos de Investimento* com o *Consolidado*

Requisitos de Qualidade de Software Financeiro					
SUBFATORES	Consolidado		Fundos Investimentos		Comparação <i>s e σ</i>
	μ	σ	\bar{x}	s	
Precisão	5,000	0,000	5,000	0,000	=
Necessidade	4,722	0,514	4,750	0,452	<
Oportunidade	4,433	0,720	4,692	0,480	<
Segurança	4,424	0,370	4,319	0,337	<
Disponibilidade	4,326	0,589	5,000	0,000	<
Robustez	4,267	0,286	4,250	0,414	>
Alterabilidade	3,979	0,530	4,269	0,525	<
Consistência	3,935	0,573	4,750	0,354	<
Modularidade	3,857	0,398	4,071	0,303	<
Completude	3,644	0,494	3,625	0,377	<
Validabilidade	3,621	0,616	3,677	0,507	<
Generalidade	3,613	0,824	4,269	0,881	>
Concisão	3,582	0,337	3,625	0,177	<
Estrutura	3,522	0,547	4,250	0,354	<
Clareza	3,487	0,342	3,000	0,566	>
Adaptabilidade	3,485	0,579	3,462	0,519	<
Aplicabilidade	3,446	0,579	3,250	1,061	>
Amenidade de Uso	3,404	0,441	3,231	0,223	<
Efic. de Armazenamento	3,402	0,533	3,000	0,707	>
Rastreabilidade	3,384	0,383	3,500	0,236	<
Estilo	3,353	0,430	2,875	0,177	<
Verificabilidade	3,211	0,472	3,192	0,384	<
Configurabilidade	2,474	0,708	2,692	0,480	<
Efic. de Execução	2,418	0,762	2,885	0,870	>
Lucratividade	2,237	0,787	2,308	0,480	<
Independ. do Ambiente	1,536	0,511	1,308	0,435	<

Tabela V.18: Comparação dos valores do desvio padrão, por subfator, dos *Sistemas de Open Market* com o *Consolidado*

Requisitos de Qualidade de Software Financeiro					
SUBFATORES	Consolidado		Sist. Open Market		Comparação <i>s e σ</i>
	μ	σ	\bar{x}	s	
Precisão	5,000	0,000	5,000	0,000	=
Necessidade	4,722	0,514	4,750	0,452	<
Oportunidade	4,433	0,720	4,583	0,793	>
Segurança	4,424	0,370	4,319	0,337	<
Disponibilidade	4,326	0,589	4,000	0,577	<
Robustez	4,267	0,286	4,250	0,414	>
Alterabilidade	3,979	0,530	4,042	0,498	<
Consistência	3,935	0,573	3,750	0,289	<
Modularidade	3,857	0,398	3,643	0,412	>
Completude	3,644	0,494	3,625	0,377	<
Validabilidade	3,621	0,616	3,533	0,492	<
Generalidade	3,613	0,824	3,667	0,862	>
Concisão	3,582	0,337	3,250	0,289	<
Estrutura	3,522	0,547	3,250	0,289	<
Clareza	3,487	0,342	3,300	0,115	<
Adaptabilidade	3,485	0,579	3,500	0,522	<
Aplicabilidade	3,446	0,579	3,000	0,577	<
Amenidade de Uso	3,404	0,441	3,181	0,335	<
Efic. de Armazenamento	3,402	0,533	3,000	0,577	>
Rastreabilidade	3,384	0,383	3,333	0,000	<
Estilo	3,353	0,430	3,000	0,000	<
Verificabilidade	3,211	0,472	3,042	0,450	<
Configurabilidade	2,474	0,708	2,500	0,674	<
Efic. de Execução	2,418	0,762	2,500	0,977	>
Lucratividade	2,237	0,787	2,583	0,793	>
Independ. do Ambiente	1,536	0,511	1,250	0,452	<

V.5 Conclusão

Os resultados apresentados, neste capítulo, poderão auxiliar as equipes de desenvolvimento e de manutenção de sistemas da área financeira, no controle da qualidade de seus produtos de software. Esta pesquisa serviu, também, com base nas informações colhidas de analistas de sistema e de usuários, para que fossem identificados os requisitos de qualidade mais importantes de um software financeiro, traçando-se, assim, o perfil de qualidade do mesmo.

Capítulo VI

CONCLUSÃO

Várias empresas japonesas verificaram que a melhoria da qualidade dá lugar, natural e inevitável, a um aumento de produtividade. Isto impregnou-se, no Japão, como um modo de vida e, desde 1950, todos os japoneses vêm buscando um mesmo objetivo: a qualidade. Desta forma, seus custos diminuíram e eles conquistaram mercados [DEMING 90].

Este trabalho teve como objetivo identificar os principais atributos de qualidade de um software financeiro, através dos resultados de uma pesquisa de campo realizada em várias instituições financeiras, traçando-se, assim, o perfil de qualidade deste produto de software. Analistas de sistemas e usuários foram entrevistados, e seus dados foram analisados, separadamente. Foi efetuada, também, nova análise descritiva dos resultados de todos os elementos da população, em conjunto, que se denominou consolidado.

As aplicações financeiras exigem que seus produtos de software possuam alto teor de qualidade. Isto se tornou evidente com os elevados pesos dos atributos de qualidade, obtidos na pesquisa. Esses pesos foram dispostos em intervalos, de acordo com o grau de importância de cada um, com o intuito de facilitar a análise dos resultados. Esta estrutura hierárquica gerada pode auxiliar as equipes de desenvolvimento e de manutenção de sistemas financeiros, no controle da qualidade de seus produtos, objetivando uma maior produtividade dos mesmos. No dizer de [CAVANO 87] o controle da qualidade de software desenvolverá, dentro de uma larga tecnologia de qualidade de software, a mudança de um processo passivo, para

um processo ativo; de descoberta de falhas, para evitação de falhas.

Observou-se que os usuários são mais exigentes que os analistas de sistemas em relação aos requisitos de qualidade dos sistemas financeiros. Eles esperam que seus produtos de software atendam às suas reais necessidades, reproduzindo seus anseios e seus objetivos, no tempo oportuno.

Dentre os atributos de qualidade pesquisados, a *precisão* recebeu de analistas e de usuários o peso máximo da escala estabelecida. Portanto, há um consenso de que um software financeiro, que não seja preciso, é de uso quase que proibitivo, na área de finanças. Notou-se, também, que a *independência do ambiente* recebeu o menor peso. Portanto, um software financeiro bem produzido não depende, em geral, do ambiente, para que opere satisfatoriamente. É óbvio, que um bom *hardware* melhora o desempenho do mesmo, o que é, sempre, desejável.

Notou-se, também, que sistemas de aplicações financeiras semelhantes têm um perfil de qualidade mais definido. Isto significa dizer que os mesmos atributos de qualidade de produtos de software semelhantes possuem menores variações de pesos entre si, em relação a todo o conjunto de aplicações financeiras.

As instituições pesquisadas, de uma maneira geral, foram bastante receptivas, demonstrando interesse no conteúdo dos questionários. Foram bastante proveitosos, os contatos mantidos com analistas de sistemas e com usuários, alguns destes, eram gerentes da área financeira de suas empresas.

O controle da qualidade de software é, sem sombras de dúvidas, o requisito básico e de vital importância, para a especificação, projeto, implementação e operação de sistemas, acompanhando-os em toda a sua vida útil. No entanto, as empresas pesquisadas não usam nenhum método de controle de qualidade de software, embora algumas delas tenham se mostrado receptivas à adoção desses métodos.

Na área de finanças, especificamente, os sistemas financeiros agilizam, de sobremaneira, o processo de tomada de decisão. Com efeito, há um proporcional incremento no significado e na responsabilidade do controle de qualidade destes que,

algumas vezes, representam o papel ativo do gerente financeiro, portanto a própria saúde financeira da empresa.

VI.1 Trabalhos Futuros

Os resultados obtidos, nesta pesquisa, deverão ser utilizados como conhecimento para um *sistema baseado em conhecimento*, cujo objetivo é auxiliar desenvolvedores de sistemas a estabelecerem os requisitos de qualidade de projetos [ROCHA 92].

Apesar de não ter sido encontrado nenhum sistema baseado em conhecimento nas instituições pesquisadas, apoia-se o desenvolvimento destes, para a área financeira, como é o caso do *Screening Financial Innovations*, produzido por pesquisadores da Universidade do Arizona e da UCLA. Um sistema especialista pode fornecer, com base na aquisição de conhecimentos de especialistas da área, vários benefícios de acordo com [RAM 90]:

- *melhorar a qualidade das decisões em relação aos novos produtos, aumentando a taxa de sucesso dos mesmos;*
- *estabelecer uma metodologia, bem definida, para a apresentação de novos produtos;*
- *treinamento de novos gerentes, minimizando a probabilidade de erros dos mesmos;*
- *sensibilizar os gerentes para pontos chaves, que poderiam ser subestimados ou até desprezados;*
- *reter a experiência de especialistas, que, se porventura, deixarem a empresa, não comprometerão o bom andamento e a funcionalidade da equipe de analistas da empresa;*
- *criar um procedimento de avaliação formal, bem organizado, reduzindo o tempo de tomada de decisão.*

As instituições financeiras devem inovar frequentemente, para aumentarem seu grau de competitividade, adequarem-se às preferências dos clientes, às novas regras governamentais, ou aos avanços tecnológicos. Portanto, seus produtos de software devem ser confiáveis, tempestivos e possuírem alto padrão de qualidade.

Referências Bibliográficas

- [ADAM 88] Adam, M. F. et alli; *Towards an observatory aiming at controlling the Software Quality*; Second IEE/BCS Conference; 1988.
- [ANDRADE 91] Andrade, Cacilda Jorge de; *Avaliação da Qualidade de Programas*; Tese de mestrado – COPPE/UFRJ; Rio de Janeiro, abril de 1991.
- [APPLE 87] Apple, Human Interface Guidelines; *The Apple Desktop Interface*; Addison-Wesley Publishing Company Inc.; New York, 1987.
- [ARTHUR 85] Arthur, L. J.; *Measuring Programmer Productivity and Software Quality*; John Wiley & Sons; 1985.
- [BACHE 88] Bache, R et alli; *Software Quality Assurance, a Rigorous Engineering Practice*; John Wiley & Sons; Second IEE/BCS Conference, 1988.
- [BANKS 89] Banks, Jerry; *Principles of Quality Control*; Georgia Institute of Technology; John Wiley & Sons; New York, 1989.
- [BARGUT 86] Bargut, Maurício Fernandes; *Qualidade de “Software” para Microcomputadores*; Tese de Mestrado – IME (Instituto Militar de Engenharia); Rio de Janeiro, fevereiro de 1986.
- [BASILI 87] Basili, V. R. and Rombach, H. D.; *Implementing Quantitative SQA: A Practical Model*; IEEE Software; September, 1987.

- [BEAUFOND 87] Beaufond, C. E. C.; *Verificação e Validação de Software na Fase de Especificação de Requisitos*; Tese de mestrado – COPPE/UFRJ; Rio de Janeiro, fevereiro de 1987.
- [BEAUFOND 89] Beaufond, C. E. C.; *Manual para Controle da Qualidade de Especificações*; Relatório Técnico COPPE/UFRJ – ES-217/87; Rio de Janeiro, outubro de 1989.
- [BERGAMO 91] Bergamo Filho, Valentino; *Gerência econômica da qualidade através do TQC: controle total da qualidade*; Editora Makron, McGraw Hill; São Paulo, 1991.
- [BIGGERSTAFF 87] Biggerstaff, Ted et alii; *Reusability Framework, Assessment, and Directions*; IEEE Software; March 1987.
- [BINDER 90] Binder, L. H. and Poore, J.H.; *Field Experiments With Local Software Quality Metrics*; Software - Practice and Experience; vol 20(7), 631-647; July 1990.
- [BOEHM 78] Boehm, B. W. et alii; *Characteristics of Software Quality*; Amsterdam, North-Holland; 1978.
- [BROWN 87] Brown, B. J.; *Static Analysis and Dynamic Analysis applied to Software Quality Assurance*; Handbook of Software Quality Assurance; Van Nostrand Reinhold Company, 1987.
- [CARD 88] Card, D. N.; *The role of measurement in Software Engineering*; Second IEE/BCS Conference; July 1988.
- [CARD 90] Card, D. N. and Glass, R. L.; *Measuring Software Design Quality*; Prentice-Hall; July 1990.
- [CARDOSO 90] Cardoso, R. N. P.; *Predição, Estimação e Medição da Confiabilidade Durante o Ciclo de Vida do Software*; Tese de mestrado – COPPE/UFRJ; Rio de Janeiro, dezembro de 1990.

- [CAVANO 87] Cavano, Joseph P. et alii; *Quality Assurance in Future Development Environments*; IEEE Software; September 1987.
- [CERINO 86] Cerino, D. A.; *Software Quality Measurement Tools and Techniques*; Proceedings 10th Computer Software and Application Conference (COMSAC), IEEE; 1986.
- [CHERRY 82] Cherry, Richard T.; *Introdução à Administração Financeira*; Editora Atlas S.A. – 2ª edição; tradução de Vera M. C. Nogueira e Danilo A. Nogueira; São Paulo, 1982.
- [CONTE 86] Conte, S. D.; Dunsmore, H. E. and Shen, V. Y.; *Software Engineering Metrics and Models*; The Benjamin/Cummings Pub Company; 1986.
- [COURA 86] Coura, E. C.; *O Estágio de Desenvolvimento do Planejamento em Empresas no Brasil - Uma Metodologia de Estudo*; Tese Mestrado da COPPE/UFRJ; Rio de Janeiro, 1986.
- [CROSBY 90] Crosby, Philip B.; *Qualidade – falando Sério*; Editora McGraw Hill; tradução de José Carlos Barbosa dos Santos; São Paulo, 1990.
- [DEMING 90] Deming, W. Edwards; *Qualidade : a revolução da administração*; Editora Marquês Saraiva S.A.; tradução de Clave Comunicações e Recursos Humanos SC Ltda.; Rio de Janeiro 1990.
- [DOBBINS 87] Dobbins, J. H.; *Software Reliability Management*; Handbook of Software Quality Assurance; New York, Van Nostrand, 1987.
- [ESTEVAM 90] Estevam, R. C.; *Estudos sobre Desenvolvimento de Interfaces: Definição de Técnicas de Classificação e de Avaliação, baseadas na Satisfação do Usuário*; Tese de mestrado – COPPE/UFRJ; Rio de Janeiro, abril 1990.

- [EUROPEAN 72] European Organization for quality control; *Glossary of terms used in quality control.*; Rotterdam, 1972. Pesquisado em [PALADINI 90].
- [EVANS 87] Evans, Michael W. et alii; *Software Quality Assurance & Management*; Wiley-Interscience publications; New York, 1987.
- [EXAME 91] Edição Especial da Revista Exame; *Exame, Melhores e Maiores de 1991*; Editora Abril; São Paulo, 1991.
- [FAIRLEY 85] Fairley, R. E.; *Software Engineering Concepts*; McGraw-Hill; Inc., New York, 1985.
- [FERREIRA 87] Ferreira, R. L.; *A Relação entre a Natureza da Aplicação e os Fatores de Qualidade do Software*; Tese de Mestrado – IME (Instituto Militar de Engenharia); Rio de Janeiro, janeiro de 1987.
- [FLINK 77] Flink, S. J.; Grunewald, D.; *Administração Financeira*; Livros Técnicos e Científicos Editora; São Paulo, 1977.
- [FREITAS 85] Freitas, A. C. C. T.; Bargut, M. F.; Rocha, A. R. C.; *Características de Qualidade de Programas*; Relatório Técnico do Programa de Engenharia de Sistemas e Computação – ES-83/85; novembro de 1985.
- [FREITAS 86] Freitas, A. C. C.; *Software Empresarial: Qualidade ao Longo do Desenvolvimento*; Tese de Mestrado – IME (Instituto Militar de Engenharia); Rio de Janeiro, janeiro de 1986.
- [GARVIN 84] Garvin, David; *What does “product quality” really mean?*; Sloan Management Review; Fall 1984. Pesquisado em [PALADINI 90].
- [GILB 77] Gilb, T.; *Software Metrics*; Winthrop Publishers; Massachusetts, 1977.

- [GILTMAN 87] Gitman, Lawrence J.; *Princípios de Administração Financeira*; Editora Harbra Ltda – 3ª edição; tradução de Jacob Ancelevicz e Francisco José S. Braga; São Paulo, 1987.
- [HALSTEAD 77] Halstead, M.; *Elements of Software Science*; North Holland; 1977.
- [HEHL 86] Hehl, M. E.; *Fortran 77* Editora McGraw Hill; 1986.
- [HORNE 79] Horne, James C. Van; *Política e Administração Financeira, Vols. 1 e 2*; tradução de Alfredo Fileline; Livros Técnicos e Científicos; Rio de Janeiro, 1979.
- [HRADESKY 89] Hradesky, John L.; *Aperfeiçoamento da qualidade e da produtividade: guia prático para a implementação do CEP: controle estatístico de processos*; tradução de Maria Cláudia de Oliveira Santos; McGraw-Hill; São Paulo, 1989.
- [IEEE 87] Draft Version 13b; *Standard for a Software Quality Metrics Methodology*; IEEE Quality Metrics Standard Committee: P1061; July 1987.
- [INCE 88] Ince, D. C. and Sheppard, M. J.; *System Design Metrics a Review and Perspective*; Second IEE/BCS Conference; July 1988.
- [JONES 88] Jones, Gerald; Prieto-Diaz, Ruben; *Building and Managing Software Libraries*; Twelfth Annual International Computer Software and Applications Conference; Chicago, Illinois, October 1988.
- [LONGWORTH 85] Longworth, G.; *Padrões em Programação: Métodos e Procedimentos*; Editora Campus; Rio de Janeiro, 1985.
- [MANNNS 88] Manns, Tom; Coleman, Michael; *Software Quality Assurance*; Macmillan Education LTD; London 1988.

- [MARTIN 85] Martin, J and Mc'Clure; *Metodologias para Análise de Projeto de Sistemas*; Compucenter; 1985.
- [MARTINS 90] Martins, Eliseu e Assaf, A.; *Administração Financeira: as finanças das empresas sob condições inflacionárias*; Editora Atlas, – 6ª edição; São Paulo, 1990.
- [McCABE 76] Mc Cabe, T. J.; *A Complexity Measure*; IEEE Transactions on Software Engineering; vol. SE 2, n^o 4, December, 1976.
- [McCALL 79] Mc Call, J. A.; *An Introduction to Software Quality Metrics*; Software Quality Management; Petrocelli, 1979.
- [MEREY 85] Merey, A.; *Automated Software Quality Assurance*; IEEE Transaction on Software Engineering, vol. SE-11, n^o 9; 1986.
- [MYERS 76] Myers, G. J.; *Software Reliability – Principles and Practices*; John Wiley & Sons; New York, 1976.
- [NASCIMENTO 86] Nascimento, E. M.; Rocha, A. R. C.; *Manual para Avaliação da Qualidade de Programas: subfator modularidade*; Relatório Técnico do Programa de Engenharia de Sistemas e Computação da COPPE / UFRF – ES-112/86; Rio de Janeiro, 1986.
- [NEIGHBORS 89] Neighbors, J. M.; *DRACO: A Method for Engineering Reusable Software Systems*; in Ted J. Biggerstaff, Software Reusability, vol.1; 1989
- [NOVELINO 89] Novelino, Cesare; *Qualidade: uma Questão em foco*; Boletim do CREA-RJ; Rio de Janeiro, abril de 1989.
- [PAGE-JONES 80] Page-Jones, M.; *The Practical Guide to Structured Design Strategies*; Yourdon Press; 1980.
- [PALADINI 90] Paladini, Edson Pacheco; *Controle da Qualidade: uma abordagem abrangente*; Editora Atlas S.A.; São Paulo, 1990.

- [PALERMO 88] Palermo, S.; Rocha, A. R. C.; *A Proposal to Evaluate Program Quality for the Delphi Off-Line Environment*; Relatório Técnico do Programa de Engenharia de Sistemas e Computação da COPPE / UFRJ – ES-180/88; dezembro de 1988.
- [PARANTHAMAN 90] Paranthaman, D.; *Controle da Qualidade*; Editora McGraw Hill; tradução de Flávio Deny Steffen; São Paulo, 1990.
- [PASSOS 91] Passos, Maria Cristina João da Fonseca; *Avaliação da Estrutura Modular de Programas na Fase de Projetos*; Tese de mestrado – COPPE/UFRJ; Rio de Janeiro, maio de 1991.
- [PRESSMAN 87] Pressman, Roger S.; *Software Engineering – A Practitioner's Approach*; McGraw – Hill Book Company; second edition 1987.
- [RADC-TR-85-37] *Specification of Software Quality Attributes*; Final Technical Report RADC-TR-85-37, vols. I, II and III; US Rome Air Development Center/Boeing Aerospace Company; 1985.
- [RAM 90] Ram, Sudha and Ram, Sundaresan; *Screening Financial Innovations: An Expert System Approach*; IEEE Expert; August, 1990.
- [ROCHA 83] Rocha, A. R. C.; *Um Modelo para Avaliação da Qualidade de Especificações*; Tese de Doutorado, PUC-RJ; 1983.
- [ROCHA 87] Rocha, A. R. C.; *Análise e Projeto Estruturado de Sistemas*; Editora Campus; Rio de Janeiro, 1987.
- [ROCHA 90] Rocha, A. R. C.; Passos, M. C. J. F.; *Critérios para Avaliação de Software Pecuária do Leite*; COPPE/UFRJ – Publicações Técnicas – ES-223/90; maio de 1990.
- [ROCHA 92] Rocha, A. R. C.; Passos, M. C. J. F.; Belchior, A. D.; Campos, G.; Fonseca A.; "An Environment for Software Qua-

lity Evaluation”; aceito para “The Ninth International Conference of the Israel Society for Quality Assurance”; Jerusalém, Israel, November 1992.

- [SEI 86] SEI – Secretaria Especial de Informática; *Relatório da Comissão Especial nº 21 / Proteção de dados*; Brasília, 1986.
- [SPSS 84] SPSS/PC.; *SPSS for IBM PC/XT*; SPSS Inc.; 1984, USA.
- [STAA 87] Staa, A. v.; *Engenharia de Programas*; Livros Técnicos e Científicos Editora; 2ª edição; Rio de Janeiro, 1987.
- [STAA 91] Staa, A. v.; *Garantia da Qualidade em Engenharia de Software*; XI Congresso Nacional da SBC; Santos-SP, agosto de 1991.
- [STAHL 88] Stahl, M. M.; *Qualidade de Programas: Avaliação do Estilo de Programação*; Relatório Técnico do Programa de Engenharia de Sistemas e Computação da COPPE / UFRJ — ES-168/88; Rio de Janeiro, setembro de 1988.
- [VICENT 88] Vicent, H. et alli; *Software Quality Assurance*; Vol. 1, Practice and Implementation; Prentice-Hall; 1988.
- [WARNIER 87] Warnier, Jean-Dominique; *LCP – Lógica de Construção de Programas*; Editora Campus; 5ª edição; Tradução de Luiz Paulo Bastos Abrahão; Rio de Janeiro, 1987.
- [WERNECK 90] Werneck, V. M. B.; *Taxonomia de Domínios de Aplicações*; Tese de mestrado – COPPE/UFRJ; Rio de Janeiro, novembro de 1990.

Anexo A

Carta de Recomendação

Prezado Senhor,

Temos a certeza de que V.Sa. compartilha conosco da opinião de que a solução dos atuais problemas do software só será atingida, quando tivermos alcançado um amadurecimento no assunto, formalizando conhecimentos, e melhorando a qualidade dos produtos de software.

Nós, do Programa de Pós-Graduação em Engenharia de Sistemas e Computação da COPPE/UFRJ, acreditamos que, ao promover pesquisas nesta área, colaboramos para a solução a médio e a longo prazo dos problemas do software. No entanto, nada pode ser realizado neste sentido sem o devido conhecimento, análise e documentação da realidade prática. Desta maneira, a COPPE-Sistemas realiza esforços de pesquisas junto às empresas.

Solicitamos o empenho de V.Sa. no sentido de receber o Sr. Arnaldo Dias Belchior, para uma entrevista de cerca de 01 (uma) hora sobre os requisitos de qualidade de sistemas financeiros. As informações individuais fornecidas são confidenciais e serão tratadas estatisticamente de maneira coletiva, e seus resultados serão divulgados de forma conjunta, os quais enviaremos a V.Sa., oportunamente. Agradecemos, antecipadamente, pela atenção e empenho de V.Sa.

Atenciosamente,

Prof. Ana Regina Cavalcanti da Rocha

Coordenadora da Linha de Pesquisa de Engenharia de Software

Anexo B

Questionário dos Analistas de Sistema

COPPE/UFRJ — Programa Pós-Graduação em Engenharia de Sistemas

Linha de Pesquisa: Engenharia de Software

QUESTIONÁRIO DE CONTROLE DE QUALIDADE

(Analistas de Sistema)

1. Preencha as Informações Cadastrais abaixo, de acordo com os dados solicitados.
2. Nos Quadros seguintes, atribua pesos (de 0 a 5) aos Requisitos de Qualidade de acordo com o seu grau de importância para o(s) Sistema(s) Financeiro(s) avaliado(s).
3. Segue uma definição sucinta, em manual anexo, dos subfatores/critérios, que são instrumentos de avaliação para a qualidade de um software financeiro.

INFORMAÇÕES CADASTRAIS

Empresa: _____

Endereço: _____

A empresa usa algum método de controle de qualidade de software? Sim() Não()

Em caso afirmativo no item anterior, sintetize este método. _____

A) Nome do Sistema: _____

Objetivos do Sistema: _____

Porte do Sistema: () Pequeno

() Médio

() Grande

Instalação: () Microcomputador

() Minicomputador

() Mainframe

Observações: _____

B) Nome do Sistema: _____

Objetivos do Sistema: _____

Porte do Sistema: () Pequeno

() Médio

() Grande

Instalação: () Microcomputador

() Minicomputador

() Mainframe

Observações: _____

C) Nome do Sistema: _____

Objetivos do Sistema: _____

Porte do Sistema: Pequeno
 Médio
 Grande

Instalação: Microcomputador
 Minicomputador
 Mainframe

Observações: _____

D) Nome do Sistema: _____

Objetivos do Sistema: _____

Porte do Sistema: Pequeno
 Médio
 Grande

Instalação: Microcomputador
 Minicomputador
 Mainframe

Observações: _____

QUADRO 01 – Analista de Sistema

REQUISITOS DE QUALIDADE DE SOFTWARE FINANCEIRO

CONFIABILIDADE DA REPRESENTAÇÃO

SUBFATORES	CRITÉRIOS	PESOS			
		A	B	C	D
CLAREZA	Fidelidade aos padrões de codificação				
	Uso de terminologia uniforme				
	Auto-descritividade				
	Domínio das interfaces				
	Transparência estrutural				
CONCISÃO	Uso de linguagem concisa				
	Complexidade estática do programa				
	Simplicidade do código				
	Especificidade				
ESTILO	Uso de comentários				
	Endentação				
	Uso de técnicas de programação estruturada				
	Significância das identificações				
MODULARIDADE	Coesão				
	Acoplamento				
	Balanceamento				
	Isolamento				
	Memória de estado				
	Relato de erro				
	Documentação				
DISPONIBILIDADE	Acessibilidade				
	Estar atualizado				
ESTRUTURA	Ordenamento dos procedimentos				
	Aderência das estruturas				
RASTREABILIDADE	Localização interna				
	Localização externa				
	Organização da documentação				

QUADRO 02 – Analista de Sistema

REQUISITOS DE QUALIDADE DE SOFTWARE FINANCEIRO

UTILIZABILIDADE

SUBFATORES	CRITÉRIOS	PESOS			
		A	B	C	D
ALTERABILIDADE	Alterabilidade corretiva				
	Alterabilidade evolutiva				
CONSISTÊNCIA	Consistência interna				
	Consistência externa				
OPORTUNIDADE	Tempestividade				
AMENIDADE DE USO	Facilidade de uso				
	Apresentação				
	Seleção de auxílios				
	Estabilidade				
INDEPEND. DO AMBIENTE	Independência de hardware				
	Independência de software				
CONFIGURABILIDADE	Conformação com o ambiente				
GENERALIDADE	Independência do tipo de dados				
	Independência da quantidade de dados				
APLICABILIDADE	Identificação dos componentes				
	Recuperação dos componentes				
ADAPTABILIDADE	Composição tecnológica				
EFICIÊNCIA DE EXECUÇÃO	Tempo de troca de comandos				
	Tempo de troca de mensagens				
EFIC. ARMAZENAMENTO	Otimização do armazenamento				
	Não redundância de dados				
LUCRATIVIDADE	Taxa de retorno				
VERIFICABILIDADE	Documentação				
	Organização da documentação				
VALIDABILIDADE	Simplicidade do código				
	Efetividade dos comentários				
	Legibilidade do programa				
	Localização interna				
	Localização externa				

QUADRO 03 – Analista de Sistema

REQUISITOS DE QUALIDADE DE SOFTWARE FINANCEIRO

CONFIABILIDADE CONCEITUAL

SUBFATORES	CRITÉRIOS	PESOS			
		A	B	C	D
PRECISÃO	Acurácia				
	Correção				
COMPLETUDE	Completo do programa com relação às funções requeridas pelo usuário				
	Completo prog. com relação às normas de elaboração de códigos fonte da organização				
NECESSIDADE	Necessidade das funções implementadas				
ROBUSTEZ	Tolerância à entrada de dados incorretos				
	Ratificabilidade				
	Detectabilidade de falhas				
	Recuperabilidade				
	Redundância				
	Reparabilidade				
	Confinamento de falhas				
SEGURANÇA	Detectabilidade de violação				
	Privacidade				
	Sensibilidade				
	Vulnerabilidade				
	Imputabilidade				
	Confidenciabilidade				

Anexo C

Manual dos Analistas de Sistema

COPPE/UFRJ — Programa Pós-Graduação em Engenharia de Sistemas
Linha de Pesquisa: Engenharia de Software

MANUAL EXPLICATIVO

REQUISITOS DE QUALIDADE DE SOFTWARE FINANCEIRO

(Analistas de Sistema)

CONFIABILIDADE DA REPRESENTAÇÃO

1. CLAREZA: funções codificadas da forma mais clara possível.

- Fidelidade aos padrões de codificação: *o programa está escrito dentro dos padrões de codificação da organização.*
- Uso de terminologia uniforme: *o programa utiliza uma terminologia uniforme, para referenciar suas entidades.*
- Auto-descritividade: *o programa apresenta identificação e comentários dos objetivos e restrições dos módulos, das sub-rotinas e dos procedimentos e possui controle de alterações.*
- Domínios das interfaces: *o programa não tem parâmetros desnecessários, utilizando a quantidade de parâmetros, que especifica em suas funções.*
- Transparência estrutural: *o programa não faz referências a declarações, que não sejam executadas e não possui variáveis de trabalho redundantes*

ou desnecessárias.

2. **CONCISÃO:** funções implementadas com a quantidade mínima de código.
 - *Uso de linguagem concisa: o programa está codificado, na linguagem mais adequada (concisa) à aplicação.*
 - *Complexidade estática do programa: baseia-se no número de operadores e operandos, gerados ou estimados, quando o projeto está completo.*
 - *Simplicidade do código: o programa não possui trechos, que nunca serão executados, gerando dificuldades no entendimento do código.*
 - *Especificidade: o programa não possui alguma condição, que o leve a estados indeterminados.*

3. **ESTILO:** codificação dos programas com recursos, que facilitem a sua compreensão (comentários, mneumônicos, etc.).
 - *Uso de comentários: comentários explicativos são inseridos no programa, para explicar aquilo que não ficou óbvio apenas com a leitura do código.*
 - *Endentação: refere-se ao formato de codificação, que facilita, esteticamente, o entendimento da estrutura lógica do mesmo.*
 - *Uso de técnicas de programação estruturada: o programa não fere as regras da programação estruturada.*
 - *Significância das identificações: os nomes dados às constantes, às funções, às variáveis, aos procedimentos e às sub-rotinas, traduzem o seu significado facilmente.*

4. **MODULARIDADE:** implementação de funções através de módulos, altamente, independentes.
 - *Coesão: é a força, que mantém unidos os elementos de um módulo. O objetivo desejado é o mais alto grau de coesão entre os módulos.*
 - *Acoplamento: é o grau de interdependência entre os módulos. O baixo acoplamento entre módulos indica um sistema bem particionado.*

- *Balanceamento: os módulos de nível mais elevado do programa tratam os dados de uma forma lógica, independentemente das características físicas.*
- *Isolamento: o que o módulo se propõe a fazer é identificado, sem necessidade de se conhecer sua estrutura interna.*
- *Memória de estado: um módulo executa sempre suas funções, como se fosse a primeira vez, independentemente de quando ou quantas vezes seja processado.*
- *Relato de erro: o módulo identifica e relata os erros.*
- *Documentação: o módulo possui documentação disponível e de boa qualidade.*

5. **DISPONIBILIDADE:** documentação sempre atualizada e pronta para o uso.

- *Acessibilidade: o código fonte e a documentação são acessáveis com facilidade.*
- *Estar atualizado: o programa fonte contém as últimas alterações realizadas.*

6. **ESTRUTURA:** forma padronizada de composição das partes do programa.

- *Ordenamento dos procedimentos: os procedimentos de entrada, de saída e de processamento lógico são, facilmente, identificados no programa.*
- *Aderência das estruturas lógicas: as estruturas na linguagem de implementação correspondem às estruturas lógicas do problema.*

7. **RASTREABILIDADE:** caminhar através do programa e de sua documentação.

- *Localização interna: existem facilidades, que permitem localizar, com facilidade, os componentes internos de um programa.*
- *Localização externa: existem facilidades que permitem localizar, com facilidade, elementos ausentes dos módulos.*
- *Organização da documentação: há uma forma sistemática de organização dos fontes dos programas e de suas documentações.*

UTILIZABILIDADE

1. ALTERABILIDADE: facilidade à incorporação de modificações (correções ou melhorias).
 - Alterabilidade corretiva: *o programa encontra-se implementado de maneira a facilitar possíveis correções de erro ou alterações funcionais.*
 - Alterabilidade evolutiva: *o programa encontra-se implementado de maneira a facilitar possíveis melhorias e evoluções.*

2. CONSISTÊNCIA: notação uniforme para os símbolos utilizados no programa.
 - Consistência interna: *não há conflito entre os nomes de referências, variáveis, etc. dentro de um mesmo módulo.*
 - Consistência externa: *o programa não contém conflitos na maneira de implementar os módulos, os protocolos utilizados para as chamadas de função e os procedimentos padronizados.*

3. OPORTUNIDADE: os resultados do programa atendem ao usuário em tempo hábil.
 - Tempestividade: *os resultados do programa atendem, no tempo oportuno, ao usuário.*

4. AMENIDADE DE USO: o programa apresenta os resultados esperados, oferecendo sentimento de satisfação e de confiança ao usuário.
 - Facilidade de uso: *o programa possui requisitos, que facilitem sua manipulação pelo usuário.*
 - Apresentação: *o programa possui interfaces gráficas ou outras características, que facilitem a apresentação das informações.*
 - Seleção de auxílios: *o programa possui meios de apresentar seus procedimentos e características, no instante em que o usuário desejar.*
 - Estabilidade: *o programa é confortável, compreensível e familiar, durante todo o tempo de execução.*

5. INDEPENDÊNCIA DE AMBIENTE: característica de um programa poder ser executado em diferentes ambientes de hardware e software.
- Independência de hardware: *o programa pode ser executado em diferentes ambientes de hardware.*
 - Independência de software: *o programa pode ser executado em diferentes ambientes de software.*
6. CONFIGURABILIDADE: facilidade de alteração das características do programa dentro de um mesmo ambiente.
- Conformação com o ambiente: *o programa faz uso de facilidades do ambiente onde executa, não comprometendo sua independência.*
7. GENERALIDADE: os módulos do programa apresentam o mínimo de restrições de tipos e de quantidade de dados.
- Independência do tipo de dados: *o programa pode operar com os diversos tipos de dados da linguagem utilizada.*
 - Independência da quantidade de dados: *o programa não tem limitações, para a utilização de qualquer volume de dados.*
8. APLICABILIDADE: facilidade de se identificar e acessar as representações do código fonte e suas respectivas documentações.
- Identificação dos componentes: *as estruturas de código têm nomes padronizados, visando a sua reutilização.*
 - Recuperação dos componentes: *o código é armazenado de acordo com um esquema de classificação padronizado.*
9. ADAPTABILIDADE: o programa pode ser facilmente reusado, sem alterações ou modificações significativas, no código fonte.
- Composição tecnológica: *o código fonte pode ser facilmente reusado, sem alterações ou modificações significativas.*

10. EFICIÊNCIA DE EXECUÇÃO: execução das funções do programa dentro do menor tempo possível.

- Tempo de troca de comandos: *o programa executa troca de comandos com rapidez.*
- Tempo de troca de mensagens: *o tempo gasto na troca de mensagens é baixo.*

11. EFICIÊNCIA DE ARMAZENAMENTO: utilização adequada da memória disponível.

- Otimização do armazenamento: *o programa utiliza características de otimização de armazenamento, disponíveis na linguagem de programação.*
- Não redundância de dados: *ausência da redundância de dados.*

12. LUCRATIVIDADE: relação custo/benefício aceitável.

- Taxa de retorno: *a taxa de retorno da utilização do programa é superior ao investimento feito em sua confecção ou em sua aquisição.*

13. VERIFICABILIDADE: o programa pode ser, facilmente, avaliado com relação a sua forma de apresentação.

- Documentação: *o programa possui documentação disponível e de boa qualidade.*
- Organização da documentação: *há uma forma sistemática de organização dos códigos fonte e de suas documentações.*

14. VALIDABILIDADE: o programa pode ser facilmente avaliado com relação ao que foi projetado.

- Simplicidade do código: *o programa utiliza uma quantidade mínima necessária de comandos e de variáveis, para realizar determinada função.*
- Efetividade dos comentários: *existem comentários simples, claros e concisos, auto-documentando, efetivamente, o programa.*

- Legibilidade do programa: *o programa não possui identificadores, que dificultem seu entendimento.*
- Localização interna: *existem facilidades, dentro do programa fonte, que permitem localizar, com destreza, seus componentes.*
- Localização externa: *existem mecanismos, que permitem localizar, com facilidade, elementos ausentes nos módulos.*

CONFIABILIDADE CONCEITURAL

1. PRECISÃO: precisão nos resultados do programa.

- Acurácia: *os métodos numéricos obtidos são coerentes com o que é esperado pelo usuário.*
- Correção: *os resultados obtidos correspondem ao esperado pelo usuário.*

2. COMPLETEZ: implementação de todas as funções especificadas.

- Completez do programa com relação às funções requeridas pelo usuário: *todas as funções pretendidas pelo usuário estão implementadas no software.*
- Completez do programa com relação às normas de elaboração de códigos fonte da organização: *o programa foi codificado de acordo com as normas de elaboração de códigos fonte da organização.*

3. NECESSIDADE: implementação somente das funções necessárias, para o alcance dos objetivos do projeto.

- Necessidade das funções implementadas: *as funções imprescindíveis ao alcance dos objetivos do projeto estão implementadas.*

4. ROBUSTEZ: o programa é capaz de enfrentar situações hostis.

- Tolerância a entrada de dados incorretos: *o programa é capaz de tolerar um certo grau de variação, na entrada de dados, sem mau funcionamento ou rejeição.*

- *Ratificabilidade: é permitido a realimentação, no programa, pelo operador dos dados de entrada, para que possam ser validados posteriormente.*
 - *Detectabilidade de falhas: o programa é capaz de detectar desvios de comportamento esperado para o software, hardware e operador.*
 - *Recuperabilidade: o programa possui mecanismos próprios de correção de erros.*
 - *Redundância: existem cópias dos módulos considerados mais críticos.*
 - *Reparabilidade: existem mecanismos de reparação do software em tempo hábil.*
 - *Confinamento de falhas: os erros que, porventura, ocorram em uma determinada região do programa, não migram para outras regiões*
5. **SEGURANÇA:** existem mecanismos, que evitem falhas, e que possam ter alto custo de reparação.
- *Detectabilidade de violação: o programa possui mecanismos, que evitem danos causados por acessos indevidos.*
 - *Privacidade: há mecanismos de proteção e de controle das operações, realizadas por um determinado indivíduo.*
 - *Sensibilidade: existem indicações da importância dos dados processados.*
 - *Vulnerabilidade: existem mecanismos, que tornem o programa seguro a determinados tipos de ataque.*
 - *Imputabilidade: existem mecanismos, que responsabilizem uma entidade ou um indivíduo, por suas ações no manuseio do software.*
 - *Confidenciabilidade: existem mecanismos, que garantam a proteção dos dados contra acessos não autorizados.*

Anexo D

Questionário dos Usuários

COPPE/UFRJ — Programa Pós-Graduação em Engenharia de Sistemas

Linha de Pesquisa: Engenharia de Software

QUESTIONÁRIO DE CONTROLE DE QUALIDADE

(Usuários)

1. Preencha as Informações Cadastrais abaixo, de acordo com os dados solicitados.
2. Nos Quadros seguintes, atribua pesos (de 0 a 5) aos Requisitos de Qualidade de acordo com o seu grau de importância para o(s) Sistema(s) Financeiro(s) avaliado(s).
3. Segue uma definição sucinta, em manual anexo, dos subfatores/critérios, que são instrumentos de avaliação para a qualidade de um software financeiro.

INFORMAÇÕES CADASTRAIS

Empresa: _____

Endereço: _____

A empresa usa algum método de controle de qualidade de software? Sim() Não()

Em caso afirmativo no item anterior, sintetize este método. _____

A) Nome do Sistema: _____

Objetivos do Sistema: _____

Porte do Sistema: () Pequeno

() Médio

() Grande

Instalação: () Microcomputador

() Minicomputador

() Mainframe

Observações: _____

B) Nome do Sistema: _____

Objetivos do Sistema: _____

Porte do Sistema: () Pequeno

() Médio

() Grande

Instalação: () Microcomputador

() Minicomputador

() Mainframe

Observações: _____

C) Nome do Sistema: _____

Objetivos do Sistema: _____

Porte do Sistema: Pequeno

Médio

Grande

Intalação: Microcomputador

Minicomputador

Mainframe

Observações: _____

D) Nome do Sistema: _____

Objetivos do Sistema: _____

Porte do Sistema: Pequeno

Médio

Grande

Intalação: Microcomputador

Minicomputador

Mainframe

Observações: _____

QUADRO 04 – Usuários

REQUISITOS DE QUALIDADE DE SOFTWARE FINANCEIRO

CONFIABILIDADE CONCEITUAL

SUBFACTORES	CRITÉRIOS	P E S O S			
		A	B	C	D
PRECISÃO	Acurácia				
	Correção				
COMPLETEUDE	Compleitude do programa com relação às funções requeridas pelo usuário				
	Compleitude prog. com relação às normas de elaboração de códigos fonte da organização				
NECESSIDADE	Necessidade das funções implementadas				
ROBUSTEZ	Tolerância à entrada de dados incorretos				
	Ratificabilidade				
	Detectabilidade de falhas				
	Recuperabilidade				
	Redundância				
	Reparabilidade				
	Confinamento de falhas				
SEGURANÇA	Detectabilidade de violação				
	Privacidade				
	Sensibilidade				
	Vulnerabilidade				
	Imputabilidade				
	Confidenciabilidade				

QUADRO 05 – Usuário**REQUISITOS DE QUALIDADE DE SOFTWARE FINANCEIRO****UTILIZABILIDADE**

SUBFATORES	CRITÉRIOS	P E S O S			
		A	B	C	D
ALTERABILIDADE	Alterabilidade corretiva				
	Alterabilidade evolutiva				
OPORTUNIDADE	Tempestividade				
AMENIDADE DE USO	Facilidade de uso				
	Apresentação				
	Seleção de auxílios				
INDEPEND. DO AMBIENTE	Independência de hardware				
	Independência de software				
CONFIGURABILIDADE	Conformação com o ambiente				
GENERALIDADE	Independência da quantidade de dados				
ADAPTABILIDADE	Composição tecnológica				
EFICIÊNCIA DE EXECUÇÃO					
LUCRATIVIDADE	Taxa de retorno				
VERIFICABILIDADE	Documentação				
	Organização da documentação				
VALIDABILIDADE					

Anexo E

Manual dos Usuários

COPPE/UFRJ — Programa Pós-Graduação em Engenharia de Sistemas
Linha de Pesquisa: Engenharia de Software

MANUAL EXPLICATIVO

REQUISITOS DE QUALIDADE DE SOFTWARE FINANCEIRO

(Usuários)

CONFIABILIDADE CONCEITURAL

1. PRECISÃO: precisão nos resultados do programa.

- *Acurácia: os métodos numéricos obtidos são coerentes com o que é esperado pelo usuário.*
- *Correção: os resultados obtidos correspondem ao esperado pelo usuário.*

2. COMPLETEDE: implementação de todas as funções especificadas.

- *Completeude do programa com relação às funções requeridas pelo usuário: todas as funções pretendidas pelo usuário estão implementadas no software.*
- *Completeude do programa com relação às normas de elaboração de códigos fonte da organização: o programa foi codificado de acordo com as normas de elaboração de códigos fonte da organização.*

3. **NECESSIDADE:** implementação somente das funções necessárias, para o alcance dos objetivos do projeto.

- Necessidade das funções implementadas: *as funções imprescindíveis ao alcance dos objetivos do projeto estão implementadas.*

4. **ROBUSTEZ:** o programa é capaz de enfrentar situações hostis.

- Tolerância a entrada de dados incorretos: *o programa é capaz de tolerar um grau de variação na entrada de dados, sem mau funcionamento ou rejeição.*
- Ratificabilidade: *é permitido a realimentação, no programa, pelo operador, dos dados de entrada, para que possam ser validados posteriormente.*
- Detectabilidade de falhas: *o programa é capaz de detectar desvios de comportamento esperado para o software, hardware e operador.*
- Recuperabilidade: *o programa possui mecanismos próprios de correção de erros.*
- Redundância: *existem cópias dos módulos considerados mais críticos.*
- Reparabilidade: *existem mecanismos de reparação do software em tempo hábil.*
- Confinamento de falhas: *os erros que, porventura, ocorram em uma determinada região do programa, não migram para outras regiões*

5. **SEGURANÇA:** existem mecanismos, que evitem falhas, e que possam ter alto custo de reparação.

- Detectabilidade de violação: *o programa possui mecanismos, que evitem danos causados por acessos indevidos.*
- Privacidade: *há mecanismos de proteção e de controle das operações, realizadas por um determinado indivíduo.*
- Sensibilidade: *existem indicações da importância dos dados processados.*
- Vulnerabilidade: *existem mecanismos, que tornem o programa seguro a determinados tipos de ataque.*

- *Imputabilidade: existem mecanismos, que responsabilizem uma entidade ou um indivíduo, por suas ações no manuseio do software.*
- *Confidenciabilidade: existem mecanismos, que garantam a proteção dos dados contra acessos não autorizados.*

UTILIZABILIDADE

1. **ALTERABILIDADE:** facilidade à incorporação de modificações (correções ou melhorias).
 - *Alterabilidade corretiva: o programa encontra-se implementado de maneira a facilitar possíveis correções de erro ou alterações funcionais.*
 - *Alterabilidade evolutiva: o programa encontra-se implementado de maneira a facilitar possíveis melhorias e evoluções.*
2. **OPORTUNIDADE:** os resultados do programa atendem ao usuário em tempo hábil.
 - *Tempestividade: os resultados do programa atendem, em tempo oportuno, ao usuário.*
3. **AMENIDADE DE USO:** o programa apresenta os resultados esperados, oferecendo sentimento de satisfação e confiança ao usuário.
 - *Facilidade de uso: o programa possui requisitos, que facilitem sua manipulação pelo usuário.*
 - *Apresentação: o programa possui interfaces gráficas ou outras características, que facilitem a apresentação das informações.*
 - *Seleção de auxílios: o programa possui meios de apresentar seus procedimentos e características, no instante em que o usuário desejar.*
4. **INDEPENDÊNCIA DE AMBIENTE:** característica de um programa poder ser executado em diferentes ambientes de hardware e software.
 - *Independência de hardware: o programa pode ser executado em diferentes ambientes de hardware.*

- Independência de software: *o programa pode ser executado em diferentes ambientes de software.*
5. CONFIGURABILIDADE: facilidade de alteração das características do programa dentro de um mesmo ambiente.
- Conformação com o ambiente: *o programa faz uso de facilidades do ambiente onde executa, não comprometendo sua independência.*
6. GENERALIDADE: os módulos do programa apresentam o mínimo de restrições de tipos e quantidade de dados.
- Independência da quantidade de dados: *o programa não tem limitações, para a utilização de qualquer volume de dados.*
7. ADAPTABILIDADE: o código fonte pode ser facilmente reusado, sem alterações ou modificações significativas.
- Composição tecnológica: *o código fonte pode ser facilmente reusado, sem alterações ou modificações significativas.*
8. EFICIÊNCIA DE EXECUÇÃO: execução das funções do programa dentro do menor tempo possível.
9. LUCRATIVIDADE: relação custo/benefício aceitável.
- Taxa de retorno: *a taxa de retorno da utilização do programa é superior ao investimento feito em sua confecção ou em sua aquisição.*
10. VERIFICABILIDADE: o programa pode ser, facilmente, avaliado com relação a sua forma de apresentação.
- Documentação: *o sistema possui documentação disponível e de boa qualidade.*
 - Organização da documentação: *há uma forma sistemática de organização da documentação do sistema.*
11. VALIDABILIDADE: o programa pode ser facilmente avaliado com relação ao que foi projetado.

Anexo F

Tabelas Descritivas dos Atributos de Qualidade

Tabela F.1: Médias dos critérios do Objetivo
 Confiabilidade da Representação
 Requisitos de Qualidade de Software Financeiro

CRITÉRIOS	Analista Sistema		Usuário		Consolidado	
	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ
Acessibilidade	4,152	0,665	–	–	4,152	0,665
Acoplamento	3,870	0,761	–	–	3,870	0,761
Aderência das estruturas	3,326	0,634	–	–	3,326	0,634
Auto-descritividade	3,609	0,576	–	–	3,609	0,576
Balanceamento	3,522	0,504	–	–	3,522	0,504
Coesão	3,891	0,706	–	–	3,891	0,706
Complexidade estática do programa	2,978	0,714	–	–	2,978	0,714
Documentação	3,717	0,583	–	–	3,717	0,583
Domínio das interfaces	3,152	0,788	–	–	3,152	0,788
Endentação	3,087	0,661	–	–	3,087	0,661
Especificidade	4,152	0,758	–	–	4,152	0,758
Estar atualizado	4,500	0,691	–	–	4,500	0,691
Fidelidade padrões codificação	3,696	0,694	–	–	3,696	0,694
Isolamento	3,870	0,686	–	–	3,870	0,686
Localização externa	3,283	0,544	–	–	3,283	0,544
Localização interna	3,543	0,503	–	–	3,543	0,503
Memória de estado	4,457	0,689	–	–	4,457	0,689
Ordenamento dos procedimentos	3,717	0,654	–	–	3,717	0,654
Organização da documentação	3,326	0,560	–	–	3,326	0,560
Relato de erro	3,674	0,761	–	–	3,674	0,761
Significância das identificações	3,609	0,648	–	–	3,609	0,648
Simplicidade do código	3,261	0,772	–	–	3,261	0,772
Transparência estrutural	3,261	0,491	–	–	3,261	0,491
Uso de comentários	3,261	0,534	–	–	3,261	0,534
Uso de linguagem concisa	3,935	0,388	–	–	3,935	0,388
Uso de técn. de prog. estruturada	3,457	0,622	–	–	3,457	0,622
Uso de terminologia uniforme	3,717	0,620	–	–	3,717	0,620

Tabela F.2: Médias dos critérios do Objetivo Utilizabilidade
Requisitos de Qualidade de Software Financeiro

CRITÉRIOS	Analista Sistema		Usuário		Consolidado	
	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ
Alterabilidade corretiva	4,043	0,469	4,216	0,808	4,134	0,671
Alterabilidade evolutiva	3,630	0,531	4,000	0,692	3,825	0,645
Apresentação	3,413	0,479	3,157	0,543	3,278	0,534
Conformação com o ambiente	2,761	0,638	2,216	0,673	2,474	0,708
Composição tecnológica	3,500	0,586	3,471	0,557	3,485	0,579
Consistência externa	3,870	0,686	–	–	3,870	0,686
Consistência interna	4,000	0,666	–	–	4,000	0,666
Documentação	3,217	0,629	3,353	0,626	3,289	0,628
Efetividade dos comentários	3,239	0,603	–	–	3,239	0,603
Estabilidade	3,739	0,743	–	–	3,739	0,743
Facilidade de uso	3,913	0,508	3,608	0,492	3,753	0,520
Identif. dos componentes	3,522	0,809	–	–	3,522	0,809
Independência de hardware	1,609	0,536	1,412	0,535	1,505	0,542
Independência de software	1,804	0,452	1,353	0,559	1,567	0,557
Independ. do tipo de dados	2,522	0,658	–	–	2,522	0,658
Independ. quantidade de dados	3,804	0,687	4,020	0,860	3,918	0,786
Legibilidade do programa	3,478	0,547	–	–	3,478	0,547
Localização externa	3,413	0,540	–	–	3,413	0,540
Localização interna	3,391	0,492	–	–	3,391	0,492
Não redundância de dados	3,565	0,687	–	–	3,565	0,687
Organiz. da documentação	3,217	0,466	3,059	0,613	3,134	0,552
Otimiz. do armazenamento	3,239	0,603	–	–	3,239	0,603
Recuper. dos componentes	3,370	0,487	–	–	3,370	0,487
Seleção de auxílios	3,239	0,992	2,980	0,787	3,103	0,895
Simplicidade do código	3,239	0,673	–	–	3,239	0,673
Taxa de retorno	2,000	0,632	2,451	0,856	2,237	0,787
Tempestividade	4,326	0,817	4,529	0,611	4,433	0,720
Tempo de troca de comandos	2,000	0,666	–	–	2,000	0,666
Tempo de troca de mensagens	2,413	0,883	–	–	2,413	0,883

Tabela F.3: Médias dos critérios do Objetivo
 Confiabilidade Conceitual
Requisitos de Qualidade de Software Financeiro

CRITÉRIOS	Analista Sistema		Usuário		Consolidado	
	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ
Acurácia	5,000	0,000	5,000	0,000	5,000	0,000
Completo prog. com relação às funções requer. pelo usuário	3,848	0,631	3,824	0,740	3,835	0,687
Completo prog. com relação às normas elabor. códigos fonte organiz.	3,478	0,658	3,431	0,830	3,454	0,750
Confidenciabilidade	4,783	0,466	4,824	0,384	4,753	0,520
Confinamento de falhas	4,370	0,487	4,706	0,460	4,546	0,500
Correção	5,000	0,000	5,000	0,000	5,000	0,000
Detectabilidade de falhas	4,326	0,634	4,588	0,496	4,464	0,578
Detectabilidade de violação	4,478	0,658	4,725	0,568	4,660	0,556
Imputabilidade	4,174	0,708	4,471	0,757	4,330	0,746
Necessidade funções implementadas	4,674	0,598	4,765	0,428	4,722	0,514
Privacidade	4,087	0,550	4,137	0,721	4,113	0,643
Ratificabilidade	4,109	0,604	4,020	0,648	4,062	0,626
Recuperabilidade	3,913	0,550	4,412	0,828	4,175	0,750
Redundância	4,543	0,721	4,353	0,626	4,443	0,749
Reparabilidade	4,130	0,618	4,490	0,758	4,320	0,714
Sensibilidade	3,870	0,581	4,078	0,820	3,979	0,721
Tolerância entrada dados incorretos	3,739	0,612	3,961	0,798	3,856	0,721
Vulnerabilidade	4,565	0,544	4,843	0,367	4,711	0,477

Tabela F.4: Médias ordenadas dos critérios
obtidos dos Analistas de Sistema

CRITÉRIOS	\bar{x}	σ
Acurácia	5,000	0,000
Correção	5,000	0,000
Confidenciabilidade	4,783	0,466
Necessidade funções implementadas	4,674	0,598
Vulnerabilidade	4,565	0,544
Redundância	4,543	0,721
Estar atualizado	4,500	0,691
Detectabilidade de violação	4,478	0,658
Memória de estado	4,457	0,689
Confinamento de falhas	4,370	0,487
Detectabilidade de falhas	4,326	0,634
Tempestividade	4,326	0,817
Imputabilidade	4,174	0,708
Acessibilidade	4,152	0,665
Especificidade	4,152	0,758
Reparabilidade	4,130	0,618
Ratificabilidade	4,109	0,604
Privacidade	4,087	0,550
Alterabilidade corretiva	4,043	0,469
Consistência interna	4,000	0,666
Uso de linguagem concisa	3,935	0,388
Facilidade de uso	3,913	0,508
Recuperabilidade	3,913	0,550
Coesão	3,891	0,706
Sensibilidade	3,870	0,581
Consistência externa	3,870	0,686
Isolamento	3,870	0,686
Acoplamento	3,870	0,761
Completo prog. com relação às funções requer. pelo usuário	3,848	0,631
Independ. quantidade de dados	3,804	0,687
Tolerância entrada dados incorretos	3,739	0,612
Estabilidade	3,739	0,743
Documentação (R)	3,717	0,583
Uso de terminologia uniforme	3,717	0,620
Ordenamento dos procedimentos	3,717	0,654
Fidelidade padrões codificação	3,696	0,694
Relato de erro	3,674	0,761
Alterabilidade evolutiva	3,630	0,531
Auto-descritividade	3,609	0,576
Significância das identificações	3,609	0,648

Tabela F.4: Continuação
Requisitos de Qualidade de Software Financeiro

CRITÉRIOS	\bar{x}	σ
Não redundância de dados	3,565	0,687
Localização interna (R)	3,543	0,503
Balanceamento	3,522	0,504
Identif. dos componentes	3,522	0,809
Composição tecnológica	3,500	0,586
Legibilidade do programa	3,478	0,547
Compleitude prog. com relação às normas elabor. códigos fonte organiz.	3,478	0,658
Uso de técn. de prog. estruturada	3,457	0,622
Apresentação	3,413	0,479
Localização externa (U)	3,413	0,540
Localização interna (U)	3,391	0,492
Recuper. dos componentes	3,370	0,487
Organização da documentação (R)	3,326	0,560
Aderência das estruturas	3,326	0,634
Localização externa (R)	3,283	0,544
Transparência estrutural	3,261	0,491
Uso de comentários	3,261	0,534
Simplicidade de código (R)	3,261	0,772
Otimiz. do armazenamento	3,239	0,603
Efetividade dos comentários	3,239	0,603
Simplicidade do código (U)	3,239	0,673
Seleção de auxílios	3,239	0,992
Organiz. da documentação (U)	3,217	0,466
Documentação (U)	3,217	0,629
Domínio das interfaces	3,152	0,788
Endentação	3,087	0,661
Complexidade estática do programa	2,978	0,714
Conformação com o ambiente	2,761	0,638
Independ. do tipo de dados	2,522	0,658
Tempo de troca de mensagens	2,413	0,883
Taxa de retorno	2,000	0,632
Tempo de troca de comandos	2,000	0,666
Independência de software	1,804	0,452
Independência de hardware	1,609	0,536

Tabela F.5: Médias ordenadas dos critérios
obtidos dos Usuários

Requisitos de Qualidade de Software Financeiro		
CRITÉRIOS	\bar{x}	σ
Acurácia	5,000	0,000
Correção	5,000	0,000
Vulnerabilidade	4,843	0,367
Confidenciabilidade	4,824	0,384
Necessidade funções implementadas	4,765	0,428
Detectabilidade de violação	4,725	0,568
Confinamento de falhas	4,706	0,460
Detectabilidade de falhas	4,588	0,496
Tempestividade	4,529	0,611
Reparabilidade	4,490	0,758
Imputabilidade	4,471	0,757
Recuperabilidade	4,412	0,828
Redundância	4,353	0,626
Alterabilidade corretiva	4,216	0,808
Privacidade	4,137	0,721
Sensibilidade	4,078	0,820
Ratificabilidade	4,020	0,648
Independ. quantidade de dados	4,020	0,860
Alterabilidade evolutiva	4,000	0,692
Tolerância entrada dados incorretos	3,961	0,798
Completo prog. com relação às funções requer. pelo usuário	3,824	0,740
Facilidade de uso	3,608	0,492
Composição tecnológica	3,471	0,557
Completo prog. com relação às normas elabor. códigos fonte organiz.	3,431	0,830
Documentação	3,353	0,626
Apresentação	3,157	0,543
Organiz. da documentação	3,059	0,613
Seleção de auxílios	2,980	0,787
Taxa de retorno	2,451	0,856
Conformação com o ambiente	2,216	0,673
Independência de hardware	1,412	0,535
Independência de software	1,353	0,559

Tabela F.6: Médias ordenadas por critério
obtidos do Consolidado

Requisitos de Qualidade de Software Financeiro		
CRITÉRIOS	\bar{x}	σ
Acurácia	5,000	0,000
Correção	5,000	0,000
Confidenciabilidade	4,753	0,520
Necessidade funções implementadas	4,722	0,514
Vulnerabilidade	4,711	0,477
Detectabilidade de violação	4,660	0,556
Confinamento de falhas	4,546	0,500
Estar atualizado	4,500	0,691
Detectabilidade de falhas	4,464	0,578
Memória de estado	4,457	0,689
Redundância	4,443	0,749
Tempestividade	4,433	0,720
Imputabilidade	4,330	0,746
Reparabilidade	4,320	0,714
Recuperabilidade	4,175	0,750
Acessibilidade	4,152	0,665
Especificidade	4,152	0,758
Alterabilidade corretiva	4,134	0,671
Privacidade	4,113	0,643
Ratificabilidade	4,062	0,626
Consistência interna	4,000	0,666
Sensibilidade	3,979	0,721
Uso de linguagem concisa	3,935	0,388
Independ. quantidade de dados	3,918	0,786
Coesão	3,891	0,706
Consistência externa	3,870	0,686
Isolamento	3,870	0,686
Acoplamento	3,870	0,761
Tolerância entrada dados incorretos	3,856	0,721
Compleitude prog. com relação às funções requer. pelo usuário	3,835	0,687
Alterabilidade evolutiva	3,825	0,645
Facilidade de uso	3,753	0,520
Estabilidade	3,739	0,743
Documentação (R)	3,717	0,583
Uso de terminologia uniforme	3,717	0,620
Ordenamento dos procedimentos	3,717	0,654
Fidelidade padrões codificação	3,696	0,694
Relato de erro	3,674	0,761
Significância das identificações	3,609	0,648
Auto-descritividade	3,609	0,576

Tabela F.6: Continuação
Requisitos de Qualidade de Software Financeiro

CRITÉRIOS	\bar{x}	σ
Não redundância de dados	3,565	0,687
Localização interna (R)	3,543	0,503
Balanceamento	3,522	0,504
Identif. dos componentes	3,522	0,809
Composição tecnológica	3,485	0,579
Legibilidade do programa	3,478	0,547
Uso de técn. de prog. estruturada	3,457	0,622
Completo prog. com relação às normas elabor. códigos fonte organiz.	3,454	0,750
Localização externa (U)	3,413	0,540
Localização interna (U)	3,391	0,492
Recuper. dos componentes	3,370	0,487
Organização da documentação (R)	3,326	0,560
Aderência das estruturas	3,326	0,634
Documentação (U)	3,289	0,628
Localização externa (R)	3,283	0,544
Apresentação	3,278	0,534
Transparência estrutural	3,261	0,491
Uso de comentários	3,261	0,534
Simplicidade do código (R)	3,261	0,772
Otimiz. do armazenamento	3,239	0,603
Efetividade dos comentários	3,239	0,603
Simplicidade do código (U)	3,239	0,673
Domínio das interfaces	3,152	0,788
Organiz. da documentação (U)	3,134	0,552
Seleção de auxílios	3,103	0,895
Endentação	3,087	0,661
Complexidade estática do programa	2,978	0,714
Independ. do tipo de dados	2,522	0,658
Conformação com o ambiente	2,474	0,708
Tempo de troca de mensagens	2,413	0,883
Taxa de retorno	2,237	0,787
Tempo de troca de comandos	2,000	0,666
Independência de software	1,567	0,557
Independência de hardware	1,505	0,542

Tabela F.7: Médias dos Subfatores
Requisitos de Qualidade de Software Financeiro

SUBFATORES	Analista Sistema		Usuário		Consolidado	
	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ
Adaptabilidade	3,500	0,586	3,471	0,577	3,485	0,579
Alterabilidade	3,837	0,408	4,108	0,594	3,979	0,530
Aplicabilidade	3,446	0,579	–	–	3,446	0,579
Amenidade de Uso	3,576	0,504	3,248	0,304	3,404	0,441
Clareza	3,487	0,342	–	–	3,487	0,342
Compleitude	3,663	0,434	3,627	0,545	3,644	0,494
Concisão	3,582	0,337	–	–	3,582	0,337
Configurabilidade	2,761	0,638	2,216	0,673	2,474	0,708
Consistência	3,935	0,573	–	–	3,935	0,573
Disponibilidade	4,326	0,589	–	–	4,326	0,589
Eficiência de Armazenamento	3,402	0,533	–	–	3,402	0,533
Eficiência de Execução	2,207	0,663	2,608	0,801	2,418	0,762
Estilo	3,353	0,430	–	–	3,353	0,430
Estrutura	3,522	0,547	–	–	3,522	0,547
Generalidade	3,163	0,483	4,020	0,860	3,613	0,824
Independência do Ambiente	1,707	0,441	1,382	0,525	1,536	0,511
Lucratividade	2,000	0,632	2,451	0,856	2,237	0,787
Modularidade	3,857	0,389	–	–	3,857	0,398
Necessidade	4,674	0,598	4,765	0,428	4,722	0,514
Oportunidade	4,326	0,817	4,529	0,611	4,433	0,720
Precisão	5,000	0,000	5,000	0,000	5,000	0,000
Rastreabilidade	3,384	0,336	–	–	3,384	0,383
Robustez	4,161	0,357	4,361	0,384	4,267	0,286
Segurança	4,326	0,346	4,513	0,372	4,424	0,370
Validabilidade	3,352	0,426	3,863	0,664	3,621	0,616
Verificabilidade	3,217	0,417	3,206	0,521	3,211	0,472

Tabela F.8: Médias ordenadas dos Subfatores
obtidos dos Analistas de Sistema

Requisitos de Qualidade de Software Financeiro		
SUBFACTORES	\bar{x}	σ
Precisão	5,000	0,000
Necessidade	4,674	0,598
Segurança	4,326	0,346
Disponibilidade	4,326	0,589
Oportunidade	4,326	0,817
Robustez	4,161	0,357
Consistência	3,935	0,573
Modularidade	3,857	0,389
Alterabilidade	3,837	0,408
Compleitude	3,663	0,434
Concisão	3,582	0,337
Amenidade de Uso	3,576	0,504
Estrutura	3,522	0,547
Adaptabilidade	3,500	0,586
Clareza	3,487	0,342
Aplicabilidade	3,446	0,579
Eficiência de Armazenamento	3,402	0,533
Rastreabilidade	3,384	0,336
Estilo	3,353	0,430
Validabilidade	3,352	0,426
Verificabilidade	3,217	0,417
Generalidade	3,163	0,483
Configurabilidade	2,761	0,638
Eficiência de Execução	2,207	0,663
Lucratividade	2,000	0,632
Independência do Ambiente	1,707	0,441

Tabela F.9: Médias ordenadas dos Subfatores
obtidos dos Usuários

Requisitos de Qualidade de Software Financeiro		
SUBFACTORES	\bar{x}	σ
Precisão	5,000	0,000
Necessidade	4,765	0,428
Oportunidade	4,529	0,611
Segurança	4,513	0,372
Robustez	4,361	0,384
Alterabilidade	4,108	0,594
Generalidade	4,020	0,860
Validabilidade	3,863	0,664
Compleitude	3,627	0,545
Adaptabilidade	3,471	0,577
Amenidade de Uso	3,248	0,304
Verificabilidade	3,206	0,521
Eficiência de Execução	2,608	0,801
Lucratividade	2,451	0,856
Configurabilidade	2,216	0,673
Independência do Ambiente	1,382	0,525

Tabela F.10: Médias ordenadas dos Subfatores
obtidos do Consolidado

Requisitos de Qualidade de Software Financeiro		
SUBFACTORES	\bar{x}	σ
Precisão	5,000	0,000
Necessidade	4,722	0,514
Oportunidade	4,433	0,720
Segurança	4,424	0,370
Disponibilidade	4,326	0,589
Robustez	4,267	0,286
Alterabilidade	3,979	0,530
Consistência	3,935	0,573
Modularidade	3,857	0,398
Compleitude	3,644	0,494
Validabilidade	3,621	0,616
Generalidade	3,613	0,824
Concisão	3,582	0,337
Estrutura	3,522	0,547
Clareza	3,487	0,342
Adaptabilidade	3,485	0,579
Aplicabilidade	3,446	0,579
Amenidade de Uso	3,404	0,441
Eficiência de Armazenamento	3,402	0,533
Rastreabilidade	3,384	0,383
Estilo	3,353	0,430
Verificabilidade	3,211	0,472
Configurabilidade	2,474	0,708
Eficiência de Execução	2,418	0,762
Lucratividade	2,237	0,787
Independência do Ambiente	1,536	0,511

Tabela F.11: Médias ordenadas dos subfatores/critérios obtidos dos Analistas de Sistema

Requisitos de Qualidade de Software Financeiro		
SUBFACTORES / Critérios	\bar{x}	σ
PRECISÃO	5,000	0,000
– Acurácia	5,000	0,000
– Correção	5,000	0,000
NECESSIDADE	4,674	0,598
– Necessidade funções implementadas	4,674	0,598
SEGURANÇA	4,326	0,346
– Confidenciabilidade	4,783	0,466
– Vulnerabilidade	4,565	0,544
– Detectabilidade de violação	4,478	0,658
– Imputabilidade	4,174	0,708
– Privacidade	4,087	0,550
– Sensibilidade	3,870	0,581
DISPONIBILIDADE	4,326	0,589
– Estar atualizado	4,500	0,691
– Acessibilidade	4,152	0,665
OPORTUNIDADE	4,326	0,817
– Tempestividade	4,326	0,817
ROBUSTEZ	4,161	0,357
– Redundância	4,543	0,721
– Confinamento de falhas	4,370	0,487
– Detectabilidade de falhas	4,326	0,634
– Reparabilidade	4,130	0,618
– Ratificabilidade	4,109	0,604
– Recuperabilidade	3,913	0,550
– Tolerância entrada dados incorretos	3,739	0,612
CONSISTÊNCIA	3,935	0,573
– Consistência interna	4,000	0,666
– Consistência externa	3,870	0,686
MODULARIDADE	3,857	0,389
– Memória de estado	4,457	0,689
– Coesão	3,891	0,706
– Isolamento	3,870	0,686
– Acoplamento	3,870	0,761
– Documentação	3,717	0,583
– Relato de erro	3,674	0,761
– Balanceamento	3,522	0,504
ALTERABILIDADE	3,837	0,408
– Alterabilidade corretiva	4,043	0,469
– Alterabilidade evolutiva	3,630	0,531

Tabela F.11: Continuação
Requisitos de Qualidade de Software Financeiro

SUBFATORES / Critérios	\bar{x}	σ
COMPLETUDE	3,663	0,434
– Completude prog. com relação às funções requer. pelo usuário	3,848	0,631
– Completude prog. com relação às normas elabor. códigos fonte organiz.	3,478	0,658
CONCISÃO	3,582	0,337
– Especificidade	4,152	0,758
– Uso de linguagem concisa	3,935	0,388
– Simplicidade de código	3,261	0,772
– Complexidade estática do programa	2,978	0,714
AMEDIDADE DE USO	3,576	0,504
– Facilidade de uso	3,913	0,508
– Estabilidade	3,739	0,743
– Apresentação	3,413	0,479
– Seleção de auxílios	3,239	0,992
ESTRUTURA	3,522	0,547
– Ordenamento dos procedimentos	3,717	0,654
– Aderência das estruturas	3,326	0,634
ADAPTABILIDADE	3,500	0,586
– Composição tecnológica	3,500	0,586
CLAREZA	3,487	0,342
– Uso de terminologia uniforme	3,717	0,620
– Fidelidade padrões codificação	3,696	0,694
– Auto-descritividade	3,609	0,576
– Transparência estrutural	3,261	0,491
– Domínio das interfaces	3,152	0,788
APLICABILIDADE	3,446	0,579
– Identif. dos componentes	3,522	0,809
– Recuper. dos componentes	3,370	0,487
EFICIÊNCIA DE ARMAZENAMENTO	3,402	0,533
– Não redundância de dados	3,565	0,687
– Otimiz. do armazenamento	3,239	0,603
RASTREABILIDADE	3,384	0,336
– Localização interna	3,543	0,503
– Organização da documentação	3,326	0,560
– Localização externa	3,283	0,544
ESTILO	3,353	0,430
– Significância das identificações	3,609	0,648
– Uso de técn. de prog. estruturada	3,457	0,622
– Uso de comentários	3,261	0,534
– Endentação	3,087	0,661

Tabela F.11: Continuação

Requisitos de Qualidade de Software Financeiro		
SUBFATORES / Critérios	\bar{x}	σ
VALIDABILIDADE	3,352	0,426
– Legibilidade do programa	3,478	0,547
– Localização externa	3,413	0,540
– Localização interna	3,391	0,492
– Efetividade dos comentários	3,239	0,603
– Simplicidade do código	3,239	0,673
VERIFICABILIDADE	3,217	0,417
– Organiz. da documentação	3,217	0,466
– Documentação	3,217	0,629
GENERALIDADE	3,163	0,483
– Independ. quantidade de dados	3,804	0,687
– Independ. do tipo de dados	2,522	0,658
CONFIGURABILIDADE	2,761	0,638
– Conformação com o ambiente	2,761	0,638
EFICIÊNCIA DE EXECUÇÃO	2,207	0,663
– Tempo de troca de mensagens	2,413	0,883
– Tempo de troca de comandos	2,000	0,666
LUCRATIVIDADE	2,000	0,632
– Taxa de retorno	2,000	0,632
INDEPENDÊNCIA DO AMBIENTE	1,707	0,441
– Independência de software	1,804	0,452
– Independência de hardware	1,609	0,536

Tabela F.12: Médias ordenadas dos subfatores/critérios obtidos dos Usuários

Requisitos de Qualidade de Software Financeiro		
PRECISÃO	5,000	0,000
– Acurácia	5,000	0,000
– Correção	5,000	0,000
NECESSIDADE	4,765	0,428
– Necessidade funções implementadas	4,765	0,428
OPORTUNIDADE	4,529	0,611
– Tempestividade	4,529	0,611
SEGURANÇA	4,513	0,372
– Vulnerabilidade	4,843	0,367
– Confidenciabilidade	4,824	0,384
– Detectabilidade de violação	4,725	0,568
– Imputabilidade	4,471	0,757
– Privacidade	4,137	0,721
– Sensibilidade	4,078	0,820
ROBUSTEZ	4,361	0,384
– Confinamento de falhas	4,706	0,460
– Detectabilidade de falhas	4,588	0,496
– Reparabilidade	4,490	0,758
– Recuperabilidade	4,412	0,828
– Redundância	4,353	0,626
– Ratificabilidade	4,020	0,648
– Tolerância entrada dados incorretos	3,961	0,798
ALTERABILIDADE	4,108	0,594
– Alterabilidade corretiva	4,216	0,808
– Alterabilidade evolutiva	4,000	0,692
GENERALIDADE	4,020	0,860
– Independ. quantidade de dados	4,020	0,860
VALIDABILIDADE	3,863	0,664
COMPLETUDE	3,627	0,545
– Completude prog. com relação às funções requer. pelo usuário	3,824	0,740
– Completude prog. com relação às normas elabor. códigos fonte organiz.	3,431	0,830
ADAPTABILIDADE	3,471	0,577
– Composição tecnológica	3,471	0,557
AMENIDADE DE USO	3,248	0,304
– Facilidade de uso	3,608	0,492
– Apresentação	3,157	0,543
– Seleção de auxílios	2,980	0,787
VERIFICABILIDADE	3,206	0,521
– Documentação	3,353	0,626
– Organiz. da documentação	3,059	0,613

Tabela F.12: Continuação
Requisitos de Qualidade de Software Financeiro

SUBFACTORES / Critérios	\bar{x}	σ
EFICIÊNCIA DE EXECUÇÃO	2,608	0,801
LUCRATIVIDADE	2,451	0,856
– Taxa de retorno	2,451	0,856
CONFIGURABILIDADE	2,216	0,673
– Conformação com o ambiente	2,216	0,673
INDEPENDÊNCIA DO AMBIENTE	1,382	0,525
– Independência de hardware	1,412	0,535
– Independência de software	1,353	0,559

Tabela F.13: Médias ordenadas dos subfatores/critérios obtidos no Consolidado

Requisitos de Qualidade de Software Financeiro		
SUBFACTORES / Critérios	\bar{x}	σ
PRECISÃO	5,000	0,000
– Acurácia	5,000	0,000
– Correção	5,000	0,000
NECESSIDADE	4,722	0,514
– Necessidade funções implementadas	4,722	0,514
OPORTUNIDADE	4,433	0,720
– Tempestividade	4,433	0,720
SEGURANÇA	4,424	0,370
– Confidenciabilidade	4,753	0,520
– Vulnerabilidade	4,711	0,477
– Detectabilidade de violação	4,660	0,556
– Imputabilidade	4,330	0,746
– Privacidade	4,113	0,643
– Sensibilidade	3,979	0,721
DISPONIBILIDADE	4,326	0,589
– Estar atualizado	4,500	0,691
– Acessibilidade	4,152	0,665
ROBUSTEZ	4,267	0,286
– Confinamento de falhas	4,546	0,500
– Detectabilidade de falhas	4,464	0,578
– Redundância	4,443	0,749
– Reparabilidade	4,320	0,714
– Recuperabilidade	4,175	0,750
– Ratificabilidade	4,062	0,626
– Tolerância entrada dados incorretos	3,856	0,721
ALTERABILIDADE	3,979	0,530
– Alterabilidade corretiva	4,134	0,671
– Alterabilidade evolutiva	3,825	0,645
CONSISTÊNCIA	3,935	0,573
– Consistência interna	4,000	0,666
– Consistência externa	3,870	0,686
MODULARIDADE	3,857	0,398
– Memória de estado	4,457	0,689
– Coesão	3,891	0,706
– Isolamento	3,870	0,686
– Acoplamento	3,870	0,761
– Documentação	3,717	0,583
– Relato de erro	3,674	0,761
– Balanceamento	3,522	0,504

Tabela F.13: Continuação

Requisitos de Qualidade de Software Financeiro		
SUBFACTORES / Critérios	\bar{x}	σ
COMPLETUDE	3,644	0,494
– Completude prog. com relação às funções requer. pelo usuário	3,835	0,687
– Completude prog. com relação às normas elabor. códigos fonte organiz.	3,454	0,750
VALIDABILIDADE	3,621	0,616
– Legibilidade do programa	3,478	0,547
– Localização externa	3,413	0,540
– Localização interna	3,391	0,492
– Efetividade dos comentários	3,239	0,603
– Simplicidade do código	3,239	0,673
GENERALIDADE	3,613	0,824
– Independ. quantidade de dados	3,918	0,786
– Independ. do tipo de dados	2,522	0,658
CONCISÃO	3,582	0,337
– Especificidade	4,152	0,758
– Uso de linguagem concisa	3,935	0,388
– Simplicidade do código	3,261	0,772
– Complexidade estática do programa	2,978	0,714
ESTRUTURA	3,522	0,547
– Ordenamento dos procedimentos	3,717	0,654
– Aderência das estruturas	3,326	0,634
CLAREZA	3,487	0,342
– Uso de terminologia uniforme	3,717	0,620
– Fidelidade padrões codificação	3,696	0,694
– Auto-descritividade	3,609	0,576
– Transparência estrutural	3,261	0,491
– Domínio das interfaces	3,152	0,788
ADAPTABILIDADE	3,485	0,579
– Composição tecnológica	3,485	0,579
APLICABILIDADE	3,446	0,579
– Identif. dos componentes	3,522	0,809
– Recuper. dos componentes	3,370	0,487
AMENIDADE DE USO	3,404	0,441
– Facilidade de uso	3,753	0,520
– Estabilidade	3,739	0,743
– Apresentação	3,278	0,534
– Seleção de auxílios	3,103	0,895
EFICIÊNCIA DE ARMAZENAMENTO	3,402	0,533
– Não redundância de dados	3,565	0,687
– Otimiz. do armazenamento	3,239	0,603

Tabela F.13: Continuação

Requisitos de Qualidade de Software Financeiro		
SUBFACTORES / Critérios	\bar{x}	σ
RASTREABILIDADE	3,384	0,383
– Localização interna	3,543	0,503
– Organização da documentação	3,326	0,560
– Localização externa	3,283	0,544
ESTILO	3,353	0,430
– Significância das identificações	3,609	0,648
– Uso de técn. de prog. estruturada	3,457	0,622
– Uso de comentários	3,261	0,534
– Endentação	3,087	0,661
VERIFICABILIDADE	3,211	0,472
– Documentação	3,289	0,628
– Organiz. da documentação	3,134	0,552
CONFIGURABILIDADE	2,474	0,708
– Conformação com o ambiente	2,474	0,708
EFICIÊNCIA DE EXECUÇÃO	2,418	0,762
– Tempo de troca de mensagens	2,413	0,883
– Tempo de troca de comandos	2,000	0,666
LUCRATIVIDADE	2,237	0,787
– Taxa de retorno	2,237	0,787
INDEPENDÊNCIA DO AMBIENTE	1,536	0,511
– Independência de software	1,567	0,557
– Independência de hardware	1,505	0,542