

VERIFICAÇÃO E VALIDAÇÃO DE SOFTWARE NA FASE DE ESPECIFICAÇÃO  
DE REQUISITOS

Clifton Eduardo Clunie Beaufond  
Universidad Tecnológica de Panamá  
Facultad de Ingenieria en Sistemas Computacionales  
Aptdo. 6-2894 - ElDorado  
Panamá - Rep. de Panamá

Ana Regina Cavalcanti da Rocha  
Universidade Federal do Rio de Janeiro - COPPE  
Caixa Postal 68511  
21945 - Rio de Janeiro - RJ - Brasil

RESUMO

A qualidade de um sistema de software, como a de qualquer produto de alta tecnologia, depende da qualidade de suas especificações. As especificações são a base do desenvolvimento futuro do software e constituem um instrumento de predição e avaliação do comportamento do software a ser construído. A falta de especificações adequadas é uma das principais causas da produção de software de baixa qualidade e com custos de desenvolvimento, operação e manutenção elevados.

Neste artigo, examina-se o problema da qualidade das especificações. A partir de um modelo de avaliação de qualidade de especificações, são identificadas características primitivas e definidos processos de avaliação. Desta forma, uma especificação pode ser avaliada numa forma organizada e disciplinada.

## INTRODUÇÃO

Qualidade é um atributo associado a alguma coisa. Assim sendo, qualidade não pode ser definida universalmente mas deve ser definida para o item em questão. (qualidade de especificações, qualidade de metodologias, qualidade de programas, etc). Qualidade é, também, um conceito multidimensional que se realiza através de um conjunto de atributos ou características.

A experiência têm demonstrado que a qualidade do software é uma responsabilidade de todos os que participam na sua realização ou em seu uso. A qualidade deve ser construída em cada fase do ciclo de desenvolvimento. Não se pode adicionar qualidade em etapas posteriores e por pessoal que não participou do desenvolvimento.

Para que uma organização possa construir software com a qualidade desejada e se possa medir se o software produto possui esta qualidade, é necessário que se defina a qualidade de software de acordo com seu contexto. Para isto deve-se analisar a natureza do produto e identificar-se quais são as características consideradas como desejáveis. Estas características devem ser possíveis de serem avaliadas, existindo um grau mínimo a ser atingido para que um software seja considerado de boa qualidade.

Cabe salientar que 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 expectativas de todos estes usuários. Podemos definir, então, qualidade de software como o conjunto de propriedades a serem satisfeitas em determinado grau, de modo a satisfazer as necessidades de seus usuários.

BOEHM (1) elaborou a curva da proporção entre custos de hardware e software, mostrando que a atividade de manutenção é uma proporção dos custos de software que aumenta progressivamente, tendo em vista o acúmulo de novos softwares produzidos e o aumento do volume de manutenção dos existentes. Mostrou, também, que o custo de correção de um erro na fase de operação de um sistema, isto é, durante a manutenção, é muitas vezes superior

ao custo de correção desse mesmo erro nas fases iniciais do desenvolvimento. Percebe-se claramente a vantagem que advém de detectar e corrigir cedo os erros.

Torna-se evidente, portanto, que o aumento de investimento nas fases iniciais do processo de desenvolvimento pode diminuir consideravelmente os gastos com manutenção, parcela maior do custo total do sistema.

A primeira fase do processo de desenvolvimento é a que, provavelmente, leva a mais erros, dada sua natureza subjetiva. Esta fase, denominada aqui especificação de requisitos, é onde se centra nosso trabalho. É altamente importante e vantajoso investir em esforços humanos e financeiros para gerar especificações de boa qualidade, se possível totalmente isentas de falhas. A qualidade dos requisitos definidos no início do processo de desenvolvimento é fator básico para que possa ser produzido um software de boa qualidade.

### O PROBLEMA DE MEDIÇÕES

A determinação da qualidade é um fator importante no nosso viver diário. Convivemos com isso. A qualidade é avaliada de uma forma quase direta, por comparação de objetos sob idênticas condições, considerando-se atributos pré-determinados. Os objetivos são julgados de acordo com características tais como cor, odor, sabor, beleza, habilidade, etc. Entretanto, este tipo de julgamento é muito subjetivo e para ter algum valor é necessário que seja conduzido por especialistas.

A subjetividade e a experiência também são aplicadas à determinação da qualidade do software. Para ajudar a resolver o problema precisamos de um método para avaliação da qualidade de software através do qual possamos realizar medições quantitativas baseadas em uma análise objetiva.

Medição de software é uma área destinada à consignação de índices quantitativos do software em desenvolvimento, PERLIS (2). Tem como objetivo ajudar a responder às seguintes perguntas que vão surgindo na medida em que se vai progredindo no desenvolvimento:

- é o momento adequado para continuar com a etapa seguinte?
- será necessário retroceder a uma etapa anterior?

As respostas a estas perguntas devem possuir um certo grau de exatidão para que se possa continuar com o controle efetivo do processo de desenvolvimento.

DENICOFF e GRAFTON (3), mostram a necessidade de que a engenharia de software, esteja marcada pela disponibilidade de parâmetros de medidas, precisas e bem entendidas. No entanto medidas para software não podem ser, como nas demais disciplinas de engenharia, baseadas em ciências físicas. Isto torna necessário que a abordagem dada às medidas de software seja marcada pelo paradigma científico tradicional de hipótese, avaliação, crítica e revisão.

O principal objetivo do estudo das medidas do software não é so a determinação estática das propriedades do software, mas, sim uma predição científica durante o ciclo de vida.

Devido a seu potencial preditivo, medidas podem ser utilizadas pelo menos de três formas, CURTIS (4):

- como ferramenta de informação gerencial, ajudando na predição de resultados futuros e consequências na área de custos, produtividade, duração do projeto, etc. Desta forma, oferecem indicadores confiáveis que ajudam a ter boa visão gerencial do projeto;
- como medida da qualidade de software, pois fornecem critérios quantificáveis sob os quais pode-se avaliar o software produto;
- como "feedback" para o pessoal envolvido no projeto.

Entretanto, medidas para serem úteis, precisam estar organizadas. A ciência moderna considera que a complexidade da realidade pode ser entendida pelo estabelecimento e compreensão de modelos hierárquicos, alguns descrevendo comportamentos macroscópicos, ignorando os detalhes e outros descrevendo um comportamento mais microscópico. A ciência trabalha, então, baseada na utilização de modelos, que explicam a relação entre o significado das variáveis sob estudo.

Toda experiência baseada num desenvolvimento científico compre uma sequência de etapas:

- início das observações (medições);
- formulação de hipóteses, baseadas num conjunto de axiomas, para explicar as observações;
- construção do modelo com suas variáveis, baseadas ou estimadas a partir de observações;
- utilização do modelo para fazer previsões das propriedades do sistema;
- execução do experimento, (controlando as observações e o meio ambiente) para determinar a precisão do modelo.

A figura I mostra o ciclo de desenvolvimento de um modelo.

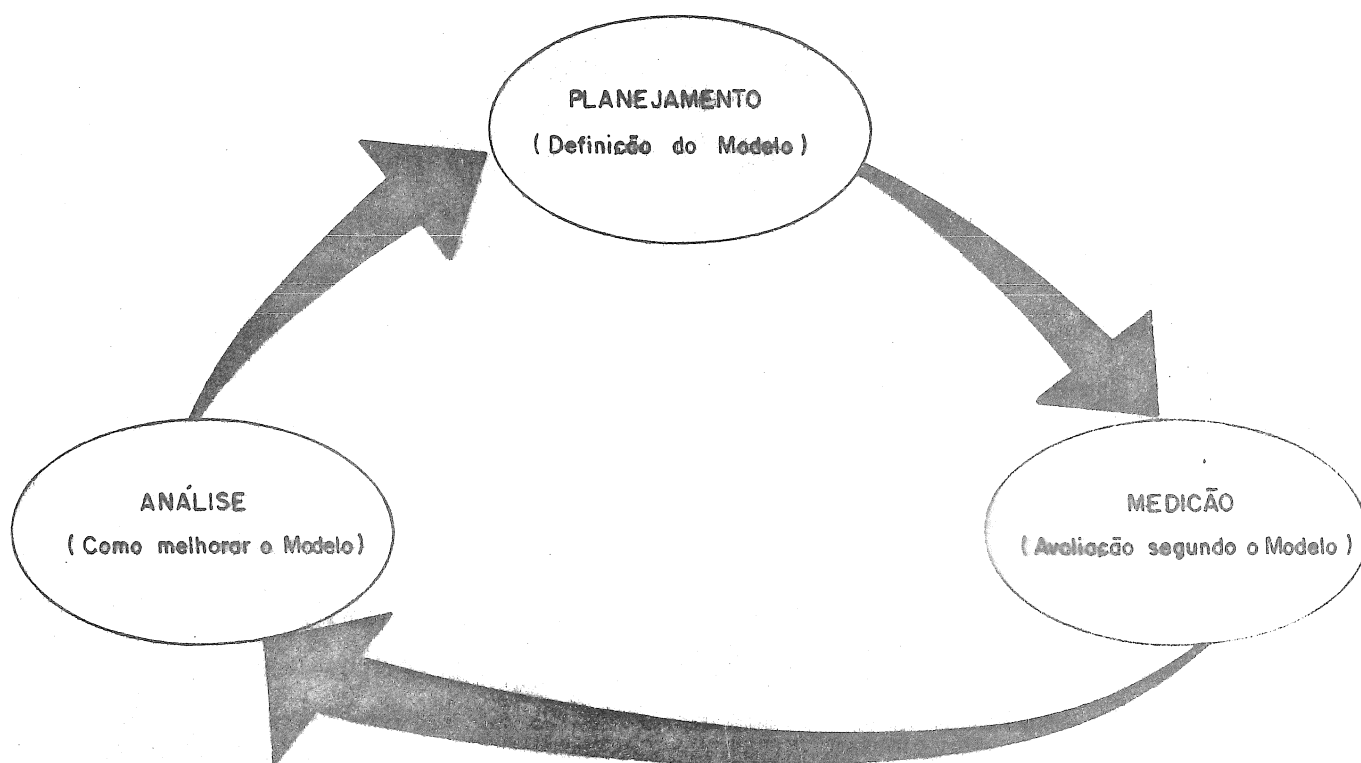


Figura I - Ciclo de Definição de Uso de um Modelo para Avaliação da Qualidade.

## UM MODELO PARA AVALIAÇÃO DA QUALIDADE DE SOFTWARE

Baseada nas considerações anteriores e nos trabalhos de BOEHM (5) e McCALL (6) foi definido um Modelo para Avaliação da Qualidade de Software, baseado nos seguintes conceitos, ROCHA (7)

- Objetivos de qualidade: são propriedades gerais que o produto deve possuir;
- Fatores de qualidade: determinam a qualidade do ponto de vista dos diferentes usuários do produto (usuário final, mantenedores, etc);
- Critérios: são atributos primitivos possíveis de serem avaliados;
- Processos de avaliação: determinam o processo e os instrumentos a serem usados de forma a se medir o grau de presença, no produto, de um determinado atributo;
- Medidas: indicam o grau de presença de um determinado critério;
- Medidas agregadas: indicam o grau de presença de um determinado fator.

A figura II mostra a estrutura do Modelo.

Os objetivos de qualidade são atingidos através dos fatores de qualidade, que podem ser compostos por outros fatores e são avaliados através de critérios. Os critérios definem atributos de qualidade para os fatores.

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

Objetivos e fatores não são diretamente mensuráveis e só podem ser avaliados através de critérios. Um critério é um atributo primitivo, isto é, um atributo independente de todos os outros atributos. Nenhum critério isolado é uma descrição completa de um determinado fator ou sub-fator. Da mesma forma, nenhum fator define completamente um objetivo.

Numa primeira etapa esse modelo foi usado para definir objetivos, fatores e sub-fatores para os diferentes produtos gerados ao longo do desenvolvimento, ROCHA (9), ROCHA (10). Agora este modelo foi refinado definindo-se critérios e processos de avaliação para cada sub-fator.



Figura II - Modelo para Avaliação da Qualidade de Software.  
Fonte: ROCHA (8).

## VERIFICAÇÃO E VALIDAÇÃO DE ESPECIFICAÇÕES

Ao desenvolver um software deve estar presente sempre a preocupação com a qualidade ao longo do processo de desenvolvimento. Portanto, é necessário que este processo de desenvolvimento se dê de forma segura e confiável. Para que isto aconteça, é necessário que os produtos intermediários (especificações, plano, projeto, etc) sejam avaliados quanto à sua qualidade, por esta implicar diretamente na qualidade do produto final. Isto evidencia a necessidade de duas atividades: verificação e validação.

BOEHM (11) define os dois conceitos, verificação e validação, de uma forma simples:

- por verificação entende-se "estamos construindo certo o produto"?;
- por validação entende-se "estamos construindo o produto certo"?

Podemos então afirmar que:

- ao verificarmos um produto, estamos preocupados com sua qualidade intrínseca e com a qualidade de seu processo de construção, e,
- ao validarmos um produto, estamos preocupados com seu conteúdo.

No que se refere a especificações, foram identificados: três objetivos de qualidade: confiabilidade da representação, confiabilidade conceitual e utilizabilidade, portanto:

- ao avaliarmos a confiabilidade da representação, estamos realizando uma verificação na especificação e;
- ao avaliarmos a confiabilidade conceitual estamos validando a especificação.

Para que seja possível avaliar se a especificação reflete o que usuário deseja, é necessário que ele seja capaz de ler e compreender a especificação. Além disso, as especificações são documentos lidos e manipulados por diversas pessoas não necessariamente conhecidas entre si. Isso evidencia a importância da confiabilidade da representação.

O objetivo confiabilidade da representação é atingido através dos seguintes fatores de qualidade:

comunicabilidade: refere-se à capacidade de comunicação da especificação;

manipulabilidade: refere-se à facilidade de manipulação da especificação no seu uso, modificação, etc.

O fator comunicabilidade é atingido através dos seguintes sub-fatores de qualidade:

correção no uso da linguagem de especificação: é a característica de uma especificação de requisitos estar correta do ponto de vista do uso da linguagem de especificação;

concisão: é a característica de uma especificação de requisitos estar escrita com um mínimo de volume de texto, isto é, com uma maximização da quantidade de informações por unidade de texto;

uniformidade de terminologia: é a característica de uma especificação de requisitos estar escrita com uniformidade de terminologia e simbologia;

uniformidade de abstração: é a característica de uma especificação de requisitos estar escrita com o mesmo grau de detalhamento nos diversos níveis de abstração;

modularidade: é a característica de uma especificação de requisitos estar dividida em módulos, permitindo um perfeito entendimento de seus diversos aspectos de forma exclusiva.

O fator manipulabilidade é atingido através dos seguintes sub-fatores de qualidade:

disponibilidade: é a característica de uma especificação de requisitos estar disponível para o acesso de seus usuários, na versão mais atualizada.

estrutura: é a característica de uma especificação de requisitos estar organizada em seus documentos de forma a facilitar ao leitor a encontrar a visão que deseja;

rastreabilidade: é a característica de uma especificação de requisitos permitir o acompanhamento de uma sequência de agregação de detalhes relacionados a um determinado aspecto ou assunto.

O objetivo fundamental é construir um produto de acordo com as necessidades e expectativas do usuário. Isto evidencia a importância da confiabilidade conceitual. O objetivo confiabilidade conceitual é atingido através dos seguintes fatores de qualidade:

fidedignidade: refere-se à confiança de que a especificação representa, o que é entendido como sendo as necessidades e expectativas de seus usuários.

suficiência: refere-se à completeza, necessidade e maneira explícita com que os requisitos são especificados;

O fator fidedignidade é atingido através dos seguintes sub-fatores de qualidade:

fidelidade: é a característica de uma especificação de requisitos estar escrita coerentemente com as necessidades do usuário;

consistência: é a característica de uma especificação de requisitos estar escrita isenta de contradições;

não ambiguidade: é a característica de uma especificação de requisitos estar escrita isenta da possibilidade de diferentes interpretações para qualquer requisito.

O fator suficiência é atingido através dos seguintes sub-fatores de qualidade:

completeza: é a característica de uma especificação de requisitos conter todos os requisitos determinantes do software em desenvolvimento, completamente definidos;

necessidade: é a característica de uma especificação de requisitos estar escrita isenta de descrições inúteis;

ser explícita: é a característica de uma especificação de requisitos estar escrita isenta de definições implícitas.

Uma especificação deve poder ser utilizada para:

- realização de alterações;
- ser reutilizada em forma parcial ou total, em outros produtos similares;
- ser avaliada;
- ser implementada, depois de feito um estudo completo de sua viabilidade.

O objetivo utilizabilidade se realiza, portanto, através dos seguintes fatores de qualidade:

manutenibilidade: refere-se à capacidade de uma especificação permitir a introdução de alterações após ter sido inicialmente definida, construída e aceita;

reutilizabilidade: refere-se à capacidade de uma especificação ter suas partes organizadas e completamente desenvolvidas de maneira a permitir sua reutilização parcial ou total em outras aplicações;

avaliabilidade: refere-se à capacidade de uma especificação poder ser avaliada com relação a suas propriedades inerentes e com relação ao produto que especifica;

viabilidade de implementação: refere-se à capacidade de uma especificação de requisitos poder ser implementada.

O fator manutenibilidade é atingido através dos seguintes sub-fatores:

evolutibilidade: é a característica de uma especificação de requisitos permitir refinamentos sucessivos;

modificabilidade: é a característica de uma especificação de requisitos permitir que alterações sejam feitas com facilidade, de forma completa e consistente, sem perda de qualidade.

O fator reutilizabilidade é atingido através dos seguintes sub-fatores:

aplicabilidade: é a característica de uma especificação de requisitos poder ser reutilizada ao se especificar outro produto na mesma área de aplicação;

adaptabilidade: é a característica de uma especificação de requisitos poder ser adaptada para corresponder, perfeitamente, a outra aplicação.

O fator avaliabilidade é atingido através dos seguintes sub-fatores:

verificabilidade: é a característica de uma especificação de requisitos poder ser avaliada em relação às propriedades desejáveis para um documento deste tipo;

validabilidade: é a característica de uma especificação de requisitos poder ser avaliada em relação ao produto que especifica.

O fator viabilidade de implementação é atingido através dos seguintes sub-fatores:

viabilidade econômica: é o atributo de se poder construir o software tendo como critério de avaliação a relação entre os custos e os benefícios;

viabilidade financeira: é o atributo de se poder construir o software tendo como critério de avaliação a existência e a disponibilidade de capital necessário para conduzir o desenvolvimento;

viabilidade tecnológica: é o atributo de se poder construir o software tendo como critério de avaliação a posse e o domínio da tecnologia necessária para conduzir o desenvolvimento;

viabilidade de mão-de-obra: é o atributo de se poder construir o software tendo como critério de avaliação a existência e a disponibilidade de mão-de-obra necessária para desenvolver o software;

viabilidade de cronograma: é o atributo de se poder construir o software tendo como critério de avaliação as restrições de cronograma;

viabilidade social: é o atributo de se poder construir o software tendo como critério de avaliação as implicações deste software na sociedade.

A figura III dá uma visão geral dos objetivos, fatores, sub-fatores e critérios que determinam a qualidade de especificações.

Para que seja possível realizar avaliações da qualidade de especificações foi elaborado um "Manual de Controle da Qualidade de Especificações" descrevendo detalhadamente objetivos, fatores, sub-fatores e processos de avaliação. Os anexos exemplificam o conteúdo deste Manual. O texto completo pode ser encontrado em CLUNIE (12).

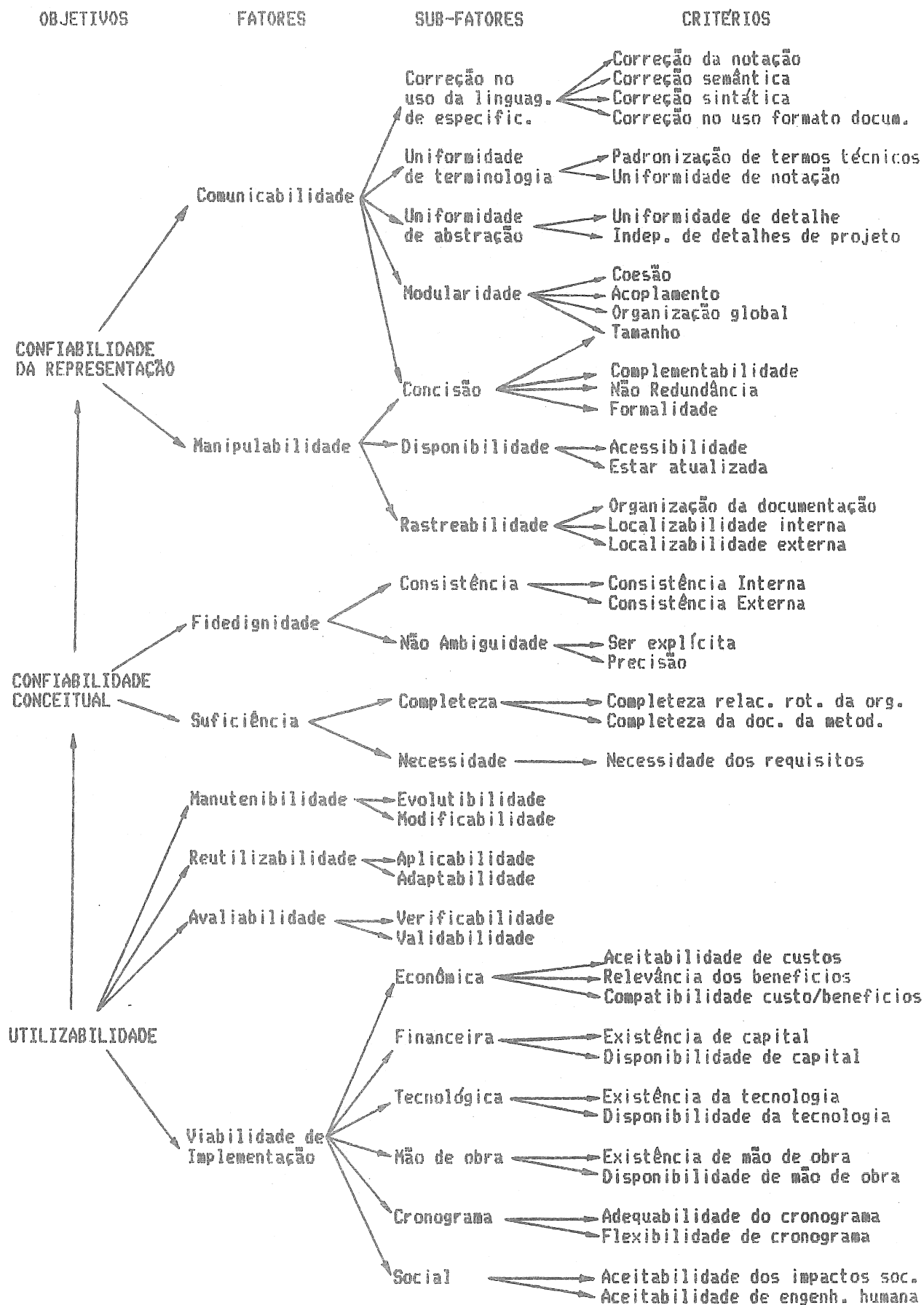


Figura III - Características de Qualidade da Especificação de Requisitos.

## CONCLUSÃO

Neste trabalho partimos de um modelo para avaliação da qualidade de software e identificamos um conjunto de características primitivas que auxiliam na construção de especificações mais confiáveis. A partir destas características definimos um "Manual para Controle da Qualidade de Especificações", cujo objetivo é permitir avaliar uma especificação de forma organizada e disciplinada. A próxima etapa deste trabalho é a construção de um sistema especialista para avaliação de especificações.

## REFERÊNCIAS

- (1) Boehm, B. Software Engineering Economics New Jersey, Prentice Hall, 1981.
- (2) Perlis, A.J. "Controlling Software Development Through the Life Cycle Model", in Software Metrics: An Analysis and Evaluation, MIT Press, 1981.
- (3) Denicoff, M.; Graftan, R. "Software Metrics: A Research Initiative", in Software Metrics: An Analysis and Evaluation MIT Press, 1981.
- (4) Curtis, B. "The Measurement of Software Quality and Complexity", in Software Metrics: An Analysis and Evaluation, MIT Press, 1981.
- (5) Boehm, B. et alli, Characteristics of Software Quality, Amsterdam, North-Holland, 1978.
- (6) McCall, J. "An Introduction to Software Quality Metrics", in, Software Quality Management, Petrocelli, 1979.
- (7) Rocha, A.R.C. da, Um Modelo para Avaliação de Qualidade de Especificações, Tese de Doutorado, PUC-RJ, junho de 1983.
- (8) Rocha, A.R.C. da, Staa, A.V. "How to Improve Objectivity when Measuring Specifications", in, Fifth International Conference of the Israel Society for Quality Assurance, Tel Aviv, Israel, Outubro 1984.

- (9) Rocha, A.R.C. da; Freitas, A.C.T., "How to Evaluate the Conceptual Reliability of Software Specifications", in, Third International Workshop on Software Specification and Design, Londres, UK, 1985.
- (10) Rocha, A.R.C. da; Bargut, M.F.; "Quality Attributes for Microcomputer Programs", in, Sixth International Conference of the Israel Society for Quality Assurance, Tel Aviv, Israel, novembro de 1986.
- (11) Boehm, B., "Verifying and Validating Software Requirements and Design Specification", IEEE Software, vol. 1, nº 1, janeiro de 1984.
- (12) Clunie Beaufond, C.E. Verificação e Validação de Software na Fase de Especificação de Requisitos, Tese de Mestrado, COPPE/UFRJ, fevereiro de 1987.

---

**Critério: Correção semântica****Objetivo: Confiabilidade da representação****Fator: Comunicabilidade****Sub-fator: Correção no uso da ling. de especific.**

---

**Definição:**

A especificação possui o critério **correção semântica** na medida em que está escrita utilizando, de forma correta, o conjunto de regras semânticas pré-definidas na linguagem de especificação.

---

**Discussão:**

A sintaxe trata da maneira de reunir os signos para formar sentenças corretas em uma determinada linguagem. A semântica trata do significado contido nestas sentenças sintaticamente corretas. Embora seja possível gerar sentenças em uma linguagem obedecendo à regras sintáticas, não é garantido que estas sentenças tenham algum significado.

Regras sintáticas não são suficientes para caracterizar uma linguagem. É necessário introduzir-se um outro grupo de regras chamado de regras semânticas. A razão é óbvia: intérpretes podem partilhar a mesma estrutura linguística com regras semânticas diferentes sendo, então, incapazes de se comunicar.

E neste contexto que se introduz a dimensão semântica, a qual pode ser definida como o estudo da relação entre objetos e sua representação.

---

**Processo de avaliação:**

Este critério pode ser avaliado através da resposta à seguinte pergunta:

SIM/NAO

1 ou 0

1. a especificação está escrita obedecendo às regras semânticas pré-definidas na linguagem de especificação?
- 

**Interpretação dos resultados:**

Rever o uso das regras semânticas, caso a resposta seja NAO.

---

**Atributos de qualidade relacionados:**

**Manutenibilidade:** Especificações escritas obedecendo as regras semânticas pré-definidas na linguagem de especificação, facilitam o processo de manutenção da especificação, pois ajudam a torná-la entendível.

**Validabilidade:** Especificações escritas obedecendo as regras semânticas pré-definidas na linguagem de especificação, são possíveis de serem avaliadas com relação a seu conteúdo pois ajudam a torná-la entendível.

---

|                                                                              |                                            |
|------------------------------------------------------------------------------|--------------------------------------------|
| <b>Critério:</b> Completeza com relação ao roteiro definido pela organização | <b>Objetivo:</b> Confiabilidade conceitual |
|                                                                              | <b>Fator:</b> Suficiência                  |
|                                                                              | <b>Sub-fator:</b> Completeza               |

---

**Definição:**

A especificação possui o critério completeza com relação ao roteiro definido pela organização na medida em que os aspectos a serem especificados são baseados num roteiro pré-definido.

---

**Discussão:**

A especificação deve apresentar a usuários e desenvolvedores uma descrição completa e objetiva das condições a que deve obedecer o software que vai ser construído. Para que isto aconteça, deve ser estabelecido um roteiro para elaboração da especificação. As especificações produzidas deverão obedecer a este roteiro.

Existe a possibilidade de que uma determinada seção do roteiro não seja aplicável ao problema. Nesse caso, deve ser incluída uma breve explicação do motivo pelo qual não é aplicável.

---

**Processo de avaliação:**

Este critério pode ser avaliado através das respostas ao seguinte questionário:

SIM/NAO

1 ou 0

- |                                                                                                                             |     |
|-----------------------------------------------------------------------------------------------------------------------------|-----|
| 1. todas as funções a serem desempenhadas pelo software, estão definidas?                                                   | --- |
| 2. todas as necessidades de informação estão identificadas?                                                                 | --- |
| 3. todas as interfaces com o usuário estão identificadas?                                                                   | --- |
| 4. todas as interfaces com o ambiente de hardware estão identificadas?                                                      | --- |
| 5. todas as interfaces com outros softwares estão identificadas?                                                            | --- |
| 6. todas as características relativas ao desempenho do software estão identificadas?                                        | --- |
| 7. todas as categorias possíveis de tratamento de erros e exceções, por falhas de hardware e software, estão identificadas? | --- |
| 8. todas as características de qualidade exigidas pelo usuário estão identificadas?                                         | --- |
| 9. todas as características de qualidade inerentes ao software são identificadas?                                           | --- |
| 10. outros requisitos necessários estão identificados?                                                                      | --- |
| 11. todas as referências estão definidas de forma precisa na especificação?                                                 | --- |

---

|                                                                       |                                     |
|-----------------------------------------------------------------------|-------------------------------------|
| Critério: Completeza com relação ao roteiro definido pela organização | Objetivo: Confiabilidade conceitual |
|                                                                       | Fator: Suficiência                  |
|                                                                       | Sub-fator: Completeza               |

---

Interpretação dos resultados:

Rever a especificação caso existam perguntas com respostas NAO.

---

Atributos de qualidade relacionados:

Viabilidade de Implementação: Só é possível implementar adequadamente o produto que se especifica, se o conteúdo da especificação atende às necessidades dos usuários, quando está completa.



## Anexo 4

---

Objetivo : Utilizabilidade  
Fator : Reutilizabilidade  
Sub-fator: Aplicabilidade

---

Definição: É a característica de uma especificação de requisitos poder ser reutilizada ao se especificar outra aplicação na mesma área do domínio do problema.

---

### Avaliação

Este sub-fator é avaliado através dos critérios:

- Padronização de termos técnicos;
- Coesão;
- Acoplamento;
- Organização Global.