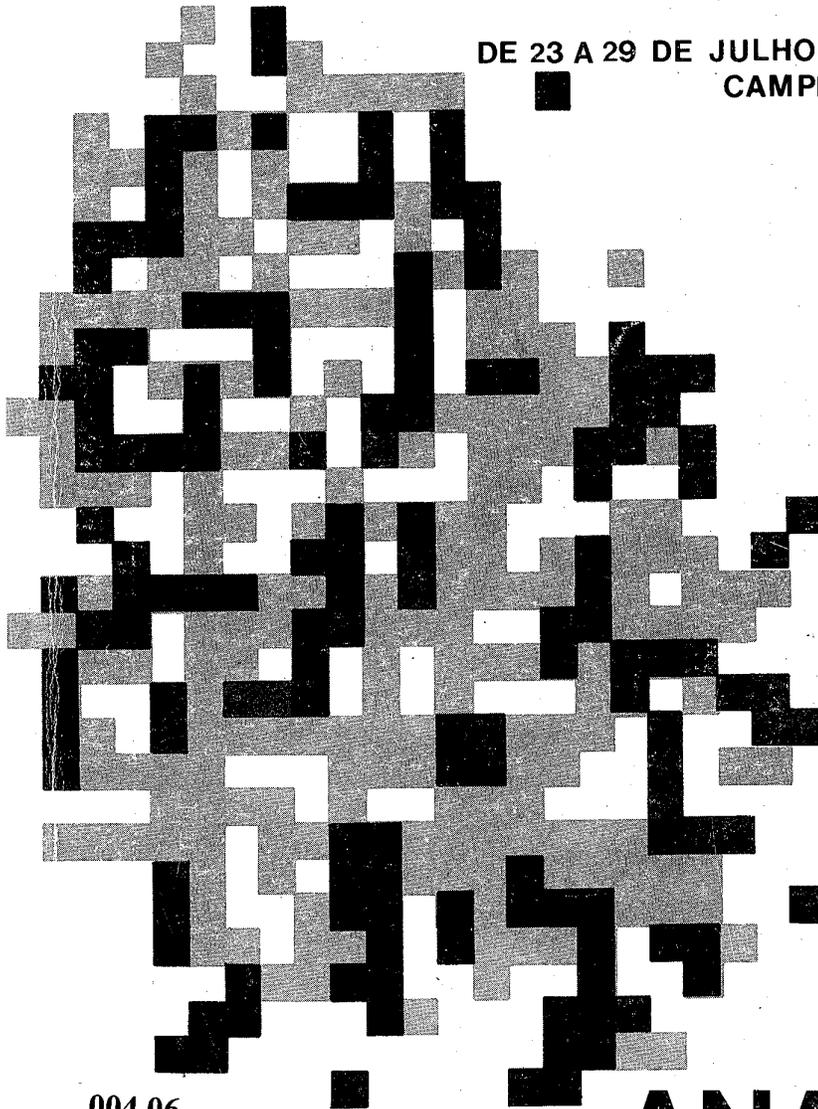


# III CONGRESSO

DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

DE 23 A 29 DE JULHO DE 1983  
CAMPINAS - SP



004.06  
S471  
v.1

**ANAIIS**  
VOL. I

III CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO  
CAMPINAS 23 a 29 de julho de 1983

ANAIS  
VOLUME I

TRABALHOS APRESENTADOS  
X SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE

EDITORES: N. MEISEL, L.J. BRAGA-FILHO

PROMOÇÃO: SOCIEDADE BRASILEIRA DE COMPUTAÇÃO - SBC  
UNIVERSIDADE ESTADUAL DE CAMPINAS - UNICAMP

PATROCÍNIO: CAPES, CNPq, DIGIBRÁS, FINEP, SEI

CO-PATROCINADORES: CPqD/TELEBRAS, PUC-Campinas, UFV  
PREFEITURA MUNICIPAL DE CAMPINAS

QUALIDADE DE ESPECIFICAÇÕES E SUA REPERCUSSÃO SOBRE METODOLOGIAS E  
LINGUAGENS DE ESPECIFICAÇÃO

A.R.C. da Rocha\*

A.v. Staa\*\*

SUMÁRIO

O desenvolvimento de software de boa qualidade depende da qualidade de suas especificações. Especificações de baixa qualidade conduzem à construção de sistemas automatizados também de baixa qualidade. Neste trabalho é definido um modelo para avaliação e predição da qualidade de especificações. Utilizando este modelo é uma análise de várias metodologias e linguagens de especificação atualmente disponíveis ou em desenvolvimento, formula-se um conjunto de hipóteses sobre as características de uma seqüência de linguagens de especificação que sirva de apoio eficaz a todo o ciclo de desenvolvimento do software.

ABSTRACT

Software quality depends directly on the quality of its specifications. It will be almost impossible to construct high quality software unless high quality specifications are used. In this paper a specification quality model is outlined. With basis on this model and the study of current methodologies and specification languages we present the characteristics of a set of specification languages which support the whole life cycle.

\* Bacharel em Matemática (UFRJ,1970). Mestre em Informática (PUC-RJ,1978); em doutoramento na PUC-RJ, engenharia de software, especificações.

\*\* Engenheiro mecânico (PUC-RJ,1965);Mestre em Informática (PUC-RJ,1969); Doutor em Ciências da Computação (University of Waterloo,1974), linguagens de programação,engenharia de software,especificações;Professor Associado da PUC-RJ.

(\*,\*\*)Departamento de Informática.Puc-RJ, Rua Marquês de São Vicente, 225 , Gávea, Rio de Janeiro-RJ, Cep.: 22453.

O desenvolvimento de software de boa qualidade depende da qualidade de suas especificações. Para atuar de forma controlada, de modo a poder avaliar e posteriormente garantir o nível de qualidade alcançado, torna-se necessário possuir um modelo de qualidade. Dentro deste objetivo, definimos um modelo de qualidade de estrutura semelhante ao utilizado por Mc Call [Mc Call 79] para avaliar a qualidade de software. Diferencia-se, no entanto, no que se refere aos objetivos, fatores e critérios específicos a serem utilizados para avaliar a qualidade de especificações. O modelo apresentado está baseado em seis conceitos fundamentais :

1. OBJETIVOS DE QUALIDADE - que são as propriedades gerais que especificações devem possuir;
2. FATORES DE QUALIDADE DE ESPECIFICAÇÕES - que são condições e características determinantes da qualidade de especificações do ponto de vista do leitor (usuário) desta especificação;
3. FATORES DE QUALIDADE DA ENGENHARIA DE ESPECIFICAÇÕES - que são condições e características da construção de especificações e determinantes de sua qualidade;
4. CRITÉRIOS - que definem os atributos de qualidade de especificações, bem como o processo e a escala de medida;
5. MÉTRICAS - que são medidas quantitativas de atributos das especificações efetuadas segundo os critérios correspondentes;
6. FUNÇÕES DE NORMALIZAÇÃO - que permitem quantificar os fatores em função de avaliações de métricas.

Em [Rocha 82], pode-se encontrar uma definição do modelo genérico. Este modelo deve ser completado e adaptado às características e requisitos de qualidade das diferentes empresas e/ou projetos.

Especificações e projetos são produzidos ao longo do ciclo de vida do sistema automatizado através de refinamentos sucessivos, partindo de uma visão mais abrangente para outras mais detalhadas. A produção destes documentos torna necessária a existência de linguagens de especificação. A partir do modelo de qualidade foram estudadas [Rocha 82] várias metodologias e linguagens de especificação. Como consequência deste estudo formulou-se um conjunto de hipóteses sobre as características desejáveis para metodologias e linguagens de especificação. Com isso pretende-se resolver, em parte, o problema formalidade - verificabilidade X naturalidade - envolvimento do usuário. O conjunto de hipóteses formulado é o seguinte :

H.1 - Metodologias que apóiam apenas parte do ciclo de desenvolvimento de software são pouco eficazes, isto é, deve-se tender ao uso de metodologias e ferramentas que sirvam de apoio a todo o ciclo de vida ou que possam ser facilmente integradas com outras de modo que o conjunto seja um apoio eficaz durante todo o ciclo de vida.

H.2 - No início do desenvolvimento deve-se, sobretudo, facilitar a interação com o usuário, isto é, no início quando o conhecimento é ainda precário e é necessária uma grande interação com o usuário, deve-se buscar um menor grau de formalidade e os fatores de qualidade da engenharia a serem privilegiados devem ser a inteligibilidade e a naturalidade da linguagem de especificação.

H.3 - Em fases posteriores deve-se crescer em formalidade, isto é, no final, quando o conhecimento já deve ser perfeito, as especificações devem utilizar uma linguagem formal. Necessita-se um maior grau de formalidade, que permita realizar verificações formais e que seja uma transição harmônica para a formalidade requerida pelo código.

H.4 - Não é desejável o uso de metodologias onde inexistente um suporte automatizado. Referências na literatura [Delisle 81], [Schindler 81], [Willis 81] ao uso de metodologias onde inexistente este suporte, apontam uma série de carências que motivaram seus usuários a desenvolver suporte automatizado para as mesmas.

Uma solução na tentativa de compaginar estes fatores é a criação de uma seqüência de linguagens de especificação com membros adequados a cada um dos níveis de detalhe, de modo que a passagem de um nível para outro se faça de uma maneira harmônica, sem rupturas desnecessárias. Para isto, esta seqüência de linguagens deve possuir membros progressivamente mais formais, à medida que cresce o nível de detalhe e se avança no ciclo de desenvolvimento do sistema. Isto nos levou a formular uma quinta hipótese:

H.5 - Deve existir um mecanismo semi-automático para auxiliar na passagem de um nível para outro, na seqüência de linguagens, isto é, para que a passagem de um nível a outro se dê sem rupturas, torna-se necessária a existência de auxílios semi-automatizados para tradução de uma linguagem para outra na hierarquia.

O projeto desta seqüência de linguagens será desenvolvido tendo em conta

as linguagens e ferramentas atualmente disponíveis. Para isto serão feitos ajustes, correções e ampliações nas linguagens e ferramentas selecionadas como base, procurando integrá-las de uma maneira harmônica.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [Delisle 81] Delisle, N.M. et al  
"Tools for supporting structured analysis"; in Automated Tools for Informations Systems Design; Schneider, H.J.; Wasserman, A.I. eds.; North-Holland; Amsterdam; 1982.
- [Mc Call 78] Mc Call, J.  
"An introduction to software quality metrics"; in Software Quality Management; Cooper, J.D.; Fisher, M.J. eds.; Petrocelli; 1978.
- [Rocha 82] Rocha, A.R.C. da; Staa, A.v.  
"Qualidade de especificações"; in Revista Brasileira de Computação; 2(2), 1982.
- [Schindler 81] Schindler, M.  
"Today's software tools point to tomorrow's tool systems in Electronic Design; Julho 1981.
- [Willis 81] Willis, R.R.  
"AIDES : Computer aided design of software systems-II"; in Software Engineering Environments; Hunke, H. ed; North-Holland; 1981.