

AValiação DA QUALIDADE DE SOFTWARE EDUCACIONAL: UMA POSSIBILIDADE DE ESTRUTURAÇÃO DE CRITERIOS

**Gilda Helena Bernardino de Campos
Ana Regina C. da Rocha**

RESUMO

Este trabalho insere-se no contexto do Projeto TABA [1] cujo objetivo é a construção de uma Estação de trabalho configurável para desenvolvimento de software em diferentes domínios de aplicação.

Este trabalho fornece subsídios para a definição das ferramentas para a avaliação da qualidade de software educacional da Estação TABA. Para isto, identifica quais os fatores mais relevantes para a avaliação da qualidade do software educacional, quais os critérios pertinentes, e finalmente, define um manual que torne viável e eficiente esta avaliação.

INTRODUÇÃO

Diversos autores, entre eles Pressman [2], Manns e Coleman [3]), vem discutindo a questão da qualidade de software. É consenso que qualidade é uma meta a ser perseguida e que o software é um produto complexo que exige em seu desenvolvimento uma postura disciplinada. Por outro lado, estes autores também demonstram que qualidade é uma palavra do cotidiano utilizada para descrever o grau de excelência de um produto ou serviço sendo, portanto, uma palavra genérica e não de uso exclusivo do software. Segundo Stahl [4] há um ponto comum entre os autores quando consideram que software de qualidade é aquele que atende às expectativas e necessidades do usuário. Mas, a qualidade de software não surge espontaneamente, e sim através de um conjunto de procedimentos, cuidadosamente, observados ao longo do desenvolvimento do software. Qualidade não pode ser definida universalmente e está, sempre associada a alguma coisa. Assim sendo, qualidade deve ser definida para o item em questão (qualidade de especificação, qualidade de projeto, qualidade de programas, etc...).

Garvin (in Manns e Coleman [3]) sugere que existem diferentes dimensões de qualidade. Neste trabalho consideramos qualidade como um conceito multidimensional que se realiza através de um conjunto de atributos ou características. Freitas et alii [5] ainda afirmam que qualidade deve ser definida em seu contexto porque a importância dos atributos ou características é variável segundo o domínio de aplicação. Assim sendo, não é o mesmo considerar-se software científico, software de tempo real ou, no nosso caso, software educacional.

Assim sendo, podemos definir qualidade de software como um conjunto de propriedades a serem satisfeitas em determinado grau, de modo que o software satisfaça as necessidades de seus usuários.

Evidentemente, é necessário que esta qualidade seja controlada e avaliada e, para tanto, são necessários métodos. Estes métodos atualmente, compõem a área de controle da qualidade definida como organização sistemática de todos os procedimentos necessários à confirmação de que o módulo ou produto está de acordo com as necessidades estabelecidas em sua fase de desenvolvimento. Manns e Coleman [3, p.5] acreditam que a definição de controle de qualidade é arbitrária e sugerem, como a definição mais difundida, aquela fornecida pelo AINSI/IEEE (1981): "Modelo planejado e sistemático de todas as ações necessárias para assegurar que o software opera de acordo com os requisitos técnicos".

MÉTODO PARA AVALIAÇÃO DA QUALIDADE DE SOFTWARE

Em trabalho anterior, desenvolvemos um método para avaliação da qualidade de software que está baseado nos seguintes conceitos:

- *Objetivos de qualidade*- Determinam as propriedades gerais que o produto deve possuir.
- *Fatores de qualidade do produto*- Determinam a qualidade do ponto de vista dos diferentes usuários do produto (usuário final, mantenedores,etc...)
- *Crítérios*- Definem atributos primitivos possíveis de serem avaliados.
- *Processos de avaliação*- Determinam os processos e os instrumentos a serem usados de forma a se medir o grau de presença, no produto, de um determinado critério.
- *Medidas*- Indicam o grau de presença, no produto, de um determinado critério.
- *Medidas agregadas*- Indicam o grau de presença de um determinado fator e são resultantes da agregação das medidas obtidas da avaliação segundo os critérios.

Os objetivos de qualidade são atingidos através dos fatores de qualidade, que podem ser compostos por outros fatores, que por sua vez, são avaliados através de critérios. Os critérios definem atributos de qualidade para os fatores. Medidas são os valores resultantes da avaliação de um produto segundo um critério específico.

OBJETIVOS DO SOFTWARE

Considerando-se que o software é desenvolvido para atender às necessidades de seus usuários e que deve ter uma vida útil produtiva e longa, ele deve atingir determinados objetivos:

- *Confiabilidade conceitual*, porque o produto precisa satisfazer às necessidades e requisitos que motivaram sua construção;
- *Confiabilidade da representação*, que refere-se às características de representação do produto que afetam sua compreensão e manipulação, e,
- *Utilizabilidade*, que determina a conveniência e a viabilidade de utilização do produto ao longo de sua vida útil. Para que um produto seja utilizável, são necessárias a confiabilidade conceitual e a confiabilidade de representação.

FATORES DE QUALIDADE DO SOFTWARE

Os objetivos são atingidos através de fatores e sub-fatores. O objetivo Confiabilidade da representação é atingido através de dois fatores: *Legibilidade e Manipulabilidade*. O fator Legibilidade avalia a possibilidade de diferentes pessoas entenderem o programa com relativa facilidade para que possam utilizá-lo. Relacionados a este fator tem-se os sub-fatores *Clareza e Concisão*. O fator manipulabilidade avalia a possibilidade de diferentes pessoas manipularem o programa com facilidade. É atingido através de três sub-fatores: *Disponibilidade, Estrutura e Rastreabilidade*.

O objetivo *Confiabilidade Conceitual* é caracterizado pela implementação satisfatória do que foi especificado e projetado, correspondendo, desta forma, às necessidades que geraram seu desenvolvimento. Dois fatores contribuem para o alcance deste objetivo: *Fidedignidade e Integridade*. O fator Fidedignidade avalia a correspondência do programa às especificações e ao projeto. Este fator é atingido através de três sub-fatores: *Precisão, Completeza e Necessidade*. O fator Integridade está relacionado à capacidade do programa resistir a situações hostis (dados errados, agressões etc..). Dois sub-fatores permitem o alcance deste fator: Robustez e Segurança.

Utilizabilidade é a característica de qualidade de programas que determina a conveniência e a viabilidade de sua utilização ao longo do tempo. Este objetivo exige a existência dos outros dois: Confiabilidade da representação e Confiabilidade conceitual. Diversos fatores estão relacionados a esse objetivo:

Manutenibilidade, que avalia a facilidade com que o programa pode ser adaptado a fim de atender às necessidades de modificações que surgirem depois de seu desenvolvimento;

Operacionalidade que avalia a facilidade de comunicação com o usuário. Este fator possui dois sub-fatores: Oportunidade e Amabilidade ao uso.

Portatibilidade é a característica de um programa poder ser operado de maneira fácil e adequada em diferentes configurações de equipamentos além da original;

Reutilizabilidade é a característica que avalia a possibilidade do reaproveitamento, total ou parcial, de funções desenvolvidas em um programa em outras aplicações;

Eficiência é a característica do programa realizar suas funções sem desperdício de recursos (memória e periféricos entre outros);

Rentabilidade é a característica do programa ter uma relação custo-benefício aceitável;

Avaliabilidade, é a característica que avalia a facilidade com que um programa pode ser avaliado. Este fator possui dois sub-fatores: Verificabilidade e Validabilidade.

A figura 1 mostra a relação entre os objetivos, fatores e subfatores. A figura 2 mostra a definição dos sub-fatores.

Figura 1. Estrutura do método para avaliação da qualidade de software [6].

Sub-fatores	Definições
<i>Clareza</i>	Funções codificadas de forma clara e de fácil entendimento.
<i>Concisão</i>	Funções implementadas com a quantidade mínima de código.
<i>Estilo</i>	Codificação com recursos que facilitam a compreensão do código.
<i>Modularidade</i>	Implementação do programa com uma estrutura o mais independente possível de outros módulos.
<i>Disponibilidade</i>	Atualização do programa e de sua documentação.
<i>Estrutura</i>	Organização hierárquica das partes que compoem o programa.
<i>Rastreabilidade</i>	Caminhamento através do programa e de sua documentação.
<i>Precisão</i>	Exatidão dos cálculos e resultados de forma que satisfaça a utilização pretendida pelo usuário.
<i>Completeza</i>	Implementação de todas as funções especificadas.
<i>Necessidade</i>	Implementação apenas das funções que foram especificadas.
<i>Robustez</i>	Resistência do programa a situações hostis.
<i>Segurança</i>	Habilidade de evitar falhas que possam provocar consequências desastrosas.
<i>Oportunidade</i>	Produção de resultados em tempo hábil.
<i>Amenidade ao uso</i>	Interação com o usuário de forma simples e natural, segundo suas aptidões.
<i>Verificabilidade</i>	Facilidade de avaliar o programa com relação à forma e representação.
<i>Validabilidade</i>	Facilidade de avaliar se o programa executa a função para a qual foi desenvolvido.

Figura 2. Objetivos, fatores e sub-fatores de qualidade de programas

O SOFTWARE EDUCACIONAL

Desde a década de 70, o software educacional vem entrando no mercado mundial de forma acelerada. Inúmeros países como Inglaterra, França e E.U.A., desenvolveram projetos de uso do computador em educação e conseqüentemente necessitaram desenvolver produtos de software específicos para suas necessidades. O mesmo ocorreu no Brasil onde diversos projetos de pesquisa vem sendo desenvolvidos não só relacionados ao uso do microcomputador em sala de aula como, também, ao desenvolvimento de software para os mais diversos conteúdos programáticos. Alguns grupos de pesquisa vem utilizando o termo software educacional ou software educativo, outros o termo "courseware". outros

ainda o termo programas educativos por computador (PEC). Todos estes termos possuem o mesmo significado: material educacional para microcomputadores [7].

A disseminação e utilização do software educacional comercializado foram assinaladas por Sthal [4]. Mas, os especialistas da área de Informática na Educação constataam que existem pouquíssimos estudos e resultados de pesquisas relativos ao desempenho e efetividade deste tipo de software [8]. Estes especialistas sugerem a necessidade da introdução de uma discussão mais ampla sobre a questão da qualidade do software educacional. Como Dudley-Marling e Owston [9] e Bork [10] assinalaram, as questões relativas a qualidade do software ainda persistem embora sua quantidade esteja aumentando. Inúmeras dificuldades contribuem para que o software educacional seja de baixa qualidade e podemos citar, como exemplo, (a) o pouco preparo de recursos humanos na área educacional como ressaltam Ennals, Gwyn e Zdrachev [11]; (b) a pressão mercadológica dos fabricantes de hardware [12]; (c) a produção descentralizada de programas para ensino [4]; e, (d) a quantidade de horas necessárias para desenvolvimento e implementação [13].

Além disso, a aplicação na área educacional é bastante complexa pois envolve características de utilização nem sempre consideradas em outras aplicações. Para demonstrar as utilizações da informática em educação Knezek, Rachlin, Sidney e Scannel [14] categorizaram os usos que o computador pode ter em educação.

Figura 3. Taxonomia para informática na educação (Knezek et alii, in Campos [7]).

A categoria Disseminação de Conhecimento proposta por Knezek et alii [14], demonstra o uso do computador como tutor onde estão discriminadas as modalidades de uso do software educacional. Esta sub-categoria é composta das seguintes aplicações: (a) exercício e prática; (b) tutoriais; (c) jogos e simulações; (d) resolução de problemas, e, (e) demonstração.

EXERCÍCIO E PRÁTICA

No contexto histórico, a forma tradicional na qual os alunos tem trabalhado no computador tem sido para o uso de reforço e recuperação. O exercício e prática, operacionalizado por meio de pergunta/resposta é o modelo de grande parte dos programas educacionais disponíveis [15]. O programa de exercício e prática bem elaborado é "paciente" e pode ser modelado para ir de encontro às necessidades individuais de cada aluno. Em geral, utilizam um "feedback" positivo e não julgam as respostas erradas. Os alunos trabalham

com uma seleção aleatória de problemas, repetindo o exercício quantas vezes forem necessárias para atingirem os objetivos determinados no programa. As respostas erradas são rapidamente detectadas, o que reduz a possibilidade de reforço em procedimentos errôneos.

TUTORIAIS

Nesta forma de trabalho, o computador introduz conceitos inteiramente novos ou, então expande os conhecimentos já adquiridos em aulas anteriores. Os programas tutoriais são, em geral, elaborados para o uso individual do aluno. Utilizam na programação a técnica de ramificação onde o programador adapta o programa às necessidades específicas do aluno. As respostas, incorretas ou não adequadas, redirecionam o programa para uma explicação ou, então, para a expansão do conceito. Após esta ramificação, o aluno retorna ao programa principal.

Atualmente, a capacidade interativa e os recursos disponíveis para elaboração dos programas oferecem inúmeras possibilidades. Cabe salientar que esta modalidade é uma das modalidades mais difíceis de serem produzidas, pois os programas precisam estar relacionados a cursos específicos e necessitam de um bom planejamento para o professor, monitor e aluno. Segundo Lathrop e Goodson [15], os programas tutoriais serão mais eficazes quando os objetivos da aprendizagem forem bem definidos e passíveis de serem medidos com facilidade. Para serem considerados programas aceitáveis, eles precisam utilizar plenamente a capacidade interativa do microcomputador. Além disso, o conteúdo deve ser apresentado em pequenas porções e os passos devem ser controlados pelo aluno de forma a não sair do fluxograma da instrução do programa. Os segmentos do programa devem estar acessíveis quando forem necessárias revisões do produto. Os programas tutoriais devem estar correlacionados ao currículo das escolas e é de suma importância que sejam acompanhados por documentação explicativa. Em um programa tutorial bem desenvolvido, o aluno envolve-se em um diálogo com o professor/ autor/ programador e utiliza o computador como um veículo de comunicação.

Figura 4. Especificação da categoria de Disseminação do conhecimento (Knezek et alii [14], in Campos [7]).

SIMULAÇÕES E JOGOS

"Uma simulação é a representação ou uma modelagem de um objeto real, de um sistema ou evento" [16]. Inúmeros fenômenos podem ser representados neste modelo simbólico e representativo da realidade. As características de realismo e imaginação são básicas à simulação; dependem do fenômeno trabalhado e de seu objetivo. Os componentes principais e invariáveis são: (a) o sistema de apresentação; (b) o aluno; (c) o sistema de controle, e, (d) o sistema de gerenciamento.

A apresentação do sistema pode trabalhar o conteúdo sob diversas formas, desde que estimule o aluno-usuário. O usuário é o elemento participante da simulação que controla o sistema, isto é, as variáveis e os parâmetros. O sistema gerenciador deve possuir capacidade de decisão e memória. O gerenciador ajusta a simulação baseado na manipulação dos sistemas de controle pelos alunos-usuários [16]. Cabe observar que as simulações podem ser estáticas ou interativas.

Os jogos em computador devem ser uma fonte de recreação. Geralmente envolvem elementos de desafio ou competição. Muitos jogos são confundidos com simulações, pois utilizam algum tipo de habilidade. Por outro lado, alguns jogos são programados para a aprendizagem de alguma habilidade. Merrill et alii [16] acreditam que estes jogos deveriam ser, mais adequadamente chamados de jogos de simulação. Os atributos motivacionais dos jogos são vários. Diversos autores, entre eles Bitter e Camuse [17] e Coburn et alii [13], identificaram os atributos básicos de jogos educacionais por computador. Estes atributos sugerem critérios de seleção para o uso. Malone e Lepper (in, Merrill et alii [16]) classificaram os atributos motivacionais em duas categorias: individual e interpessoal. São considerados motivações individuais o desafio, a curiosidade, o controle e a fantasia. São consideradas motivações interpessoais a cooperação, a competição e o reconhecimento.

RESOLUÇÃO DE PROBLEMAS

O enfoque de resolução de problemas no ensino por computador se baseia na hipótese de que o raciocínio necessário para escrever um programa ajuda o desenvolvimento da técnica de resolução de problemas em geral. A filosofia subjacente neste enfoque, segundo O'Shea e Self [18], consiste em aprender fazendo, enquanto a simulação corresponderia a um aprender observando. Bitter e Camuse [17], no entanto, ressaltam que o software de resolução de problemas requer estratégia por parte do usuário. Uma situação problema é apresentada ao aluno-usuário e ele deve resolvê-la. Os conceitos não são ensinados e os alunos utilizam o que já sabem, aprendem com seus erros e redefinem padrões a medida em que atingem o domínio de determinadas técnicas. Neste sentido, muitos autores salientam que a resolução de problemas é, na verdade, parte integrante de outras modalidades.

As modalidades de uso do software educacional demonstram como o universo de avaliação da qualidade deste produto é amplo e como é fundamental organizar e clarificar critérios para que possa haver confiança e credibilidade do software, reduzindo o nível de subjetividade.

RELEVÂNCIA DOS FATORES DE QUALIDADE NO SOFTWARE EDUCACIONAL

Os fatores de qualidade do software educacional tem sido discutidos tanto por especialistas da área de educação como por especialistas da área de informática. Existe uma certa indefinição entre os educadores sobre quem deve elaborar os programas. Alguns autores [15] deixam claro que os professores não precisam se tornar programadores nem especialistas em ciência da computação e outros [16] tem como certo que a maior parte dos professores não desenvolverão coursewares. Estes autores, entretanto, acreditam que os especialistas em educação devem aprender a avaliar programas prontos. Inúmeras organizações internacionais como CONDUIT [19], EPIE Institute [20], MECC [21], Scholastic [22] e especialistas como Dassance [23], Oliveira et alii [24] vem estudando e desenvolvendo categorias e critérios para a avaliação de software educacional. Estes critérios, no entanto, se referem a produtos já desenvolvidos e em disponibilidade no mercado de tal forma que o professor possa selecionar o software de acordo com sua clientela e para um contexto determinado. Porém, como afirma Stahl [4] "é necessário que fatores de qualidade para o desenvolvimento do produto sejam definidos e utilizados". Stahl [4] selecionou a partir do método para avaliação da qualidade de software definido por Rocha [6] quais os fatores e sub-fatores mais relevantes para a avaliação de software educacional, durante seu processo de desenvolvimento, que são apresentados a seguir.

Objetivo	Fator
Utilizabilidade	Operacionalidade Eficiência Manutenibilidade Portatilidade Rentabilidade Avaliabilidade
Confiabilidade Conceitual	Integridade
Confiabilidade da Representação	Legibilidade

Figura 5. Objetivos e fatores selecionados por Stahl [4].

Este trabalho partiu dos fatores e sub-fatores selecionados por Stahl [4] e, acrescentou novos fatores e sub-fatores, definindo critérios e processos de avaliação. A figura 6 fornece uma visão geral dos objetivos, fatores e sub-fatores que identificamos como relevantes para a avaliação do software educacional segundo a especificação do computador como tutor na classificação de Knezek et alii [14].

Objetivo	Fator	Sub-fator	
Utilizabilidade	Operacionalidade	Amenidade ao uso.	Eficiência
	Eficiência do desenvolvimento	Eficiência do processamento.	
	Manutenibilidade	Alterabilidade.	
	Portatibilidade	Independência do ambiente	
	Rentabilidade	Validabilidade	
		Adequação	
		Integração	
Confiabilidade conceitual	Integridade	Robustez	
	Fidedignidade	Necessidade	
Confiabilidade da Representação	Legibilidade	Clareza	

Figura 6. Objetivos e sub-fatores para a avaliação do software educacional.

Alguns sub-fatores foram especificados para o software educacional, além dos que já estavam definidos pelo método para avaliação da qualidade de software proposto por Rocha [6]. Estes sub-fatores estão definidos na Figura 7.

Sub-fatores	Definições
Eficiência de processamento	Característica do programa de realizar suas funções sem desperdício de recursos.
Eficiência do desenvolvimento	Ramificações alternativas para atendimento das necessidades pedagógicas do aluno-usuário.
Alterabilidade	Possibilidade do programa sofrer modificações depois de seu desenvolvimento.
Independência do ambiente	Verificação da compatibilidade do programa com diferentes equipamentos.
Adequação	Adequação do programa ao currículo regular da escola.
Integração	Facilidade de entrosamento outros recursos ou materiais instrucionais.

Figura 7. Definição de sub-fatores acrescentados à figura 2.

MANUAL PARA CONTROLE DA QUALIDADE DO SOFTWARE EDUCACIONAL

Para tornar viável a avaliação de produtos de software educacional foi elaborado um manual para controle da qualidade [12].

O conteúdo do Manual define os objetivos, fatores e sub-fatores. Para cada objetivo foram definidos os fatores de qualidade a ele relacionados e, para cada fator, os sub-fatores pertinentes. São, também, identificados critérios de forma a permitir a realização de avaliações.

Foram selecionados dois tipos de processos de avaliação:

- (a) A medida é obtida através de uma escala de 0 a 1, onde o 0 representa a avaliação mais negativa do critério e o 1, a mais positiva.

! ! ! ! !
0 0,25 0,50 0,75 1

- (b) A medida é do tipo binário, apresentada da seguinte forma: () sim () não

No caso (a), interpreta-se o resultado da medida a partir do valor assinalado na escala. No caso de mais de um critério por fator, recomenda-se utilizar a média ponderada. No caso da medida binária, recomenda-se o somatório dos índices positivos e negativos, utilizando-se como fator para decisão o índice que apresentar o maior resultado.

O ponto de corte definido para qualidade considerada satisfatória foi retirado de estudos desenvolvidos pelo "Military Management Comand, Systems Management Division " [25] adaptado para fins deste Manual.

Valor da medida	Interpretação
0,95 - 1,00	Alta qualidade do software.
0,90 - 0,94	Qualidade boa, devendo-se resolver em paralelo os problemas detectados.
0,60 - 0,89	Qualidade mediana, problemas existentes resultaram em produto final pobre e com custos elevados.
0,00 - 0,59	Sem qualidade, problemas existentes não justificam o uso do produto. O produto deve ser alterado e/ou revisto.

As figuras 8 e 9 mostram exemplos de definição de critérios com os respectivos processos de avaliação.

<i>Critério:</i>	Fornecimento de feedback.		
<i>Objetivo:</i>	Utilizabilidade		
<i>Fator:</i>	Operacionalidade	<i>Sub-fator:</i>	Amenidade ao uso.

Definição: É a possibilidade, interna ao programa, do usuário analisar suas respostas, indicando correção ou erro, além de fornecer o caminho ou a indicação da resposta correta.

Discussão: O fornecimento de feedback ou realimentação ao aluno- usuário é fundamental em um software educacional do tipo tutor. O programa deve ser preparado para analisar as respostas dadas pelo usuário, compará-la com as respostas desejadas e discriminá-las. Caso a resposta esteja errada, o programa deverá escolher uma sequência remediativa que dará a oportunidade ao aluno de responder novamente a questão. A toda resposta errada o programa deverá fornecer a resposta certa.

Processo de avaliação: A avaliação segundo este critério é realizada através da resposta ao seguinte indicador:

A resposta errada não tem explicação.

O programa fornece sequências explicativas para as respostas erradas.

!	!	!	!	!	!
0		0,25		0,50		0,75		1		

Figura 8: Exemplo de definição de critério para fins da elaboração do Manual de avaliação.

<i>Critério:</i>	Clareza dos comandos		
<i>Objetivo:</i>	Utilizabilidade		
<i>Fator:</i>	Operacionalidade	<i>Sub-fator:</i>	Amenidade ao uso.

Definição: É a característica do programa utilizar ícones, convenções ou pedidos de comando de forma que estes correspondam as teclas do dispositivo de entrada do computador.

Discussão: É necessário que exista uma correspondência entre o comando que é pedido pelo programa e as teclas de entrada do computador, pois o aluno-usuário pode não estar familiarizado com o equipamento e não realizar o comando esperado. O uso de ícones e convenções pode facilitar a compreensão do programa.

Processo de avaliação: Resposta ao indicador:

Não há explicação
para os comandos.

Os comandos são
explicados claramente.

!	!	!	!	!	!	!
0					0,25					0,50					0,75					1										

Figura 9: Exemplo de definição de critério para fins da elaboração do Manual de avaliação.

CONCLUSÕES

Este trabalho estabeleceu atributos para avaliação de software educacional do ponto de vista do computador como tutor, segundo a classificação de Knezek et alii [14], in Campos [7]. Partiu-se de trabalhos anteriores (Rocha [6] e Stahl [4] ampliando-se o domínio do tema com novos atributos e processos de avaliação. Para possibilitar seu uso prático, definiu-se um Manual para avaliação da qualidade de software educacional.

Novos trabalhos estão em desenvolvimento com o objetivo de particularizar o Manual para cada um dos tipos de software educacional descritos neste trabalho e em usos emergentes como o hipertexto.

REFERÊNCIAS

1. Rocha, A.R.C. da; Aguiar, T.C. & Souza, J.M. *TABA: a heuristic workstation for software development*. (1990). Tel Aviv:COMPEURO 90.
2. Pressman, Roger S. *Software Engineering. A practioner's approach*. (1897). New York: Macgraw Hill.
3. Manns, Tom & Coleman, Michael. *Software Quality Assurance*.(1988). London: Macmillan Education.
4. Stahl, Marimar M. *Avaliação da Qualidade de software educacional*. (1988). Rio de Janeiro: COPPE/UFRJ.
5. Freitas, Afonso C.; Bargut, Maurício F. & Rocha, Ana Regina C. da. *Características de qualidade de programas*. (1985). Rio de Janeiro:COPPE/UFRJ.
6. Rocha,Ana Regina C. da. *Análise e Projeto Estruturado de Sistemas*. (1987). Rio de Janeiro: Campus.
7. Campos, Gilda Helena Bernardino de. *Construção e Validação de ficha de avaliação de produtos educacionais para microcomputadores*. (1989). Rio de Janeiro:Faculdade de Educação/ UFRJ.
8. Jolicoeur, Karen & Lergner, Dale E. Do we really know what makes educational software effective? A call for empirical research on effectiveness. *Educational Technology*, **26** (12), 7-11, 1986.
9. Dudley-Marling, Curt & Owston, Ronald D. The State of educational software. A criterion based evaluation. *Educational Technology*, **27** 930, 25-29, 1987.
10. Bork, Alfred. *Learning with personal computers*.(1987). New York: Harper & Row Computer Science Technology.
11. Ennals, Richard; Gwyn, Rhys & Zdrachev, Levcho. *Information technology and Education: The changing school*. (1986). England: Ellis-Horwood.
12. Campos,Gilda Helena Bernardino de & Rocha, Ana Regina C. da. *Manual para avaliação da qualidade de software educacional*.(1990).Rio de Janeiro: COPPE/UFRJ.
13. Coburn, Peter et alii. *Practical guide to computers in education*. (1982). New York: Addison-Wesley.
14. Knezek, Gerald A.; Rachlin, Sidney L. & Scannel, Peter. A taxonomy for educational computing. *Educational Technology*, **28** (3), 15-19, 1988.
15. Lathrop, Ann & Goodson, Bobby. *Courseware in the classroom: Selecting, Organizing and using educational software*. (1983). California: Addison-Wesley.
16. Merrill, Paul F.; Tolman, Christensen; Hammons, R. & Vicent, Reynolds. *Computers in Education*. (1986). New Jersey: Prentice- Hall.
17. Bitter, Gary & Camuse, Ruth A. *Using a microcomputer in the classroom*. (1984) Virginia: Reston Publishing Co.
18. O'Shea Jim & Self, John. *Ensenanza y aprendizaje con ordenadores. Inteligencia artificial en educacion*. (1985). Madrid: Ediciones Anaya Multimedia.
19. Conduit. Evaluation questionnaire. In Nancy Baker Jones & Larry Vaughan (editores). *Evaluation of educational software. A guide to guides*. (1983). Massachussets: The northeast regional exchange.
20. Epie Institute. In Nancy B. Jones & Larry Vaughan (editores). *Evaluation of educational software. A guide to guides*. (1983).Massachussets:The northeast regional exchange.
21. Minnesota Education Computing Consortium - MECC. In Nancy B. Jones & Larry Vaughan (editores). *Evaluation of educational software. A guide to guides*. (1983). Massachussets: The northeast regional exchange.
22. Scholastic. In, Poirot, James & Billings, Karen. *Microcomputers in education: A Scholastic in-service training program*. (1982). New York: Scholastic Book Services.
23. Dassance,Richard A. Quality control in the development of courseware utilizing computer based interactive video.*28th international conference proceedings*. (1986).EUA: Association for the development of computer-based instructional systems.
24. Oliveira, Celina C.; Menezes, Eliane & Moreira, Mércia. Avaliação de software educacional. (1986). *Documento base da Reuniao Anual da ABT*. Belo Horizonte: ABT.
25. Cardoso, Rogério Nesi Pereira. *Predição, Estimativa e Medição da confiabilidade durante o ciclo de vida do software*. (1990). Rio de Janeiro: COPPE/ UFRJ.

