

AVALIAÇÃO DA QUALIDADE DE SOFTWARE EDUCACIONAL

Ana Regina Rocha*
Gilda H. Bernardino de Campos*

Introdução

Diversos autores, entre eles Pressman (1987), Manns e Coleman (1988), vêm 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 (1988) há um ponto comum entre os autores quando consideram que software de qualidade é aquele que atende às 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).

Uma revisão da literatura disponível sobre avaliação da qualidade de software educacional (S.E.) sugere que o impacto da introdução deste software no currículo regular das escolas só pode ser avaliado no contexto do projeto educacional como um todo. Desta forma, se os produtos de software educacional podem contribuir efetivamente no processo educacional, técnicas específicas para o controle da qualidade de software devem ser utilizadas. O controle da qualidade de produtos técnicos pode trazer benefícios tanto no que se refere ao desempenho do produto como nos resultados da utilização destes software pelos alunos.

Garvin (in Manns e Coleman, 1988) sugere que existem diferentes dimensões de qualidade. Neste trabalho consideramos qualidade como

* Do COPPE/UFRJ. Programa de Engenharia de Sistemas e Computação.

um conceito multidimensional que se realiza através de um conjunto de atributos ou características. Freitas et al. (1985) ainda afirmam que qualidade deve ser definida em seu contexto porque a importância dos atributos ou características é variável segundo o domínio da aplicação. Assim sendo, não é 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.

Os usuários do software educacional podem ser identificados como desenvolvedores, mantenedores, professores e alunos. Percebe-se que a qualidade para cada categoria terá um significado, pois ela deverá refletir o ponto de vista destes diferentes usuários. Desta forma, o desenvolvimento do S.E. possui características específicas e os atributos que compõem sua qualidade devem ser definidos na fase de análise de requisitos. A especificação da qualidade inclui o modelo de ensino-aprendizagem selecionado, isto é, a filosofia de aprendizagem subjacente ao software. Este é o único padrão a ser especificado a *priori* no desenvolvimento do software educacional.

Davis (1988) apresenta o clássico modelo em cascata para o desenvolvimento de software e seu ciclo de vida, ao qual acrescentamos um passo anterior, referente à definição da teoria de aprendizagem (Figura 1).

Evidentemente, é necessário que a qualidade seja controlada e avaliada e, para tanto, são necessários métodos. Estes métodos atualmente, compõem a área de controle da qualidade definida como organização ou 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 (1988) acreditam que a definição de controle de qualidade é arbitrária e sugerem, como a definição mais difundida, aquela fornecida pelo ANSI/IEEE (1981): "Modelo planejado e sistemático de todas as ações necessárias para assegurar que o software opera de acordo com os requisitos técnicos" (p.5).

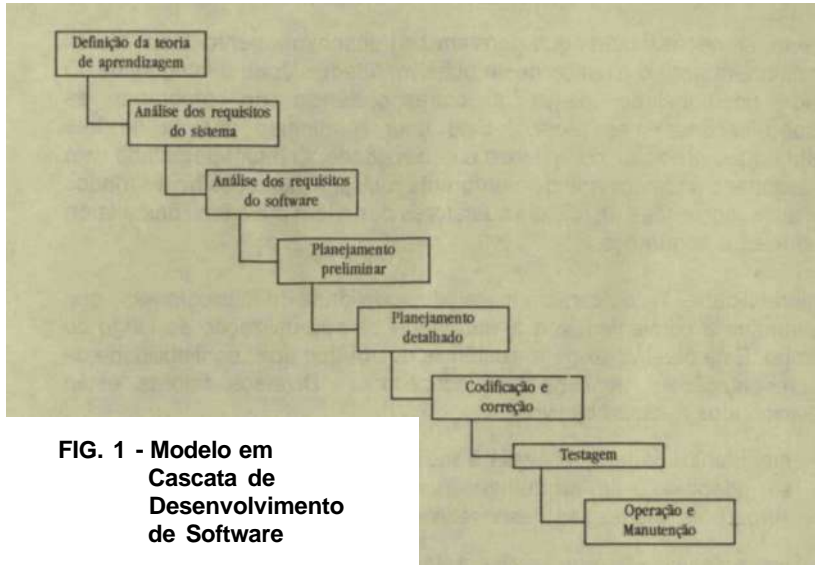


FIG. 1 - Modelo em Cascata de Desenvolvimento de Software

Acrescentar-se-ia, ainda, a esta definição, que o software deve refletir a teoria de aprendizagem especificada por seus autores, o que seria um requisito educacional.

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, alunos e professores)

Critérios — Definem atributos primitivos possíveis de serem avaliados.

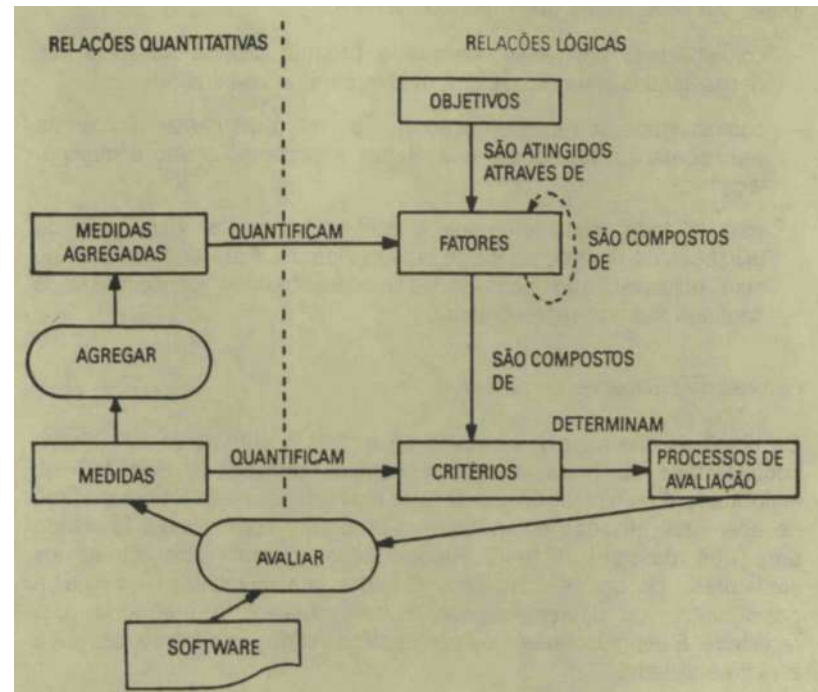


FIG. 2 - Estrutura do Método para Avaliação da Qualidade de Software (Rocha, 1987)

Os objetivos de qualidade, que são atingidos através dos fatores de qualidade, 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 se refere 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 subfatores. 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 têm-se os subfatores: clareza e concisão. O fator manipulabilidade avalia a possibilidade de diferentes pessoas manipularem o programa com facilidade. É atingido através de três subfatores: 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 subfatores: precisão, completeza e necessidade. O fator integridade está relacionado à capacidade de o programa resistir a situações hostis (dados errados, agressões etc). Dois subfatores 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 modificação que surgem depois de seu desenvolvimento;
- operacionalidade que avalia a facilidade de comunicação com o usuário. Este fator possui dois subfatores: oportunidade e amenidade ao uso;
- portabilidade 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 de o programa realizar suas funções sem desperdício de recursos (memória e periféricos entre outros);
- rentabilidade é a característica de o 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 subfatores: verificabilidade e validabilidade.

Quadro 1 — Fatores e Subfatores de Qualidade de Programas

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

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 EUA, entre outros, desenvolveram projetos de uso do microcomputador em educação e conseqüentemente necessitaram desenvolver produtos de software específicos para suas necessidades. O mesmo ocorreu no Brasil onde diversos projetos de pesquisa vêm 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 vêm 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 (Campos, 1989).

A disseminação e utilização do software educacional comercializado foram assinaladas por Stahl (1988). 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 (Jolicoeur e Lerger, 1986). 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 (1987) e Bork (1987) assinalaram, as questões relativas à 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 (1986); b) a pressão mercadológica dos fabricantes de hardware (Campos, 1989); c) a produção descentralizada de programas para ensino (Carnoy e Loop, Stahl, 1988); d) a quantidade de horas necessárias para desenvolvimento e implementação (Coburn, 1982); e) a dificuldade de montagem de uma equipe multidisciplinar que desenvolva trabalho cooperativo (Rocha et al. 1992).

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 (1988) categorizaram os usos que o computador pode ter em educação.

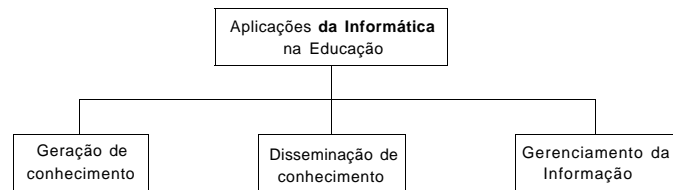


FIG. 3 - Taxionomia para Informática na Educação Segundo Knezek et al (1988), in Campos (1989)

Atributos Necessários

O software educacional tem despertado questões relativas a sua qualidade e à abrangência; verificam-se preocupações em torno da

qualidade dos instrumentos utilizados, propriedade dos critérios avaliativos relacionados e, finalmente, pouca ajuda no que se refere à tomada de decisão quanto ao uso ou não uso do software avaliado. Por outro lado, autores como Ragsdale (1982), Schaefermeyer (1990), Branson (1990) e Underwood (1990) afirmam ainda que a qualidade do software que está sendo produzido é de difícil mensuração e procuram padrões de planejamento e desenvolvimento para o ensino por computador, relacionando-os à aquisição de habilidades mentais básicas e com a aquisição de conhecimento.

Percebe-se, na literatura especializada em Informática e Educação, que os autores procuram definir os atributos necessários ao software para a aprendizagem com a tecnologia.

Ora, a tecnologia computacional permite maior poder de interação com o usuário do que os outros meios tecnológicos usualmente difundidos. É no poder interativo do computador que reside a sua maior potencialidade em educação (Merrill et al. 1986).

Schaefermeyer (1990) afirma que, atualmente, a qualidade educacional do software disponível no mercado depende do que o programador considera um bom *design* instrucional. Argumenta a autora que estes programas possuem mais "charme artístico" do que objetivos, análise de tarefas e sistematização do conteúdo. Recomenda que um mínimo de características deveria ser estabelecido — mesmo que este mínimo seja extremamente amplo, para que haja uma possibilidade concreta e segura de observação.

Twitchell, Anderton e Parry (1990) demonstram que o planejamento instrucional para facilitar a efetiva aquisição de informação e para fornecer estímulos à aprendizagem sempre foi uma preocupação da teoria instrucional. Mas que, curiosamente, muito poucas teorias do *design* instrucional preocuparam-se com a prescrição pragmática e com um caminho objetivo.

Os teóricos da aprendizagem, por sua vez, buscaram teorias de instrução que apontam caminhos para a elaboração e desenvolvimento do software educacional, enquanto material instrucional.

Ambientes Interativos de Aprendizagem com o Computador

Balacheff e Gras (1992) definem a didática como "o estudo das situações respondentes ao projeto social de aquisição de certos conhecimentos pelos alunos, estudantes ou adultos em formação, sob o ponto de vista das características destas situações e aprendizagens que daí resultam". Ressaltam que a particularidade da didática, em relação a outras áreas, quando aplicada em educação ou psicologia, reside na dimensão epistemológica de sua problemática que considera a especificidade do conhecimento a ser trabalhado. Este fato revela a importância das situações geradas onde é determinado um valor funcional aos conhecimentos, aos métodos implícitos ou explícitos que determinam a interação aluno/professor relacionada às situações e às dificuldades cognitivas que podem ser encontradas.

A contribuição da didática às pesquisas, em ambientes interativos de aprendizagem com o computador, é de ordem metodológica e teórica. Trata-se, na verdade, de: a) caracterização e da modelização de situações de ensino; b) análise das condutas das concepções dos alunos, diante de um conteúdo do conhecimento em um contexto determinado; c) estudo dos fenômenos das transferências do saber; d) estudo das formulações do saber; e) métodos de validação.

Sob o ponto de vista da informática, as ferramentas para a pesquisa podem colaborar, essencialmente, na observação detalhada das diferentes fases da introdução e da construção de um conhecimento específico.

Desta forma, a integração da informática, nas situações de ensino, introduz uma nova complexidade no campo da didática, pois permite abordar a modelização computacional dos processos didáticos. Esta modelização é uma nova área de pesquisa, possui características próprias tais como a necessidade de constituição pelo aluno de representações concernentes à organização e ao funcionamento do computador e à interação desta habilidade com o conteúdo a ser trabalhado.

Mendelsohn (1990), ao estudar os ambientes inteligentes de

aprendizagem (AIA), demonstrou que o surgimento e o desenvolvimento das tecnologias educativas podem provocar uma renovação da pesquisa cognitiva, pois as novas tecnologias de tratamento da informação permitem imaginar e realizar cenários a fim de assistir e melhorar a eficácia da aprendizagem através do ensino. Três argumentos são propostos como pontos de partida para a definição de ambientes:

- a construção de sistemas informatizados dedicados ao ensino necessitam de conteúdos adaptados aos alunos;
- a gestão de interação entre o sistema e o aluno, isto é, a concepção de interface deve ser considerada;
- as tecnologias permitem a construção de um sistema de coleta de dados sobre a aprendizagem e suas condições de aplicação.

Os AIAs podem ser descritos em dois eixos ortogonais: um caracteriza o ambiente informatizado e o outro, as aprendizagens visadas pelo sistema.

O primeiro eixo representa a abertura do sistema informatizado em relação às ações que o sujeito pode realizar sobre o ambiente. O segundo eixo representa o tipo de conhecimento sobre o qual se prevê que o software possa contribuir para a aprendizagem. Mendelsohn (1990) afirma que em uma das extremidades (eixo 1) se encontram as atitudes do metac conhecimento (planificação da ação e heurística de resolução de problemas) e sob o eixo 2, os sistemas especializados construídos para permitir a aquisição de um conhecimento bem definido. Este conhecimento tem a propriedade de ser dependente de um contexto específico.

O produto cartesiano destas duas dimensões delimita quatro zonas distintas nas quais os principais ambientes de aprendizagem disponíveis no momento podem ser explicitados na figura 4.

Os softwares aplicativos (editores de texto, planilhas, base de dados e software gráfico) podem ser considerados como abertos do ponto de vista da informática. Oferecem a possibilidade de construção de macrocomandos similares aos procedimentos de linguagem de programação clássica. Sendo software especializados, destinam-se a

uma atividade com conteúdo preciso como esquematizar, classificar objetos ou resolução de problemas numéricos.

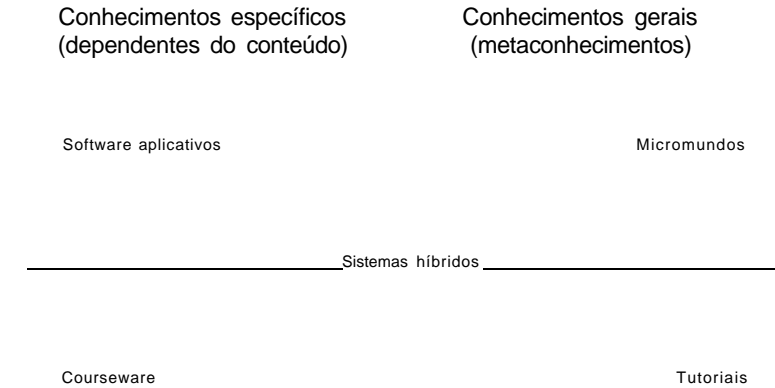
Os micromundos são sistemas informatizados, onde o aluno deve explorar um domínio com um mínimo de ajuda do sistema, combinado às primitivas de uma linguagem (exemplo: LOGO). O objetivo deste ambiente é ambicioso, afirma Mendelsohn (1990), pois o aluno aprende a aprender utilizando o ambiente para espelhar seus conhecimentos e construir novos objetos. A concepção destes sistemas é similar à concepção de uma linguagem de programação, sendo do tipo construtivista.

Os *coursewares* são software educacionais clássicos que, a partir de uma situação interativa entre o aluno e um problema, leva o aluno a resolvê-la. A gama de atividades possíveis é vasta, mas cada seqüência é fechada por respostas interpretáveis pelo programa. São considerados como parte de um ambiente que favorece pouco a iniciativa do aluno e são muito especializados em relação aos objetivos pedagógicos. A concepção deste ambiente repousa no diálogo interativo e a aprendizagem consiste em sujeito (aluno) realizar a seqüência de procedimentos associados a determinados conceitos.

Os tutoriais são como alguns *coursewares*, onde a característica de resolução de problemas se acrescentou o componente tutorial, onde são representados o modelo do aluno, o conhecimento e a técnica do professor e a especialização do conhecimento a ser ensinado. A idéia é permitir aprendizagem de alto nível, a lógica e a compreensão, através da tutoria entre o sistema de professor e o sistema do aluno. A concepção destes sistemas é análoga às ajudas *on line* disponíveis, por exemplo, nos aplicativos.

Embora numerosos trabalhos a respeito de modalidades de uso da informática na educação tenham sido apresentados, a categorização das modalidades vem sendo definida a partir de uma abordagem sobre conteúdos sistematizados e da noção de processo de construção de modelos. Pesquisas apontam para vantagens ou problemas que uma ou outra abordagem possui, percebe-se que os conteúdos não são

dicotomizados do processo de raciocínio e, portanto, as abordagens não devem tender ao reducionismo.



**FIG.4 - Taxionomia dos Ambientes de Aprendizagem
(Mendelsohn, 1990)**

Assim sendo, as formas de utilização dos computadores na educação trazem como fundamentação uma filosofia educacional e uma teoria de aprendizagem (Campos e Santos, 1990). Kemmis (in Underwood & Underwood, 1990) descreve quatro paradigmas educacionais dentro dos quais o software educacional pode ser inserido: instrucional, revelatório, conjectural e emancipatório. Segundo os autores, estes paradigmas podem ser compreendidos da seguinte forma:

- paradigma instrucional, incluindo instrução programada e exercício e prática;
- paradigma revelatório, no qual o aluno faz descobertas usando simulações;

- paradigma conjectural, com o computador sendo usado para construção e avaliação de modelos;
- paradigma emancipatório, no qual o computador é usado como ferramenta para manipulação de números ou textos ou para tratamento e recuperação da informação, liberando o usuário para concentrar-se no processo de aprendizagem.

Relevância dos Fatores de Qualidade no Software Educacional

Os fatores de qualidade do software educacional têm 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 como Lathrop e Goodson (1983) deixam claro que os professores não precisam se tornar programadores nem especialistas em ciência da computação, e Merrill et al. (1986) têm 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 (1983), EPIE Institute (1982), MECC (1982), Scholastic (1982) e especialistas como Dassance (1986), Oliveira et al. (1986) vêm 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 (1988) "é necessário que fatores de qualidade para o desenvolvimento do produto sejam definidos e utilizados".

Este trabalho partiu dos fatores e subfatores selecionados por Stahl (1988) e acrescentou novos fatores e subfatores, definindo critérios e processos de avaliação. A figura 6 dá visão geral dos objetivos, fatores e subfatores que identificamos como relevantes para a avaliação do software educacional.

Quadro 2 - Objetivos, Fatores e Subfatores para a Avaliação do software Educacional

Objetivo	Fator	Subfator
Utilizabilidade	Operacionalidade	Amenidade ao Uso
		Eficiência do Processamento Eficiência do Desenvolvimento
	Manutenibilidade	Alterabilidade
		Independência do Ambiente
	Rentabilidade	Validabilidade Adequação Integração
	Confiabilidade Conceitual	Integridade Fidedignidade
Clareza		
Confiabilidade da Representação	Legibilidade	

Alguns subfatores foram especificados para o software educacional, além dos que já estavam definidos pelo método para avaliação da qualidade de software proposta por Rocha (1987). Estes subfatores estão definidos no Quadro 3.

Quadro 3 - Definição de Subfatores Acrescidos ao Quadro 2

Subfatores	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 de o 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 com outros recursos ou materiais instrucionais

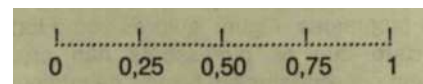
Manual para Controle da Qualidade do Software Educacional

Para tornar viável a avaliação de produtos de software educacional foi elaborado um manual para controle da qualidade (Campos e Rocha, 1990).

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

Foram selecionados dois tipos de processos de avaliação:

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.



A medida é do tipo binário, apresentada da seguinte forma:

() sim () não

No primeiro caso, interpreta-se o resultado da medida a partir do valor assinalado na escala. No caso de mais de um critério por fator, recomenda-se a média aritmética. Pensa-se em trabalhar no futuro com a média ponderada uma vez que se deve estabelecer pesos para cada critério. Para chegarmos a estes pesos, está sendo realizado uma série de trabalhos cujo objetivo é determinar, em cada caso, a relevância de um determinado critério.

O ponto de corte definido para qualidade considerada satisfatória foi retirado de estudos desenvolvidos pelo "Military Management Comand, Systems Management Division" (Cardoso, 1990) 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

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.

Os quadros 4, 5 e 6 mostram exemplos de definição de critérios com os respectivos processos de avaliação.

Quadro 4 - Exemplo de Definição de Critério com o Processo de Avaliação

Critério: Fornecimento de <i>Feedback</i>	Objetivo: Utilizabilidade Fator: Operacionalidade Subfator: Amabilidade ao uso
---	--

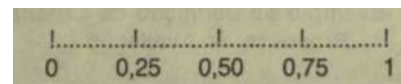
Definição: é a possibilidade interna ao programa de o usuário analisar suas respostas, indicando correção ou erro, além de fornecer o caminho ou a indicação da resposta correta.

Discussão: o fornecimento de *feedback* ou realimentação ao aluno-usuário é fundamental em um software educacional. O programa deve ser preparado para analisar as respostas dadas pelo usuário, compará-las com as respostas desejadas e discriminá-las. Caso a resposta não seja a desejada, o programa deverá escolher uma seqüência remediativa que dará a oportunidade ao aluno de responder novamente à questão. A toda resposta inadequada o programa deverá fornecer indicações para a resposta correta.

Processo de avaliação: a avaliação segundo este critério é feita através da resposta ao seguinte indicador:

A resposta errada não tem explicação.

O programa fornece seqüências explicativas para as respostas erradas



Quadro 5 - Exemplo de Definição de Critério com o Processo de Avaliação

Critério: Clareza dos Comandos	Objetivo: Utilizabilidade Fator: Operacionalidade Subfator: Amabilidade de Uso
--------------------------------	--

Definição: é a característica de o programa utilizar ícones, convenções ou pedidos de comando, de forma que estes correspondam às 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 facilita a compreensão do programa.

Processo de avaliação: resposta ao indicador:
 Não há explicação dos comandos Os comandos são explicados claramente



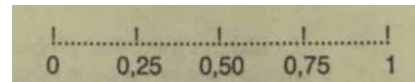
Quadro 6 -Exemplo de Definição de Critério com o Processo de Avaliação

Critério: Adequação do Vocabulário em Nível do Usuário
 Objetivo: Utilizabilidade
 Fator: Operacionalidade
 Subfator: Amabilidade ao Uso

Definição: é a característica de o programa possuir um vocabulário adequado à compreensão da clientela.

Discussão: o programa educacional deve procurar facilitar a comunicação com o usuário, evitando o jargão computacional. Além disso, o programa deve possuir vocabulário compatível com o tipo de usuário ao qual se dirige (nível de escolaridade e faixa etária são, por exemplo, variáveis que devem ser consideradas).

Processo de avaliação: resposta ao indicador:
 O vocabulário utilizado não está adequado ao usuário O vocabulário utilizado está adequado ao usuário



Referências Bibliográficas

BITTER, Gary, CAMUSE, Ruth A. *Using a microcomputer in the classroom*. Virgínia: Reston Publ., 1984.

BEAUFOND, Clifton E.C. *Manual para controle da qualidade de especificações: relatório técnico do Programa de Engenharia de Sistemas e Computação*. Rio de Janeiro: UFRJ, COPPE, 1989.

BORK, Alfred. *Learning with personal computers*. New York: Harper & Row Computer Science Technology, 1987.

CAMPOS, Gilda Helena Bernardino de. *Construção e validação de ficha de avaliação de produtos educacionais para microcomputadores*. Rio de Janeiro, 1989. Dissertação (Mestrado) — Faculdade de Educação, UFRJ.

CAMPOS, Gilda Helena Bernardino de, ROCHA, Ana Regina O da. *Manual para a avaliação da qualidade de software educacional: relatório técnico do Programa de Engenharia de Sistemas e Computação*. Rio de Janeiro: UFRJ, COPPE, 1990.

- CAMPOS, Gilda H.B. de, SANTOS, Neide. Informática na educação: algumas tendências e necessidades para década de 90. In: REUNIÃO ANUAL DA SBPC, 43, 1991. *Anais*. [São Paulo: SBPC, 1991].
- CARDOSO, Rogério Nesi Pereira. *Predição, estimativa e medição da confiabilidade durante o ciclo de vida do software*. Rio de Janeiro, 1990. Dissertação (Mestrado) — Programa de Engenharia de Sistemas e Computação, COPPE, UFRJ.
- CONDUIT. Evaluation questionnaire. In: JONES, Nancy B., VAUGHAN, Larry (Eds.). *Evaluation of educational software: a guide to guides*. Massachusetts: Northeast Regional Exchange, 1983.
- COBURN, Peter et ai. *Practical guide to computers in education*. New York: Addison-Wesley, 1982.
- DUDLEY-MARUNG, Curt, OWSTON, Ronald D. The state of educational software: a criterion based evaluation. *Educational Technology*, v.27, n.3, p.25-29, 1987.
- ENNALS, Richard, GWYN, Rhys, ZDRACHEV, Levcho. *Information technology and education: the changing school*. England: Ellis- Horwood, 1986.
- EPIE Institute. In: JONES, Nancy B., VAUGHAN, Larry (Eds.). *Evaluation of educational software: a guide to guides*. Massachusetts: Northeast Regional Exchange, 1983.
- FREITAS, Afonso C, BARGUT, Maurício F., ROCHA, Ana Regina C. *Características de qualidade de programas: relatório técnico do Programa de Engenharia de Sistemas e Computação*. Rio de Janeiro: UFRJ, COPPE, 1985.
- JOLICOEUR, Karen, LEGER, Dale E. Do we really know what makes educational software effective? A call for empirical research on effectiveness. *Educational Technology*, v.26, n.12, p.7-11, 1986.
- KNEZEK, Gerald A., RACHLIN, Sidney L, SCANNEL, Peter. A taxonomy for educational computing. *Educational Technology*, v.28, n.3, p.15-19, 1988.
- LATHROP, Ann, GOODSON, Bobby. *Courseware in the classroom: selecting, organizing and using educational software*. Califórnia: Addison-Wesley, 1983.
- MANN, Tom, COLEMAN, Michael. *Software quality assurance*. London: Macmillan Education, 1988.
- MENDELSON, Patrick. *Uordinateur dans l'enseignement*. Colloque "Enseignement et apprentissage avec l'ordinateur". Martigny: Université de Genève, 1989. (TECFA document 90-6).
- MERRILL, Paul F., TOLMAN, Christensen, HAMMONS, VICENT, Reynolds. *Computers in education*. New Jersey: Prentice-Hall, 1986.
- MINNESOTA Education Computing Consortium — MECC. In: JONES, Nancy B., VAUGHAN, Larry (Eds.). *Evaluation of educational software: a guide to guides*. Massachusetts: Northeast Regional Exchange, 1983.
- OLIVEIRA, Celina O, MENEZES, Eliane, MOREIRA, Mercia. *Avaliação de software educacional*. Documento base para discussão apresentado na XVIII Reunião da ABT. Belo Horizonte, 1986.
- PRESSMAN, Roger S. *Software engineering: a practitioner's approach*. New York: Macgraw-Hill, 1987.
- RAGSDALE, Ronald G. *Evaluation of microcomputer courseware*. [S.I.]: Ontario Institute for Studies in Education, 1982.
- ROCHA, Ana Regina C. *Análise e projeto estruturado de sistemas*. Rio de Janeiro: Campus, 1987.

ROCHA, Ana Regina O, PASSOS, Maria Cristina J. Fonseca. *Critérios para avaliação de software para pecuária do leite: relatório técnico do Programa de Engenharia de Sistemas e Computação*. Rio de Janeiro: UFRJ, COPPE, 1990.

ROCHA, A.R.C. da, SEREZA, J.M. *TABA: a heuristic workstation for software development* — COMPEURO 90. Tel Aviv, 1990.

STAHL, Marimar M. *Avaliação da qualidade de software educacional: relatório técnico do Programa de Engenharia de Sistemas e Computação*. Rio de Janeiro: UFRJ, COPPE, 1988.

SCHOLASTIC. In: POIROT, James, BILLINGS, Karen. *Microcomputers in education: a scholastic in-service training program*. New York: Scholastic Book Services, 1982.

SCHAEFERMEYER, Shanna. Standards for instructional computing software design and development. *Educational Technology*, New Jersey, v.30, n.5, p.9-15, 1990.

UNDERWOOD, Jean O.M., UNDERWOOD, Geoffrey. *Computers and learning: helping children acquire thinking skills*. Massachusetts: Basil Blackwell, 1990.