

Análise de construção de *software* educativo com qualidade: Sugestão de ficha para registro e avaliação de *software* educativo

Eliane Cristina Amaral
*Mestre em Semiótica, Tecnologias de
Informação e Educação pela
Universidade Braz Cubas – Mogi das
Cruzes - SP*
e-mail: elianeamaral@terra.com.br
eliane@lac.inpe.br

Ulisses Thadeu Vieira Guedes
*Doutor em Ciência Espacial e Orbital
pelo Instituto Nacional de Pesquisas
Espaciais – São José dos Campos – SP*
e-mail: utvg@dem.inpe.br

Resumo

Pretendemos analisar os critérios para avaliação de um software com qualidade para o uso educacional. Nesse sentido, uma avaliação criteriosa pode contribuir e apresentaremos uma sugestão para registro desta avaliação. A análise qualitativamente de um software educativo, quanto ao critério de engenharia de software e uma avaliação para contribuir com o aprendiz a construir seu conhecimento e a modificar sua compreensão de mundo, elevando sua capacidade de participar da realidade que está vivendo. Poderá ser avaliada pelo formato da tela, pela navegabilidade, usabilidade, pela interação ou interação instrutiva com o usuário.

Palavras-chaves: *software educacional, engenharia de software, tutores inteligentes, interatividade, interação instrutiva.*

1. Introdução

A utilização da informática no ensino acrescentará uma ótica transformadora na formação das novas gerações. O computador aparece como um instrumento de ensino e aprendizagem. Outros autores também mencionam que os computadores são as melhores ferramentas utilizadas na educação, principalmente em virtude da possibilidade do uso dos recursos audiovisuais.

Temos o objetivo de estudar qualitativamente *software* educativo e analisar os *tutores* inteligentes neste tipo de mídia educacional.

Abordamos o desenvolvimento de um *software*, a partir da observação de critérios de qualidade e normas técnicas, e caminhamos para a análise e construção de *software* educativo com qualidade, considerando tanto o aspecto educacional quanto os critérios de engenharia de *software*. Também abordamos a metodologia e análise do *software*.

2. Metodologia

Pesquisa de caráter fenomenológica, que serviu para que fosse possível apontar alguns caminhos para se ter um referencial de análise de *software* educacional.

Para avaliar sua qualidade sob a perspectiva de suas possíveis adequações, fiz um breve panorama sobre a normatização da qualidade dos produtos e as possíveis preocupações sobre o assunto.

O *software* educacional deve ser escolhido e elaborado de acordo com as teorias de aprendizagem que diferenciavam cada ambiente educacional. Assim, temos ambientes educacionais mais ou menos interativos, que exigem maior ou menor grau de participação dos aprendizes e um controle maior ou menor no aluno no processo de construção do conhecimento.

3. Interação entre aprendizagem e desenvolvimento do *Software* Educativo

A interatividade pode ser de três tipos:
- alta interatividade: que permite a descoberta imprevista e a descoberta de exploração livre;

- média interatividade: que permite a descoberta guiada;

- baixa interatividade: que privilegia a aprendizagem de recepção direcionada, a exposição indutiva e a exposição dedutiva.

Levamos em consideração alguns pontos apontados na ISO/IEC 9.126, na série ISO/IEC 14.598, na norma ISO/IEC 12.119.

Como o computador está inserido no processo de ensino-aprendizagem por meio da sua integração na escola, desde sua utilização como recurso de ensino-aprendizagem. No início de 80, o número de pessoas que dedicavam uma parte significativa do seu tempo profissional aos computadores e à Educação disparou de uns cem para dezenas de milhares. O livro de Ed Taylor (professor no *Columbia Teachers's College* e criador do primeiro programa de mestrado em computadores e Educação) era chamado *Computers in schools: tutor, tutee, tool*. O sentido atribuído ao primeiro e último termos do subtítulo corresponde suficientemente, na acepção popular, ao que os computadores podem fazer em Educação. Se acrescentarmos a isso a possibilidade de conexão à Internet, constatamos que a informática é a mais poderosa ferramenta para uso em educação, em virtude de sua capacidade e da variedade de recursos e serviços que oferece.

Pelo critério de instrução, o *software* deve ser avaliado por aspectos pedagógicos, técnicas de ensino ou estratégias de aprendizado. O primeiro passo é escolher a estratégia de aprendizado; o segundo é determinar se os objetivos do *software* coincidem com os do plano curricular; o terceiro é estabelecer se a aplicação é adequada para os resultados de aprendizagem, e o quarto é especificar se o *software* incorpora a estratégia de aprendizado.

Quanto ao critério de apresentação, o *software* poderá ser avaliado pelo formato na tela, pela navegabilidade, usabilidade, pela interação instrutiva com o usuário.

A interação instrutiva compreende um conjunto de técnicas baseadas no modelo “*learning-by-doing*” (aprender-fazendo), o qual está baseado em três princípios: exploração das habilidades (*explorability*), a previsibilidade (*predictability*) e orientação (*guidance*). As técnicas da interação instrutiva são baseadas nas formas como o ser humano aprende a desenvolver tarefas. As pessoas aprendem ouvindo, vendo ou fazendo. O aprender-fazendo é a mais eficiente técnica. O aperfeiçoamento vem através de uma constante repetição.

No entanto, o computador pode enriquecer ambientes de aprendizagem onde o aluno,

interagindo com os objetos desse ambiente, tem chance de construir o seu conhecimento. Sendo assim, o Tutor inteligente é um *software* educacional que vai servir de guia no processo de aprendizagem do aluno.

3.1 Desenvolvimento de *software* visando a qualidade

Os processos de desenvolvimento de *software*, visando à qualidade na sua construção, desde o processo do ciclo de vida aos diversos processos de documentação, garantia da qualidade, testes, revisões, tipos de defeitos, melhoria de processos e algumas categorias de *software*. Nossa argumentação está construída a partir do tópico *software* educativo, suas características em relação à qualidade, visando os aspectos técnicos e uma sugestão de ficha para registro do *software* educativo.

Muita gente associa o termo *software* aos programas de computador. Na verdade, essa é uma visão muito restrita. *Software* não é apenas o programa, mas também toda a documentação associada e os dados de configuração necessários para fazer com que programas operem corretamente. *Human-friendly* o *software* tem que ser amigável.

A exigência cada vez maior dos usuários quanto à qualidade, facilidade de uso, segurança e confiabilidade de dados, e com relação ao baixo custo fez com que surgisse, a partir de 1970, a Engenharia de *Software*. A avaliação de um *software* educativo é um processo que deve envolver diversos e diferentes tipos de profissionais, tanto da área de informática como da área pedagógica.

A qualidade do *software* pode ser visto como um conjunto de características que devem ser alcançadas em um determinado grau para que o produto atenda às necessidades de seus usuários. Para proceder à avaliação do *software*, tendo em vista a apreciação de seus educadores, usuários e especialistas, capazes de, num esforço multidisciplinar, avaliar a contribuição daquele produto levando em consideração o seu próprio projeto pedagógico.

A norma ISO/IEC 9126 apresenta um conjunto de características de qualidade aplicável a qualquer produto de *software*. Para se alcançar qualidade, produtividade da equipe e redução de custo, faz-se necessário implementar um processo de desenvolvimento definido e institucionalizado.

Um processo de desenvolvimento de *software* está presente na norma internacional NBR ISO/IEC 12207 – Tecnologia da Informação – Processos de Ciclo de Vida de *Software* (ISO,

1995 a).

Os programas e aplicativos devem passar por uma avaliação criteriosa antes de serem colocados a serviço do modelo pedagógico adotado, sua análise deve partir, também, das concepções do homem, de mundo e de um modelo epistemológico e psicológico coerente.

Na norma ISO/IEC 12207, os processos que envolvem o ciclo de vida do *software* são agrupados em três classes que representam sua natureza.

Já se sabe que, quando o *software* desenvolvido utiliza ferramentas adequadas para o seu desenvolvimento, têm seu custo diminuído a manutenção normalmente de um *software* é alta.

A documentação é um processo muito importante no desenvolvimento do *Software*. Estima-se que cerca de 20% a 30% de todo o tempo empregado no desenvolvimento de um *software* é dedicado à elaboração de sua documentação.

O processo de garantia da qualidade da norma ISO/IEC 12207 serve para garantir que os processos e produtos de *software*, no ciclo de vida do projeto, estejam em conformidade com os requisitos especificados e referentes aos planos estabelecidos.

Esse processo evidencia a relação entre a qualidade do produto e a qualidade do processo de *software* utilizado em sua construção, bem como relaciona essa norma com a ISO 9000.

O teste de *software* também está inserido na norma ISO/IEC 12207.

Antes de ser liberado para o mercado, o *software* deve ser revisado, pois se deve verificar se o produto possui a qualidade de que o usuário necessita. As revisões de *software* são atividades que garantem a qualidade.

A melhoria do processo se faz necessária, com vistas ao aperfeiçoamento e desenvolvimento do *software* para a obtenção de produtos com os níveis desejáveis de qualidade. Várias pesquisas sobre processo de desenvolvimento e várias normas e modelos de qualidade, visam melhorar as etapas do ciclo de desenvolvimento. Entretanto, definir, coletar e analisar um conjunto de métricas não é uma tarefa trivial.

As métricas foram selecionadas por meio de três tipos de processos: métricas de tempo, métricas de utilização de recursos e métricas de eventos. Para caracterizar o projeto e seu contexto - medir, avaliar e sugerir melhorias em um processo de *software* específico: tempo, precisão da estimativa de cronograma, esforço, precisão da estimativa de esforço, tamanho do sistema, número de erros, números de modificações, densidade de defeitos, rotatividade do pessoal, produtividade, deterioração do *software* e

experiência da equipe.

À medida que o desenvolvimento de *software* se torna mais competitivo, a melhoria da qualidade dos produtos de *software* passa a não ser apenas um diferencial para as empresas, mas um fator crítico para a sobrevivência das mesmas. Dessa forma, encontrar uma abordagem que garanta a melhoria do processo de produção de *software* tem sido uma preocupação muito importante. A norma ISO/IEC 9126 apresenta um conjunto de características de qualidade aplicável a qualquer produto de *software*.

3.2. *Software* Educativo – SE

Nas literaturas especializadas definem o grau de iniciativa que permitem ao aluno interagir ou o grau de direcionamento conferido a ele. Classificam o grau de interatividade do *software* da seguinte maneira: de alta interatividade: - que permite a descoberta imprevista e a descoberta de exploração livre; de média interatividade: - que permite a descoberta guiada; de baixa interatividade: - que privilegia a aprendizagem de recepção direcionada, a exposição indutiva e a exposição dedutiva.

3.2.1 Características da qualidade do *software* educacional

A qualidade do processo de desenvolvimento de um *software* educacional é essencial para a obtenção da qualidade do produto, que não foge à regra e nem da concorrência do mercado interno e externo de qualquer outra produção de *software* que deve atrair o interesse do consumidor. A qualidade do produto é tratada, entre outras, na série de normas ISO/IEC 9126, na série ISO/IEC 14598, e na norma ISO/IEC 12119. Esta última enfoca os requisitos de qualidade de pacotes de *software*.

O *software* apresenta característica que favorecem a atuação do professor, como no caso da programação; outros, em que certas características não estão presentes e requer um maior envolvimento do professor para auxiliar o aluno a aprender, como no caso do tutorial.

Para avaliarmos o *software* educacional é importante considerar as seguintes características:

- características pedagógicas: ambiente educacional, pertinência ao programa curricular, aspectos didáticos;
- facilidade de uso: facilidade de aprendizado, facilidade de memorização, robustez;
- características da interface: condução, afetividade, consistência, significado dos códigos e denominações, gestão de erros;

- adaptabilidade: personalização, adequação ao ambiente;
- documentação: “*help on-line*”, documentação do usuário;
- portabilidade: adequação tecnológica, adequação aos recursos da escola;
- retorno do investimento: preço compatível, taxa de retorno;
- outros aspectos: preço acessível, disponibilidade no mercado, recomendações de outros usuários, possibilidade de obtenções de cópias, convênios e análises de demos.

O *software* educacional pode ser classificado em grandes grupos, tais como: tutoriais, programação, exercitação, investigação, simulação e modelagem, programação, aplicativos, multimídia e internet, jogos.

Quanto ao nível educacional, o *software* é assim classificado:

seqüencial: - o aprendiz memoriza o conteúdo apresentado e repete quando necessário, torna-se passivo;

relacional: - o aprendiz adquirir e habilidades e faz relação com outros fatos e outras fontes de informação, enquanto que a interação fica entre o aprendiz e a tecnologia, de modo a isolá-lo;

criativo: - cria a interação entre pessoas e tecnologias, favorecendo o aprendizado participativo.

O *software* educacional apresenta características que podem favorecer o processo de construção do conhecimento, mas devem ser analisados antes de serem colocados em uso. O *software* educacional tem necessidades implícitas e explícitas como qualquer outro produto.

A qualidade está diretamente ligada à satisfação do cliente ou do usuário, e por isso, deverão ser observados os seguintes aspectos nos *software*: mídias empregadas, qualidades de telas, interfaces disponíveis, clareza de instruções, compartilhamento em rede local e Internet, compatibilização com outros *software*, hardware e funcionalidade em rede (importação e exportação de objetos), apresentação auto-executável, recursos hipertexto e hiperlink, disponibilidade de help-desk, manual técnico com linguagem apropriada ao professor usuário, facilidade de instalação, desinstalação e manuseio, etc.

3.3. Sugestão de ficha de registro de um *software* educativo

A- Identificação do Produto

Designação do	
---------------	--

Empresa	
Autor(es)	
Versão	
Data de produção	
Data da aquisição	
Data de validade	
Suporte	
Ambiente ou sistema operativo	
Procedimento de instalação	
Manual de utilização	
Idioma(s)	
Preço	
Faixa etária	
Observação	

IDENTIFICAÇÃO DO *SOFTWARE*

1- ARMAZENAMENTO

Preencha com um X, caso afirmativo	Tipo de disco	Quantidade
	Disquete de 5 ¼"	
	Disquete de 3 ½"	
	CD-Rom	
	ZIP	
	Outros	

2- CONFIGURAÇÃO DO EQUIPAMENTO

Modelo mínimo necessário do computador:

() 486 ou similar

() Pentium _____ ou similar

() outro _____

2.2.1- Memória RAM: _____ MB (megabyte)

2.2.2- Espaço necessário em disco:

2.2.3- Tipo de vídeo:

2.2.4- Tipo de som:

2.2.5- Alguns outros equipamentos: _____

3- Classificação

Quanto ao tipo do *software*:

	Tutorial		Investigação
			Exercitação
	Simulação e Modelagem		Aberto
			Fechado
	Programação		
	Aplicativo		Editor de texto
			Planilha eletrônica
			Bancos de dados
			Gráficos
			Apresentações
			Auditoria
			Multimídia e Internet
	Jogos		

4- BASE PEDAGÓGICA:

Concepção Teórica de Aprendizagem:

Construtivista: _____

Behaviorista: _____

Outra: _____

5- Como o *software* possibilita a realização do ciclo descrição-execução-reflexão-depuração-descrição:

Onde :

0 = ruim 1=fraco 2= satisfatório 3= bom 4 = ótimo

Descrição: 0 1 2 3 4

Execução: 0 1 2 3 4

Reflexão: 0 1 2 3 4

Depuração: 0 1 2 3 4

6- O *software* propicia a interação entre:

Aprendiz x Agente de Aprendizagem: 0 1 2 3 4

Aprendiz x Agente de Aprendizagem x Grupo:

0 1 2 3 4

Aprendiz x Máquina: 0 1 2 3 4

7- O *software* tem processos de avaliação, ou seja, de que forma será feito o "feedback" ao aluno?

0 1 2 3 4

8- Em relação ao processo de construção do conhecimento do aluno:

- Apresenta múltiplos caminhos para a solução do problema?

0 1 2 3 4

- De que forma possibilita a formulação e verificação de hipóteses, a análise e depuração dos resultados?

0 1 2 3 4

9- Possibilita a integração de diferentes disciplinas?

Sim () Não ()

10-Em que nível de integração?

0 1 2 3 4

11-Quais disciplinas? _____

12- São adequados:

Telas - 0 1 2 3 4

Gráficos - 0 1 2 3 4

Sons - 0 1 2 3 4

Imagens - 0 1 2 3 4

Textos - 0 1 2 3 4

13- Tempo sugerido para utilização: _____

14- Público Alvo: (faixa etária, escolaridade, outras informações)

15- Quanto ao nível de aprendizado:

Sequencial	Relacional	Criativo
------------	------------	----------

16- ASPECTOS TÉCNICOS:

sim	não	Apresenta as instruções de forma clara
sim	não	Indica as possibilidades de uso
sim	não	Especifica os requisitos de <i>hardware/software</i>
sim	não	Facilidade de instalação e desinstalação
sim	não	Fornecer manual de utilização com linguagem apropriada
sim	não	É compatível com outro <i>software</i>
sim	não	É compatível com outro <i>hardware</i>
sim	não	Funciona em rede
sim	não	Importa e exporta objetos
sim	não	É auto-executável
sim	não	Possui recursos de hipertexto e hiperlink
sim	não	Dispõe de help-desk
sim	não	Apresenta facilidade de navegação

17- CONCLUSÕES:

Processo de Avaliação: _____

Conclusões / Recomendações / Sugestões:

Equipe avaliadora: _____

4. Considerações Finais

É preciso adotar um posicionamento crítico face a qualquer inovação tecnológica, o que inclui o computador. Para contribuir com a

avaliação de *software* educativo e tutores inteligentes, principalmente, leve os professores, pais, desenvolvedores de *softwares* educativos ou responsáveis pela educação de crianças uma reflexão sobre a prática pedagógica e lhes propicie a visão de que os princípios interacionistas (instrução interativa). Sob essa ótica de avaliar um *software* para o uso educativo exige muito mais do que conhecimento sobre informática instrumental, conceitos e aplicações de engenharia de *software*, exige a construção de conhecimentos sobre as teorias de aprendizagens,

concepções educacionais e práticas pedagógicas, técnicas computacionais e reflexões sobre o papel do computador e do aluno nesse contexto, pois a construção do conhecimento do aprendiz não é processo simples e imediato, mas um caminho árduo e longo. A proposta do Programa Nacional de Informática Educativa do MEC é utilizar o computador na escola com o objetivo de criar um ambiente de aprendizagem onde o aprendiz processe a informação, agregue-a a seus esquemas mentais e coloque-a para funcionar mediante um desafio ou situação problema.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, F. J., **Educação e Informática os comportamentos na escola**. São Paulo: Cortez, 2000.

ALMEIDA, Maria Elizabeth de, **Proinfo: Informática e formação de professores**, Secretaria de Educação a Distância, Brasília - DF, Seed, 2000.

ASSIS, M., **A educação e a formação profissional na encruzilhada das velhas e novas tecnologias**. In: FERRETTI, Celso J. et al. (orgs.). **Tecnologias, trabalho e educação: Um debate multidisciplinar**. Petrópolis – Rio de Janeiro: Vozes, pp. 189-203, 1994.

CASTORINA, José A., FERREIRO, Emília, LERNER, Delia, OLIVEIRA, Marta Kohl, **Piaget – Vygotsky: Novas contribuições para o debate**, tradução: Cláudia Schiling, São Paulo: 4ª ed., Ática, 1997.

CHERMANN, Maurício; BONINI, Luci Mendes, **Educação a Distância: novas tecnologias em ambientes de aprendizagem pela internet**, Mogi das Cruzes- São Paulo, Universidade Braz Cubas, 2000.

CONSTANTINE, Larry. L.; LOCKWOOD, Lucy A. D.,(2000) **Instructive Interaction: Making Innovative Interfaces Self-Teaching**, University of Technology, Sydney, MA: Addison-Wesley. [http:// www.forUse.com](http://www.forUse.com) (20/11/2002).

CONSTANTINE, Larry. L.; LOCKWOOD, Lucy A. D.,(2002), **Usage-Centered Engineering for Web Applications**, *IEEE Software*, March/April

HALL, P. e WOOD, P. **Intelligent tutoring systems: a review for beginners**. *Canadian Journal of Educational Communication*, 19 (2), p. 107-123, 1990.

OLIVEIRA, Celina C., COSTA, José W., MOREIRA, Mercia, **Ambientes Informatizados de Aprendizagem: Produção e avaliação de software educativo**. Campinas – São Paulo: Papyrus, 2001.

PANQUEVA, Álvaro H. Galvis, **Software Educativo Multimídia aspectos críticos no seu ciclo de vida**, Santafé de Bogotá, DC 1, Colômbia, Sur América. [http:// www.inf.ufsc.br/sbc-ie/revista/nrl/galvis-p.html](http://www.inf.ufsc.br/sbc-ie/revista/nrl/galvis-p.html) (10/08/1999)

PARNAS, D.L.; MADEY, J. e IGLEWSKI, M., **Precise documentation of well-structured programas**. *IEEE Transactions on Software Engineering*, v.20, n. 12, 1994.

POLSON, M. C., RICHARDSON, J.J., **Foundations of Intelligent Tutoring Systems**, New Jersey, Lawrence Erlbaum Associates, Inc, 1988.

PRESSMAN, Roger S., **Engenharia de Software**, São Paulo, Makron Books, 1995.

PFROMM NETTO, Samuel, **Telas que ensinam. Mídia e aprendizagem: do cinema ao computador**, Campinas – São Paulo, 2ª edição, Editora Alínea, 2001.

ROCHA, Ana Regina C.; MALDONADO, José C.; WEBER, Kival C., **Qualidade de Software: teoria e prática**, São Paulo, Prentice Hall, 2001.

SOMMERVILLE, Ian, **Engenharia de Software**, tradução: Mauricio Andrade, São Paulo: Addison Wesley, 2003.

VALENTE, José Armando, **Computadores e conhecimentos: repensando a educação**. Campinas – São Paulo: Unicamp/Nied, 1998.

VIEIRA, Fábila M. S., **Avaliação de Software Educativo: Reflexões para uma análise criteriosa**, In: Valente, José A., **Análise dos diferentes tipos de Software usados na Educação** - Unicamp/Nied, 2002.

.....

.....