



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I - CAMPINA GRANDE-PB
CENTRO DE CIÊNCIA E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE LICENCIATURA PLENA EM COMPUTAÇÃO**

JEFFERSON FELIPE SILVA DE LIMA

QUALI-EDU: Um processo de Avaliação da Qualidade do Produto de Software Educacional

CAMPINA GRANDE – PB
2014

JEFFERSON FELIPE SILVA DE LIMA

QUALI-EDU: Um processo de avaliação da qualidade do produto de software educacional

Monografia apresentada ao Curso de Licenciatura Plena em Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de graduado.

Orientadora: Prof^a Dr^a. Luciana de Queiroz Leal Gomes

CAMPINA GRANDE – PB
2014

É expressamente proibida a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano da dissertação.

L732q Lima, Jefferson Felipe Silva de.

Quali-edu [manuscrito] : um processo de avaliação da qualidade do produto de software educacional / Jefferson Felipe Silva de Lima. - 2014.

82 p.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2014.

"Orientação: Profa. Dra. Luciana de Queiroz Leal Gomes, Departamento de Computação".

1. Software Educacional. 2. Qualidade de Software. 3. Modelos de Qualidade de Software. I. Título.


21. ed. CDD 371.33


JEFFERSON FELIPE SILVA DE LIMA


QUALI-EDU: Um processo de avaliação da qualidade do produto de software educacional

Monografia apresentada ao Curso de Licenciatura Plena em Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de graduado.

Aprovada em 14/03/2014.


Prof^ª Dr^ª. Luciana de Queiroz Leal Gomes / UEPB
Orientadora


Prof. Ma. Ana Isabella Muniz Leite/ UEPB
Examinador


Prof^ª Me. Edson Holanda Cavalcante Júnior/ UEPB
Examinadora

DEDICATÓRIA

À Maria de Fátima Silva (Mainha), “Dona Zefinha e Seu Zé Henrique” (Mãe e Pai), responsáveis pelo que fiz, e pelo que sou. DEDICO.

AGRADECIMENTOS

À minha bela flor, Maria de Fátima Silva, minha mãe, amiga, consultora, escudeira e incentivadora e até ditadora quando necessário, minha eterna gratidão.

À minha irmã Joyce Francine Silva de Lima, segunda mãe, e portadora de todos os melhores adjetivos possíveis.

Aos meus avós maternos José Henrique da Silva, e minha avó Josefa Maria Silva, pela compreensão ante a minha ausência nos encontros familiares, e pela pureza nos simples gesto de amar e apoiar-me nessa caminhada.

À todos os meus Tios e Tias e ao meu Pai, por sempre acreditar no meu potencial, e claro, sempre me incentivar à rumar pelo viés da educação.

À Mariana Neves Nóbrega Torres, pelo amor, força, apoio e compreensão durante a minha graduação.

À minha “orientadora/mãe” Luciana de Queiroz Leal Gomes, pela compreensão, apoio e constante incentivo durante todas as nossas pesquisas.

À toda a turma de Licenciatura Plena em Computação, em especial Renato Celestino Marques (*in memoriam*), por terem me recebido de sempre braços abertos, sempre durante essa jornada “num diploma de sofrer de outra universidade”...

À toda a equipe da Secretaria Municipal de Educação, em especial Betânia Macêdo, Maria Sandra, José Antônio e José Nilton (*in memoriam*), pela compreensão e constante apoio sempre quando precisei me ausentar para cumprir com os deveres acadêmicos.

Ao amigo Edivaldo Silva de Brito (Nenem do Ônibus), por me transportar diariamente durante esses anos de curso.

Aos meus amigos Michael Ohler, Gerhard di Bernardo, Nicolai Grospietsch, pelo grande incentivo dado durante toda a minha vida.

Aos meus amigos/irmãos Anderson Barros, Franklin Paulino, José Luiz Paulino, Amaury Aragão, Victor Barros, por sempre me proverem momentos de distração e apoio.

Aos meus amigos de laboratório Maelson Marques e Alanna Coelho, pelas experiências trocadas durante a pesquisa e claro, as risadas e descontrações nos “cafezinhos”.

“Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível.”

Charles Chaplin

RESUMO

Atualmente, no Brasil, por meio do programa PROINFO proveniente do Ministério da Educação, é possível observar a crescente aparição de laboratórios de informática nas escolas públicas por todo território. Dos grandes centros, até os lugares mais remotos, o acesso à informática vem tentando conseguir seu espaço, como ferramenta nos processos de ensino-aprendizagem. Mediante pesquisa feita em trabalhos acadêmicos e científicos, observou-se que na maioria dos trabalhos que tratavam do desenvolvimento de Software Educacional, a avaliação da qualidade não era considerada nestes. Diante deste fato, surgem as indagações: Os softwares que são disponibilizados nos laboratórios de informática das escolas são detentores de qualidade suficiente e garantida para a utilização por parte do alunado? Quais foram os modelos utilizados para a avaliação destes? Nos trabalhos pesquisados, foi observada também a presença (na grande maioria destes) das normas ISO/IEC 9126 e ISO/IEC 14598, como guias para o alcance da qualidade. Porém, nas situações encontradas, a presença do usuário final do Software Educacional (quais sejam, o professor e o aluno) foi descartada. Além disso, aspectos de qualidade relacionados ao uso desses softwares no ensino-aprendizagem também foram omitidos. Diante do exposto, este trabalho apresenta um modelo de processo de avaliação de qualidade do produto de Software Educacional, considerando aspectos educacionais na sua concepção, e que leva em conta um maior envolvimento do aluno e do professor na avaliação desse software.

PALAVRAS-CHAVE: Software Educacional, Qualidade de Software, Modelos de Qualidade de Software.

A B S T R A C T

Currently, in Brazil, by the program PROINFO, from the FederalA Education Department, it is possible to observe the increase of informatics laboratories at the public school spreading by all territory. From the big cities, until the remotest places, the access to informatics is trying to get its space, as a tool in the teaching-learning processes. Based on researches made in academic and scientific works, it was possible to observe that most of those works that were about Educational Software Development, the quality evaluation weren't considered on them. Based on this fact, it came the inquiries: The software that are available on the public schools' informatics laboratories have enough and guaranteed quality for the students' utilization? Which were the models used to its evaluation? At the researched works, it was observed also the presence (in the most of the works) the rules ISO/IEC 9126 and ISO/IEC 14598, as a guide aiming the quality. However at the encountered situations, the presence of the Educational software's final user (in our case, the teacher and the student), were ruled. Besides all, the quality aspects related with the use of those software in the teaching-learning processes were omitted. So, at this work we present a process model for quality evaluation of Educational Software products, considering educational aspects on its conception, giving more involvement of the students and teachers on this software evaluation.

KEYWORDS: Educational Software, Software Quality, Software Quality Models.

LISTA DE TABELAS

Tabela 1 - Descrição dos tópicos para Qualidade de Software (SWEBOK, 2004).....	19
Tabela 2 - Comparação entre os modelos de avaliação e seus respectivos fatores (FANG, 2008).....	25
Tabela 3 - Módulos observados no LAPS (PERELLI, 2004).	28
Tabela 4 - Documentos que compõem norma ISO/IEC 9126 e suas respectivas abordagens (ISO/IEC 9126, 2002).	32
Tabela 5 - Descrição da Característica Funcionalidade ISO/IEC 9126-1 e suas respectivas sub-características (ISO/IEC 9126, 2002).....	32
Tabela 6 - Descrição da Característica Confiabilidade ISO/IEC 9126-1 e suas respectivas sub-características (ISO/IEC 9126, 2002).....	33
Tabela 7 - Descrição da Característica Usabilidade ISO/IEC 9126-1 e suas respectivas sub-características (ISO/IEC 9126, 2002).....	33
Tabela 8 - Descrição da Característica Eficiência ISO/IEC 9126-1 e suas respectivas sub-características (ISO/IEC 9126, 2002).....	34
Tabela 9 - Descrição da Característica Manutenibilidade ISO/IEC 9126-1 e suas respectivas sub-características (ISO/IEC 9126, 2002).....	34
Tabela 10 - Descrição da Característica Portabilidade ISO/IEC 9126-1 e suas respectivas sub-características (ISO/IEC 9126, 2002).....	35
Tabela 11 - Descrição das sub-características contempladas na Qualidade em Uso na norma ISO/IEC 9126-1(ISO/IEC 9126, 2002).	35
Tabela 12 - Definição do processo de avaliação segundo a ISO/IEC 14598, Fonte: (ISO/IEC 14598. 1998).....	36
Tabela 13 - Relação de Trabalhos Científicos analisados e suas respectivas abordagens adaptada a partir de (LIMA, 2013).....	45
Tabela 14 - Verificação da existência (ou não) de aferição de qualidade nos produtos gerados na relação de trabalhos da Tabela 13.....	46
Tabela 15 - Pesos e seus respectivos graus de importância (BRATEFIXE, 2013).....	52
Tabela 16 - Notas e seus respectivos significados (BRATEFIXE, 2013).....	52
Tabela 17 - Julgamentos e classificações de acordo com valores de notas finais obtidas nos questionários (MARÇAL, 2009).....	53
Tabela 18 - Atribuição de pesos às questões sob a responsabilidade da E.U.....	69
Tabela 19 - Notas inseridas nos artefatos por cada um dos professores da E.U.	70
Tabela 20 - Resultados Finais da Avaliação de acordo com (MARÇAL, 2009).	70

LISTA DE FIGURAS

Figura 1 - Características McCall (1977) - (SOFTWARE QUALITY ATTRIBUTES, acessado em 20/01/2014).....	23
Figura 2 - Estrutura do Modelo de Reeves de acordo com (LITCHFIELD, 1994).....	29
Figura 3 - Processo de avaliação segundo a ISO/IEC 14598-1, sendo subsidiada pela ISO/IEC 9126 (ISO/IEC 14598, 1999).....	38
Figura 4 - Relação entre a ISO/IEC 9126 e ISO/IEC 14598 (NETO, 2005).....	40
Figura 5 - Ciclo de vida da norma ISO/IEC 9126 – 1.	47
Figura 6 - Fases da avaliação do Software Educacional descrito no modelo QUALI-EDU....	60
Figura 7 - Momentos da 3ª Fase (Fase de Execução da Avaliação) do QUALI-EDU.....	63
Figura 8 - Menu Principal do Jogo	67
Figura 9 - Tela Inicial do Jogo	68
Figura 10 - Jogo em Execução	68

SUMÁRIO

1.	INTRODUÇÃO.....	13
1.1	CONTEXTO E MOTIVAÇÃO.....	13
1.2	OBJETIVOS.....	15
1.2.1	OBJETIVO GERAL.....	15
1.2.2	OBJETIVOS ESPECÍFICOS.....	15
1.3	METODOLOGIA DE PEQUISA E DESENHO DE PESQUISA.....	16
1.4	A ESTRUTURA DO TRABALHO.....	17
2.	FUNDAMENTAÇÃO TEÓRICA.....	18
2.1	QUALIDADE DE SOFTWARE.....	18
2.2	MODELOS DE AVALIAÇÃO DE QUALIDADE DE SOFTWARE.....	20
2.2.1	HEURÍSTICAS DE USABILIDADE.....	20
2.2.2	AXIOMA ANTI-DECOMPOSIÇÃO DE WEYUKER.....	22
2.2.3	MODELO MCCALL (1977).....	22
2.2.4	MODELO BOEHM (1978).....	23
2.2.5	ESHTRI MODEL.....	24
2.2.6	MEDE-PROS.....	26
2.2.7	METODOLOGIA DO LAPS (LABORATÓRIO DE AVALIAÇÃO DE PRODUTOS DE SOFTWARE).....	27
2.2.8	REEVES (1994).....	28
2.3	FAMÍLIA DE NORMAS ISO/IEC 9126.....	31
2.4	FAMÍLIA DE NORMAS ISO/IEC 14598.....	36
2.5	RELAÇÃO ENTRE AS NORMAS ISO/IEC 9126 (9126-1, 9126-2, 9126-3 E 9126- 4) E ISO/IEC 14598 (14598-1, 14598-2, 14598-3, 14598-4 E 14598-5).....	39
2.6	SOFTWARE EDUCACIONAL.....	40
2.6.1	TIPOS DE SOFTWARE EDUCACIONAL DE ACORDO COM A FUNÇÃO EDUCATIVA.....	42
2.6.2	QUALIDADE DE SOFTWARE EDUCACIONAL.....	43
3.	QUALI-EDU: UM MODELO DE QUALIDADE DE SOFTWARE EDUCACIONAL.....	48
3.1	ESTRUTURA GERAL DO PROCESSO.....	50
3.2	DESCRIÇÃO DAS CARACTERÍSTICAS A SEREM ANALISADAS.....	54
3.3	FASES DA AVALIAÇÃO.....	60
3.3.1	FASE 1 – MOTIVAÇÃO DA AVALIAÇÃO.....	60
3.3.2	FASE 2 – ESPECIFICAÇÕES DA AVALIAÇÃO.....	61
3.3.3	FASE 3 - EXECUÇÃO DA AVALIAÇÃO.....	62
4.	AVALIAÇÃO DO QUALI-EDU.....	65
4.1	CONFIGURAÇÃO DO CENÁRIO DE AVALIAÇÃO.....	65
4.1.1	INSTRUMENTOS.....	65
4.1.2	EQUIPAMENTOS.....	65
4.1.3	PROFESSORES.....	65
4.1.4	SOFTWARE A SER AVALIADO.....	66
4.2	RELATO.....	68
5.	CONSIDERAÇÕES FINAIS.....	71
5.1	TRABALHOS FUTUROS.....	72
	REFERÊNCIAS.....	74
	ANEXOS.....	78

1. INTRODUÇÃO

1.1 CONTEXTO E MOTIVAÇÃO

É mais que notório o constante avanço, quase que diariamente das tecnologias disponíveis no mercado. Serviços e tarefas que demandavam muito esforço e tempo, atualmente conseguem ser executadas em tempo mais hábil, com o mínimo de esforço possível, graças às tecnologias vigentes. Nos mais diversos campos de atuação e nas mais diversas aplicações, a tecnologia está muito presente, de sistemas computacionais, até dispositivos que controlam nossos veículos e eletrodomésticos. Os chamados nativos digitais são os indivíduos que nasceram diante essa revolução tecnológica, onde, praticamente não conseguem viver sem tais tecnologias à disposição deles (BENNET ET AL., 2008).

Com a educação, não poderia ser diferente. As tecnologias computacionais e o constante desenvolvimento das tecnologias da informação desde os últimos anos tem transformado a forma com que a educação é conduzida atualmente (LOU, 2001). Por mais que, quando comparado ao avanço nas demais áreas, a educação continua o seu em passos lentos, apesar de existirem diversos sistemas que podem auxiliar no processo de ensino-aprendizagem.

A exemplo da realidade brasileira, foram adquiridos pelo governo federal no período de 1997 à 2011, por meio do Programa PROINFO, 100.900 laboratórios, totalizando o investimento de R\$1.045.813.917,22, onde destes apenas 85.073 laboratórios pleiteados foram entregues às escolas em todo território, ocasionando um investimento de R\$ 899.626.115,36 no período compreendido entre Janeiro de 1999 e Dezembro de 2012 (PAINEL MEC, acessado em 03/02/2014).

Deve-se considerar que, todos os computadores entregues, possuem como Sistema Operacional, o Linux Educacional, variando das versões 3.0 até a 5.0 (a mais recente). Tal

sistema operacional se diferencia dos demais por possuir diversos objetos de aprendizagem e softwares educacionais que buscam incentivar e assessorar a aprendizagem do alunado.

O presente trabalho tratará mais especificadamente do Linux Educacional 3.0. O mesmo é baseado no Kubuntu 8.04, com ambiente gráfico KDE 3.5.10. O núcleo do sistema é o Linux Ubuntu 2.6.24-22-generic. Os programas instalados nos computadores dos laboratórios podem ser subdivididos de acordo com a relação encontrada no anexo 1 deste trabalho.

Conforme é possível observar, o Linux Educacional 3.0 é um sistema relativamente completo, abordando além da parte educacional, também temas mais específicos, como edição de mídia (áudio, imagem, vídeo).

Para que os objetivos esperados com a implantação de tais laboratórios sejam alcançados é esperado dos “cidadãos do século 21”, que os mesmos sejam tecnologicamente preparados para serem: a) usuários capacitados ante as tecnologias da informação, b) Buscadores, analisadores e avaliadores de Informação, c) Solucionadores de problemas e tomadores de decisões, d) Usuários criativos e eficientes de ferramentas de produtividade, e) Comunicadores, colaboradores, publicadores e produtores (COUTINHO, 2011).

Porém, é confirmado que, embora as tecnologias computacionais possuam o potencial de serem uma ferramenta poderosa e flexível (SCARDAMALIA & BEREITER, 1996), experiências anteriores onde, tecnologias recentes foram utilizadas nas escolas (rádio, televisão, entre outros), acentuam o fato de que a simples instalação de hardware não irá levar-nos aos resultados esperados.

Considerando as afirmações acima, o presente trabalho propõe o QUALI-EDU, um Processo de Avaliação da Qualidade do Produto de Software Educacional, baseado nas Normas da ISO (International Organization for Standardization) 9126 e 14558, as quais

tratam da definição de atributos necessários para avaliação e a avaliação do produto de software baseados nestes, respectivamente.

Como contribuições desta pesquisa pode-se elencar o próprio QUALI-EDU, como processo de avaliação de software educacional, carência observada em vários dos trabalhos pesquisados. Além disto, o QUALI-EDU, além de um modelo de avaliação de qualidade de software educacional, pode ser utilizado também como um modelo de seleção de Software Educacional, considerando os aspectos educacionais presentes no mesmo e os clientes finais como avaliadores, permitindo que o software seja selecionado ou não, para determinado contexto.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

Definir um modelo de qualidade de avaliação de qualidade de software educacional, baseado nas famílias de normas ISO/IEC 9126 e ISO/IEC 14598, e nas metodologias ágeis, que consiga aproximar-se ao máximo dos clientes finais (professores e alunos), considerando que nas normas supracitadas, não existem módulos destinados à aferição das características inerentes aos aspectos educacionais.

1.2.2 OBJETIVOS ESPECÍFICOS

- Especificar regras e artefatos para que o julgamento do software envolva as características existentes na família de normas supracitadas, com o adendo dos aspectos educacionais;
- Descrever etapas que envolvem o processo de avaliação de qualidade de software educacional;
- Aplicar tal modelo a um grupo de professores da área selecionada.

1.3 METODOLOGIA DE PEQUISA E DESENHO DE PESQUISA

Foi realizada uma pesquisa de caráter bibliográfico e estudo de caso. Quanto aos meios, foi utilizada pesquisa qualitativa. O instrumento de avaliação da pesquisa será a pesquisa de campo, com o uso de questionário e de observação pessoal. Os sujeitos da pesquisa foram escolhidos por acessibilidade.

Conforme (CLARK, 1983) a pesquisa bibliográfica é o estudo desenvolvido com base em material publicado em livros, revistas, jornais, redes eletrônicas, ou seja, material acessível ao público em geral. Fornece instrumento analítico para qualquer outro tipo de pesquisa, mas também pode esgotar-se em si mesma.

Para (VERGARA, 2005) e (BOGDAN ET. AL., 1994): “Conceituam estudo de caso como sendo uma observação detalhada de um contexto ou indivíduo, de uma única fonte de documentos ou de um acontecimento específico.”.

De acordo com (TRIVIÑOS, 1987) estudo de caso é caracterizado pelo estudo aprofundado de um ou de poucos objetos, de forma que permita o seu detalhamento amplo de conhecimento, tarefa praticamente impossível mediante os outros delineamentos considerados.

Quanto ao tipo de pesquisa, (GIL, 1995) apresenta um esboço acerca destas duas estratégias:

A pesquisa qualitativa, por sua vez, descreve a complexidade de determinado problema, sendo necessário compreender e classificar os processos dinâmicos vividos nos grupos, contribuir no processo de mudança, possibilitando o entendimento das mais variadas particularidades dos indivíduos.

1.4 A ESTRUTURA DO TRABALHO

Esta seção apresenta a estrutura utilizada na organização deste trabalho. Neste primeiro capítulo foi apresentada a introdução, a motivação para a pesquisa realizada, objetivos, metodologia e desenho de pesquisa.

No segundo capítulo, será tratada a fundamentação teórica, onde será explicada a qualidade de software onde serão apresentados os principais modelos de avaliação de software encontrados nos trabalhos científicos pesquisados, lembrando que todos os artigos científicos abordaram a produção/aplicação de softwares educacionais.

Neste mesmo capítulo, serão apresentadas mais profundamente as famílias de normas ISO/IEC 9126 e ISO/IEC 14598, além da relação existente entre estas. Foram escolhidas essas normas principalmente pelo fato dessas serem as normas que mais foram trabalhadas e pesquisadas nos trabalhos científicos os quais foram encontrados na pesquisa feita.

Considerando que o trabalho envolve diretamente a avaliação de software educacional, ainda na fundamentação teórica (capítulo 2), será explicado esse tipo de software e seus tipos e também o que existe no que se diz respeito à qualidade desta taxonomia de software.

Já no terceiro capítulo, é apresentado o QUALI-EDU: Um modelo de qualidade de software educacional, sua estrutura geral, características à serem analisadas e as fases que compreendem a avaliação do software educacional por meio desse modelo.

Após a aplicação deste modelo em situação real, no quarto capítulo, o QUALI-EDU é avaliado, em condições reais de sala de aula, por um grupo de professores de matemática do Ensino Fundamental I, da rede municipal de ensino de uma cidade no interior da Paraíba.

No quinto capítulo serão descritas as considerações finais sobre o trabalho e a abertura existente para trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 QUALIDADE DE SOFTWARE

A ISO-8402 define a qualidade como sendo a totalidade das características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas.

A qualidade de software, quando descrita nos guias do SWEBOK Guide, demonstra que autores e entidades da área, tratam do termo “qualidade” de forma diferente. A exemplo de (ISO 8402, 1994), que trata do termo como sendo a “conformidade dos requisitos do usuário”, já (HUMPHREY, 1989) trata como sendo o momento em que “os níveis de adequação de uso são atingidos em sua excelência”. Mais recentemente, a (ISO 9000, 2005) citou como sendo “o grau em que um conjunto de características inerentes ao produto, satisfaz os requisitos levantados”.

A avaliação de qualidade depende de características qualitativas necessárias que descrevam o objetivo do produto do software (requisitos x usabilidade) (FANG, 2008).

Foi observado conforme a seguir , na tabela 1 , retirada do (SWEBOK, 2004) , que trata sobre a qualidade de software, os tópicos relacionados à qualidade de software.

Qualidade de Software	Fundamentos da Qualidade de Software	Cultura e ética da Engenharia de Software
		Valores e Custos da Qualidade
		Modelos e Características da Qualidade
	Processos de Gerenciamento da Qualidade de Software	Garantia da Qualidade do Software
		Verificação de Validação
		Revisões e Auditorias
	Considerações Práticas	Requisitos da Qualidade do Software
		Caracterização de Defeitos
		Técnicas de Gerenciamento da Qualidade do Software
	Ferramentas de Qualidade de Software	

Tabela 1 - Descrição dos tópicos para Qualidade de Software (SWEBOK, 2004)

É interessante ser destacado que o engenheiro de software deve ter o conhecimento sobre os conceitos e características da qualidade de software e o seu valor, seja na fase de desenvolvimento ou nas manutenções corriqueiras (BOTT ET. AL, 2000). É fato que uma engenharia de software saudável, deve possuir (além de outras características), procedimentos que garantam a qualidade como uma das prioridades (KAN, 2002).

A análise da qualidade de software deve ser abrangente, inovadora e flexível, ante seu contexto (KELLY, 2008). Segundo a norma ISO/IEC 9126-1 (ISO/IEC 9126, 2002), deve ser considerado um modelo de qualidade interno e externo, para que a qualidade seja assegurada no período de desenvolvimento (*design*, implementação, execução e validação), e também um modelo de qualidade “em uso”, abordando diretamente usuários específicos de determinado contexto.

Tratando do padrão de qualidade interno, devem ser observadas diretamente seis características (funcionalidade, confiabilidade, usabilidade, eficiência, manutenção e

portabilidade), já no padrão de qualidade “em uso”, devem ser observadas quatro características (efetividade, produtividade, segurança e satisfação).

É considerado que as opiniões mais recentes estão mais focadas na satisfação do usuário, de acordo com os dados levantados, a presença do usuário final está mais presente no que se trata na confirmação da qualidade do software.

2.2 MODELOS DE AVALIAÇÃO DE QUALIDADE DE SOFTWARE

Será descrito abaixo os modelos que foram encontrados no levantamento de literatura realizado. Dos encontrados, a grande maioria teve como base as normas ISO/IEC 9126, 14598 e 12119. Nos trabalhos averiguados também foram encontradas metodologias de avaliação produzidas no Brasil (a exemplo do MEDE-PROS (MEDE-PROS, 2004) e do LAPS (PERELLI, 2004)).

2.2.1 HEURÍSTICAS DE USABILIDADE

Utilizada principalmente para avaliação de usabilidade de sistemas computacionais. Desenvolvido por Nielsen e Molich, na década de 90, aborda 10 (dez) características as quais, caso o sistema as possua, o mesmo possui qualidade suficiente para ser utilizado diante o contexto pré-determinado. As regras seguem abaixo (Nielsen & Jakob, 1994).

1) Feedback

O sistema deve informar continuamente ao usuário sobre o que ele está fazendo. 10 segundos é o limite para manter a atenção do usuário focalizada no diálogo.

2) Falar a linguagem do usuário

A terminologia deve ser baseada na linguagem do usuário e não orientada ao sistema. As informações devem ser organizadas conforme o modelo mental do usuário.

3) Saídas claramente demarcadas

O usuário controla o sistema, ele pode, a qualquer momento, abortar uma tarefa, ou desfazer uma operação e retornar ao estado anterior.

4) Consistência

Um mesmo comando ou ação deve ter sempre o mesmo efeito. A mesma operação deve ser apresentada na mesma localização e deve ser formatada/apresentada da mesma maneira para facilitar o reconhecimento.

5) Prevenir erros

Evitar situações de erro. Conhecer as situações que mais provocam erros e modificar a interface para que estes erros não ocorram.

6) Minimizar a sobrecarga de memória do usuário

O sistema deve mostrar os elementos de diálogo e permitir que o usuário faça suas escolhas, sem a necessidade de lembrar um comando específico.

7) Atalhos

Para usuários experientes executarem as operações mais rapidamente. Abreviações, teclas de função, duplo clique no mouse, função de volta em sistemas hipertexto. Atalhos também servem para recuperar informações que estão numa profundidade na árvore navegacional a partir da interface principal.

8) Diálogos simples e naturais

Deve-se apresentar exatamente a informação que o usuário precisa no momento, nem mais nem menos. A sequência da interação e o acesso aos objetos e operações devem ser compatíveis com o modo pelo qual o usuário realiza suas tarefas.

9) Boas mensagens de erro

Linguagem clara e sem códigos. Devem ajudar o usuário a entender e resolver o problema. Não devem culpar ou intimidar o usuário.

10) Ajuda e documentação

O ideal é que um software seja tão fácil de usar (intuitivo) que não necessite de ajuda ou documentação. Se for necessária a ajuda deve estar facilmente acessível on-line (Nielsen & Jakob, 1994).

2.2.2 AXIOMA ANTI-DECOMPOSIÇÃO DE WEYUKER

O referido modelo retrata que testar todos os componentes do software pode ser adequado, mas não necessariamente, adequado para o software. Isso implica dizer que testar um nível de qualidade de baixa granularidade, pode ser adequado, porém, não adequado o suficiente para níveis mais altos de granularidade.

Ou seja, um programa adequadamente testado, não quer dizer que seus componentes em sua totalidade estão testados, pois não é possível garantir que os procedimentos internos funcionam corretamente, não é possível garantir que o programa produza o que se espera (WEYUKER, 1986).

Diante tal modelo, é de suma importância testar cada procedimento separadamente, com testes de unidade, de programa e de integração.

2.2.3 MODELO MCCALL (1977)

O modelo McCall foi produzido por Jim McCall para as Forças Aéreas americanas, com o intuito de criar uma ponte entre os usuários e os desenvolvedores, com o intuito principal de deixar mais clara para a equipe de desenvolvimento a visão do usuário, visando a maior coerência entre o solicitado pelo cliente com o que foi entregue (SOFTWARE QUALITY ATTRIBUTES, acessado em 20/01/2014).

As três principais perspectivas descritas por McCall, foram a da Revisão do Produto (possibilidade à alterações), Transição do Produto (adaptabilidade a novos ambientes), e Operações do Produto (Características Operacionais Básicas) (SOFTWARE QUALITY ATTRIBUTES, acessado em 20/01/2014).

Neste modelo, é possível identificar grande similaridade com os que são baseados nas normas ISO anteriormente descritas. O quadro abaixo descreve quais fatores são avaliados em cada perspectiva (SOFTWARE QUALITY ATTRIBUTES, acessado em 20/01/2014).

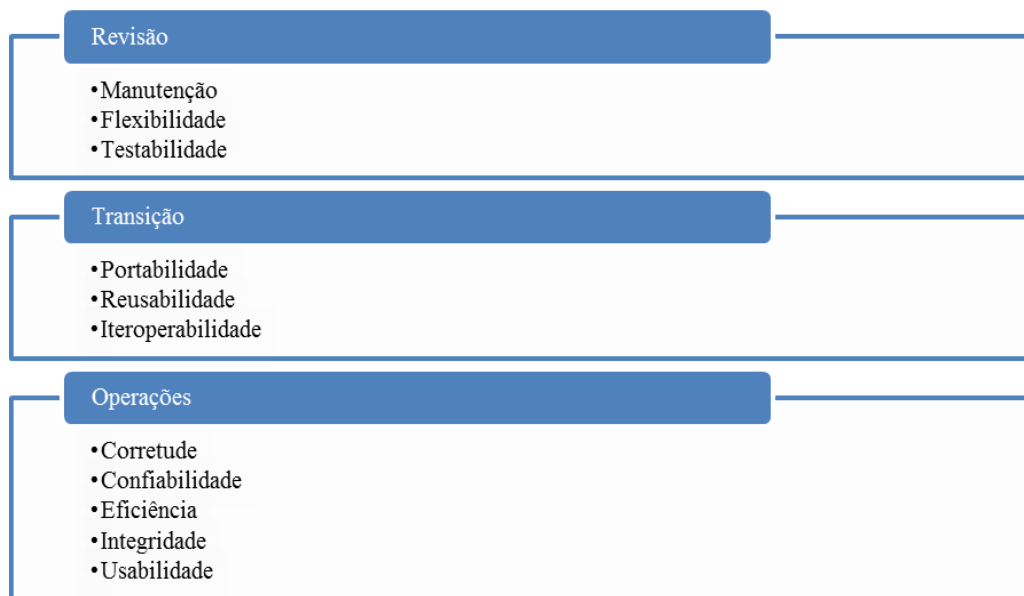


Figura 1 - Características McCall (1977) - (SOFTWARE QUALITY ATTRIBUTES, acessado em 20/01/2014).

2.2.4 MODELO BOEHM (1978)

Barry W. Boehm, na tentativa de definir a qualidade de software qualitativamente, definiu um modelo hierárquico de características de qualidade de software, de acordo com um conjunto de atributos e métricas. O mesmo definiu três princípios de qualidade (requisitos básicos de software), sendo eles Portabilidade, Manutenção e a Facilidade de uso.

Cada um desses princípios de qualidade possuíam fatores qualitativos agregados, formando assim o segundo nível hierárquico do modelo (SOFTWARE QUALITY ATTRIBUTES, acessado em 20/01/2014).

Muito similar ao modelo anterior, os fatores eram: Portabilidade, Confiança, Eficiência, Usabilidade, Testabilidade, Compreensibilidade e Flexibilidade (SOFTWARE QUALITY ATTRIBUTES, acessado em 20/01/2014).

Vale ressaltar que cada um desses fatores, são correlacionados com outros sub-fatores, a exemplo de Testabilidade, pode se subdividir em acessibilidade, comunicabilidade, entre outros.

2.2.5 ESHTRI MODEL

É a sigla para “Educational Software Quality Hierarchy Triangle”, em tradução livre Triângulo Hierárquico de Qualidade de Software Educacional. O mesmo é baseado nas prerrogativas cabíveis para software educacional, descritas nos modelos de McCall, Boehm e também na ISO 9126 (FANG, 2008).

Tal modelo possui três papéis principais: o de produtor de software (empresas de desenvolvimento), de receptor (escolas e instituições que venham a utilizar o software educacional), e produto de software (que prescreve os fatores de qualidade de software).

O modelo considera 13 (treze) fatores como significantes e diretamente influenciáveis para a qualidade final do software educacional. Para cada fator, métricas diretas (pois facilitam no entendimento, comparação) são associadas, para servirem como base para representações quantitativas, de acordo com a tabela a seguir, que compara o modelo ESHtri, com os demais modelos.

Fatores de Qualidade	McCall 1977	Boehm 1978	ISO9126 1993	ESHTri Model
Corretude	X	X	X	X
Confiança	X	X	X	X
Integridade	X	X		X

Usabilidade	X	X	X	X
Eficiência na Execução	X	X	X	X
Manutenção	X	X	X	X
Testabilidade	X			X
Interoperabilidade	X			X
Flexibilidade	X	X		X
Reusabilidade	X	X		X
Portabilidade	X	X	X	X
Clareza		X		
Instabilidade		X	X	X
Autodescrição		X		X
Facilidade modificação		X		
Compreensibilidade		X		
Eficiência		X	X	X
Funcionalidade			X	
Generalidade		X		
Economia		X		X

Tabela 2 - Comparação entre os modelos de avaliação e seus respectivos fatores (FANG, 2008).

2.2.6 MEDE-PROS

Criado pelo Centro de Tecnologia da Informação Renato Archer (CenPRA), tem como principal objetivo avaliar a qualidade do produto de software, de forma genérica, segundo a visão do usuário, verificando o quanto este está em conformidade com as Normas Internacionais de qualidade. Como resultado, o solicitante recebe um relatório de avaliação, com pontos positivos e pontos à serem revisados (CENPRA, 2004).

O referido modelo é baseado nas normas ISO/IEC 9126, ISO/IEC 14598, ISO/IEC 12119 e NBR 13596. O MEDE-PROS, possui uma abrangência maior em relação ao produto de software em geral, os módulos que ele possui são os seguintes:

- Interface (baseado nas normas: NBR 13596/ISO 9126, ISO/DIS 9241-11,12,14,16 ERGOLIST);
- Software (baseado na norma NBR 13596 – ISO 9126);
- Documentação (baseado na norma NBR ISO/IEC 12119, ANSI/IEEE 1063, NBR 13596/ISO 9126);
- Descrição do Produto (baseado na norma NBR ISO/IEC 12119);
- Embalagem (baseado na norma NBR ISO/IEC 12119);

Os instrumentos utilizados para avaliação da qualidade são feitos com relatórios pré-modelados, manuais de avaliação e listas de verificação. Os relatórios de avaliação (que abordam aspectos qualitativos e quantitativos) são todos documentados e analisados pela equipe de desenvolvimento, onde, buscam expor/discutir os possíveis e recorrentes problemas que possam ser observados quando a avaliação venha a ser executada.

2.2.7 METODOLOGIA DO LAPS (LABORATÓRIO DE AVALIAÇÃO DE PRODUTOS DE SOFTWARE)

É uma proposta de avaliação de qualidade de software, embasada nas normas da ISO/IEC 14598, e seguindo o modelo de qualidade da ISO/IEC 9126, é toda estruturada em módulos avaliativos.

A análise por ser baseada em módulos permite ao usuário, selecionar quais os módulos (ou conjuntos de módulos) que considera crítico ou específico o suficiente, para que o feedback dado, seja de acordo com o que o mesmo necessita averiguar. Também de forma modularizada, é aumentada a percepção global da avaliação (NETO, 2005).

A principal característica do LAPS, quando comparado aos demais, é o fato deste ter o envolvimento de um especialista no domínio do produto, auxiliando na precisão da avaliação final.

Os resultados de cada módulo são dados por meio de uma pontuação quantitativa de 0 à 100 pontos, além de uma análise qualitativa, descrevendo textualmente as opiniões do avaliador.

O custo de uma avaliação de qualidade de software de acordo com o LAPS é diretamente relacionado ao porte do produto, módulos utilizados na avaliação (pois nem sempre são todos, dependendo do domínio/contexto) e as técnicas utilizadas (NETO, 2005).

Abaixo, a tabela que representa claramente os módulos do LAPS.

MÓDULO	PROPÓSITO
Código-fonte	Verificar o grau de qualidade do código-fonte, de acordo com os padrões.
Performance	Verificar se o tempo e recurso gasto por operação estão de acordo com as restrições estabelecidas.
Documentação do usuário	Verificar se a documentação do usuário está de fácil entendimento.

Usabilidade	Verificar o grau de compreensão do sistema quando utilizado pelo usuário, assim como a facilidade de aprendizado e uso.
Funcionalidade	Verificar se o produto realiza as funções para o qual foi desenvolvido.
Falhas e Recuperação	Verificar o desempenho do produto na ocorrência de falhas e sua recuperação em tempo hábil.
Arquitetura	Verificar se a arquitetura utilizada promove a modularidade e reusabilidade das aplicações.
Portabilidade	Verificar o grau de adequação do sistema a diferentes ambientes e plataformas.
Segurança e Controle de Acesso	Verificar que o sistema permite acesso as funcionalidades e informações apenas para usuários autorizados.
Documentação do Sistema	Verificação se a documentação do sistema foi dada conforme os padrões da empresa.
Competidores	Verificar a qualidade do produto em consideração a relação custo-benefício dados outros softwares do segmento.
Especialista	Verificar a existência de possíveis problemas referentes à ausência de regras de negócio importantes no contexto do sistema e funcionalidades que fogem do escopo do software; como também sugerir novas funcionalidades que agreguem valor ao produto.

Tabela 3 - Módulos observados no LAPS (PERELLI, 2004).

2.2.8 REEVES (1994)

O modelo de design multimídia de Reeves, é um típico modelo de design instrucional de sistemas, destinado à educação e treinamento. O mesmo inclui basicamente 4 (quatro) funções prioritárias, conforme a ordem que segue: Análise, *Design*, Produção e Avaliação. Cada fase destas incita a um conjunto de atividades específicas desenvolvidas por um grupo de pessoas na equipe de desenvolvimento (LITCHFIELD, 1994).

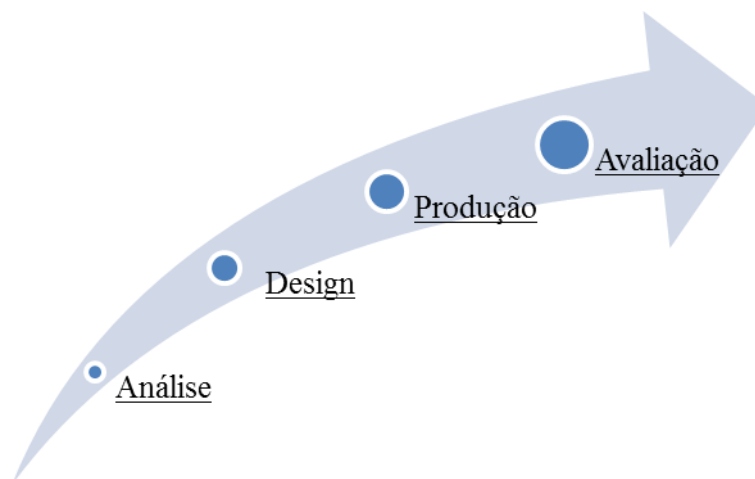


Figura 2 - Estrutura do Modelo de Reeves de acordo com (LITCHFIELD, 1994).

Os personagens da equipe de desenvolvimento no modelo de Reeves são: Gerente do Projeto, Especialista, Designer Instrucional, Avaliador do Projeto, Programador, Especialista em gráficos e Especialista em vídeo. Explicitaremos as atividades e os personagens de cada fase abaixo.

FASE 1 – ANÁLISE

- a) Personagens: Gerente do Projeto, Especialista, Designer Instrucional, Avaliador do Projeto e Programadores.
- b) Processos: Especificar conteúdos e objetivos, preparar audiência de avaliação, realizar avaliação das necessidades, planejamento de projeto e definição de estratégias para avaliação de planejamento.
- c) Saídas: Relatório da avaliação de necessidades, perfil do aprendiz, esboço do conteúdo, hierarquia do sistema, objetivos educacionais, definição de diagramas PERT¹, cronograma do projeto e plano de avaliação.

¹ Program Evaluation Review Technique, Técnica de Revisão de Avaliação de Programas, desenvolvido pela marinha americana na década de 50, com o intuito de coordenar as tarefas dentro de um projeto. (Definição do Diagrama Pert, acessado em 02/02/2014)

FASE 2 – DESIGN

- a) Personagens: Gerente do Projeto, Especialista, Designer Instrucional, Avaliador do Projeto, Programadores, Especialista em Gráficos e Especialista em Vídeo.
- b) Processos: Especificar interações instrucionais de acordo com os objetivos educacionais do sistema, projeto de interface, fluxo de gráficos, prototipagem, telas de formatação e realização de revisões constantes.
- c) Saídas: Arquétipos de instrução, fluxogramas, scripts, formatos padronizados, protótipos IMM, especificações de possíveis melhorias de acordo com as revisões.

FASE 3 – PRODUÇÃO

- a) Personagens: Gerente do Projeto, Especialista, Designer Instrucional, Avaliador do Projeto, Programadores, Especialista em Gráficos e Especialista em Vídeo.
- b) Processos: Especificar interações de sistema de autoria (SCORM² por exemplo), criação de gráficos e materiais de apoio, realizar a pré-produção, realizar a produção e a pós produção, preparar modo de difusão do sistema (CDs, *Web*, etc.).
- c) Saídas: Código interativo, gráficos, materiais de apoio, documentação do programa, componentes/animações do sistema em perfeito funcionamento, mídia ou endereço de disponibilização do Sistema.

FASE 4 – AVALIAÇÃO

- a) Personagens: Gerente do Projeto, Especialista, Designer Instrucional, Avaliador do Projeto, Programadores.
- b) Processos: Documentação do processo, teste e validação do IMM, condução a avaliação de impacto.

² Sharable Content Object Reference Model (SCORM) – coleção de padrões e especificações destinados à EAD baseados na web. Tem como objetivo padronizar conteúdos, reutilizar OA's, flexibilizar a aprendizagem e permitir migração para outras plataformas. (Definição do SCORM, acessado em 20/02/2014).

- c) Saídas: Documentação do projeto, validação funcional do IMM, validação instrucional do IMM, relatório de avaliação formativa, relatório de avaliação de eficácia e relatório de avaliação de impacto.

2.3 FAMÍLIA DE NORMAS ISO/IEC 9126

Com o avanço da computação, os sistemas computacionais necessitaram mudar a ótica existente no que diz respeito à qualidade de software. Pelo fato dos softwares, anteriormente, atentarem diretamente para a substituição de hardware, as avaliações eram baseadas apenas na funcionalidade.

Diante disso, e pensando no quesito da confiabilidade e visando a maior satisfação do cliente (visto que anteriormente o critério de qualidade era estabelecido pelo fornecedor (PUNTER ET. AL., 1997), surgiu a família de normas ISO/IEC 9126, propondo um modelo de qualidade de produto de software.

Divida em quatro documentos, essa família de normas aborda as seguintes situações:

Norma	Abordagem
ISO/IEC 9126-1	Modelo de qualidade: definindo um modelo de qualidade para o produto de software e descrevendo um conjunto de características e sub-características de qualidade.
ISO/IEC 9126-2	Métricas externas: apresentar formas de medir atributos das características descritas na ISO/IEC 9126-1. Utilizada quando o mesmo já está pronto para execução
ISO/IEC 9126-3	Métricas internas: apresentar formas de medir atributos das características descritas na ISO/IEC 9126-1, porém diferenciando da ISO/IEC 9126-2, esta está diretamente associada aos produtos intermediários (projeto e código).
ISO/IEC 9126-4	Métricas de qualidade em uso: apresentar formas de medir atributos das características descritas na ISO/IEC 9126-1, porém diferenciando da ISO/IEC 9126-2 e

	da ISO/IEC 9126-3, esta representa a perspectiva do usuário.
--	--

Tabela 4 - Documentos que compõem norma ISO/IEC 9126 e suas respectivas abordagens (ISO/IEC 9126, 2002).

A primeira norma, a ISO/IEC 9126-1, aborda 6 (seis) categorias principais, onde descreveremos cada uma destas.

A primeira característica à ser averiguada, a funcionalidade, nessa norma, corresponde à capacidade que o software tem de prover funções que atendam as necessidades explícitas e implícitas quando o software estiver sendo utilizado nas condições especificadas. As sub-características apresentadas abaixo buscam averiguar a existência dessa capacidade.

FUNCIONALIDADE	Adequação	Capacidade de prover funções para tarefas e objetivos descritos pelo usuário.
	Acurácia	Capacidade de prover de forma precisa resultados e efeitos corretos de acordo com descrito pelo usuário.
	Interoperabilidade	Capacidade de interação com um ou mais sistemas especificados.
	Segurança de Acesso	Capacidade de proteger as informações e dados, para que pessoas ou sistemas não autorizados, utilizem/modifiquem dados, e que não seja negado às pessoas ou sistemas autorizados.
	Conformidade relacionada à funcionalidade	Capacidade do software estar de acordo com normas, convenções ou regulamentações previstas em leis ou prescrições similares relacionadas à funcionalidade.

Tabela 5 - Descrição da Característica Funcionalidade ISO/IEC 9126-1 e suas respectivas sub-características (ISO/IEC 9126, 2002).

A segunda característica à ser mesurada, a confiabilidade, busca averiguar a capacidade do software de manter um nível de desempenho correto (diante do especificado nos requisitos) quando usado nas condições especificadas. As sub-características da confiabilidade são descritas na tabela abaixo.

CONFIABILIDADE	Maturidade	Capacidade de evitar falhas decorrentes de defeitos no software.
	Tolerância à falhas	Capacidade de manter um nível de desempenho especificado em casos de defeitos no software ou violação de sua interface especificada.
	Recuperabilidade	Capacidade de restabelecer nível de desempenho especificado e recuperar os dados afetados no caso de falhas.
	Conformidade relacionada à confiabilidade	Capacidade do software estar de acordo com normas, convenções ou regulamentações previstas em leis ou prescrições similares relacionadas à confiabilidade.

Tabela 6 - Descrição da Característica Confiabilidade ISO/IEC 9126-1 e suas respectivas sub-características (ISO/IEC 9126, 2002).

A terceira característica a ser verificada, a usabilidade, busca averiguar o quão compreensível, aprendível e operável e atraente o software é para o usuário, de acordo com o que foi especificado nos requisitos do software. As sub-características estão descritas na tabela abaixo.

USABILIDADE	Inteligibilidade	Capacidade do software possibilitar ao usuário compreender se este é apropriado para as tarefas e condições de uso específicas.
	Apreensibilidade	Capacidade de permitir que o usuário aprenda sua aplicação.
	Operacionalidade	Capacidade de possibilitar controle e operacionalidade por parte do usuário.
	Atratividade	Capacidade de ser atraente para o usuário.
	Conformidade relacionada à usabilidade	Capacidade do software estar de acordo com normas, convenções ou regulamentações previstas em leis ou prescrições similares relacionadas à usabilidade.

Tabela 7 - Descrição da Característica Usabilidade ISO/IEC 9126-1 e suas respectivas sub-características (ISO/IEC 9126, 2002).

A quarta característica a ser avaliada, a eficiência, busca analisar o desempenho do software, relacionando o desempenho com a utilização de recursos, e dessa forma, comparar com o que foi considerado apropriado nos requisitos do software. As sub-características estão descritas na tabela abaixo.

EFICIÊNCIA	Comportamento em relação ao tempo	Capacidade de fornecer tempos de resposta e de processamento, taxas de transferência, quando as executadas as funções nas condições estabelecidas pelo usuário.
	Utilização de recursos	Capacidade de usar tipos e quantidades apropriados de recursos
	Conformidade relacionada à eficiência	Capacidade do software estar de acordo com normas, convenções ou regulamentações previstas em leis ou prescrições similares relacionadas à eficiência.

Tabela 8 - Descrição da Característica Eficiência ISO/IEC 9126-1 e suas respectivas sub-características (ISO/IEC 9126, 2002).

A quinta característica avaliada a manutenibilidade, visa avaliar o quão modificável o software é, considerando principalmente a possibilidade de mudanças que o software possui, principalmente se for o caso do software ser modificado e ainda assim, ser validado. As sub-características estão descritas na tabela abaixo.

MANUTENIBILIDADE	Analisabilidade	Capacidade de permitir diagnóstico de causas de falhas no software, bem como adaptações devido a mudanças no ambiente, requisitos ou especificações funcionais.
	Modificabilidade	Capacidade de implementação de modificações.
	Estabilidade	Capacidade de evitar efeitos ante as modificações realizadas.
	Testabilidade	Capacidade de permitir a validação do software após as modificações realizadas.
	Conformidade relacionada à manutenibilidade	Capacidade do software estar de acordo com normas, convenções ou regulamentações previstas em leis ou prescrições similares relacionadas à manutenibilidade.

Tabela 9 - Descrição da Característica Manutenibilidade ISO/IEC 9126-1 e suas respectivas sub-características (ISO/IEC 9126, 2002).

A sexta e última característica a ser avaliada, a portabilidade, busca verificar principalmente a capacidade do software ser transferido de um ambiente para outro, ou seja, verificar o quão adaptável à diferentes plataformas/dispositivos o software é. As sub-características estão dispostas na tabela abaixo.

PORTABILIDADE	Adaptabilidade	Capacidade de adaptação a diferentes ambientes sem necessidade de aplicações de outros meios além dos existentes destinados à execução da tarefa esperada.
	Capacidade para ser instalado	Capacidade de ser instalado em ambiente especificado.
	Coexistência	Capacidade de coexistir com outros softwares em ambiente de recursos compartilhados.
	Capacidade para substituir	Capacidade de ser utilizado em substituição à outro software nas mesmas condições pré-estabelecidas.
	Conformidade relacionada à portabilidade	Capacidade de o software estar de acordo com normas, convenções ou regulamentações previstas em leis ou prescrições similares relacionadas à portabilidade.

Tabela 10 - Descrição da Característica Portabilidade ISO/IEC 9126-1 e suas respectivas sub-características (ISO/IEC 9126, 2002).

A norma ISO/IEC 9126-1, também trata diretamente um modelo de qualidade para qualidade em uso, que seria o que o usuário de fato poderia observar, conforme tabela abaixo.

QUALIDADE EM USO	Eficácia	Capacidade de permitir que as metas especificadas sejam atingidas pelo usuário com acurácia e completude.
	Produtividade	Capacidade de permitir ao usuário empregar a quantidade apropriada de recursos em relação a eficácia obtida.
	Segurança	Capacidade de apresentar níveis aceitáveis de riscos de dados a pessoas, negócios, softwares, propriedades e ambientes.
	Satisfação	Capacidade de satisfazer o usuário.

Tabela 11 - Descrição das sub-características contempladas na Qualidade em Uso na norma ISO/IEC 9126-1(ISO/IEC 9126, 2002).

Para o desenvolvimento desta pesquisa escolheu-se detalhar melhor a norma ISO/IEC 9126-1, pelo fato desta ser a que mais será utilizada no processo de avaliação de software educacional proposto neste trabalho. As demais normas (ISO/IEC 9126-2, ISO/IEC 9126-3 e ISO/IEC 9126-4), tratam do fornecimento e descrição de métricas externas e também da apresentação de métricas de qualidades de uso, respectivamente.

2.4 FAMÍLIA DE NORMAS ISO/IEC 14598

Segundo (PUNTER ET. AL., 1997) um dos pontos cruciais da avaliação do produto de software é a objetividade, ou seja, a mesma não pode ser baseada em opiniões e sim, em observações. Ao ser reproduzida por diferentes avaliadores, devem produzir resultados idênticos ou repetíveis.

Daí surge a importância da família de normas ISO/IEC 14598, a qual é dividida em 6 (seis) partes, que estabelecem de forma processual, a definição precisa das ações para: análise dos requisitos de avaliação, especificação da avaliação, projeto e planejamento da avaliação, execução da avaliação e por último, porém não menos importante, a documentação dos resultados (ISO/IEC 14598, 1998).

A referida família de normas segue basicamente 4 (passos) conforme descreveremos na tabela abaixo.

Passos Papéis	Análise	Especificação	Projeto	Execução
Desenvolvedor	Definir requisitos de qualidade e análise de exequibilidade	Quantificar os requisitos de qualidade	Planejar a avaliação durante o desenvolvimento.	Monitorar a qualidade e controlar o desenvolvimento
Adquirente	Estabelecer propósito e escopo da avaliação	Definir métricas externas e medições à serem realizadas	Planejar, programar e documentar toda a avaliação.	Realizar, documentar e analisar a avaliação
Avaliador	Descrever os objetivos da avaliação	Definir escopo da avaliação e das medições	Documentar todo processo a ser utilizado.	Obter resultados a partir da realização de ações de medição e verificação do produto

Tabela 12 - Definição do processo de avaliação segundo a ISO/IEC 14598, Fonte: (ISO/IEC 14598, 1998).

Em linhas gerais podemos apresentar cada parte da seguinte forma:

- ISO/IEC 14598-1 (ISO/IEC 14598-1, 1999) – Esta norma fornece uma visão geral do processo de avaliação, além de definir a estrutura de funcionamento das normas desta família;

- ISO/IEC 14598-2 (ISO/IEC 14598-2, 1999) – Esta norma refere-se ao planejamento e gestão do processo, nesta parte requisitos, recomendações e orientações são apresentados, auxiliando os avaliadores, caso seja necessário suporte ao processo;
- ISO/IEC 14598-3 (ISO/IEC 14598-3, 1999)– Define o processo de avaliação para os desenvolvedores, utilizando durante o processo de desenvolvimento e manutenção do produto;
- ISO/IEC 14598-4 (ISO/IEC 14598-4, 1999) – Define o processo de avaliação para os adquirentes, estabelecendo sistematicamente um processo para avaliação de softwares do tipo pacote (de acordo com (ISO/IEC 12119, 1998)), software sob encomenda ou modificações em softwares já existentes.
- ISO/IEC 14598-5 (ISO/IEC 14598-5, 1999) – Define o processo de avaliação para os avaliadores, fornecendo orientações para implementação prática da avaliação
- ISO/IEC 14598-6 (ISO/IEC 14598-6, 1999) – Fornece orientação para documentação dos módulos de avaliação, os quais possuem as especificações do modelo de qualidade, informações e dados sobre a aplicação prevista e real do modelo.

Basicamente o Processo de Avaliação segundo a norma ISO/IEC 14598-1, pode ser descrito de acordo com o diagrama abaixo.

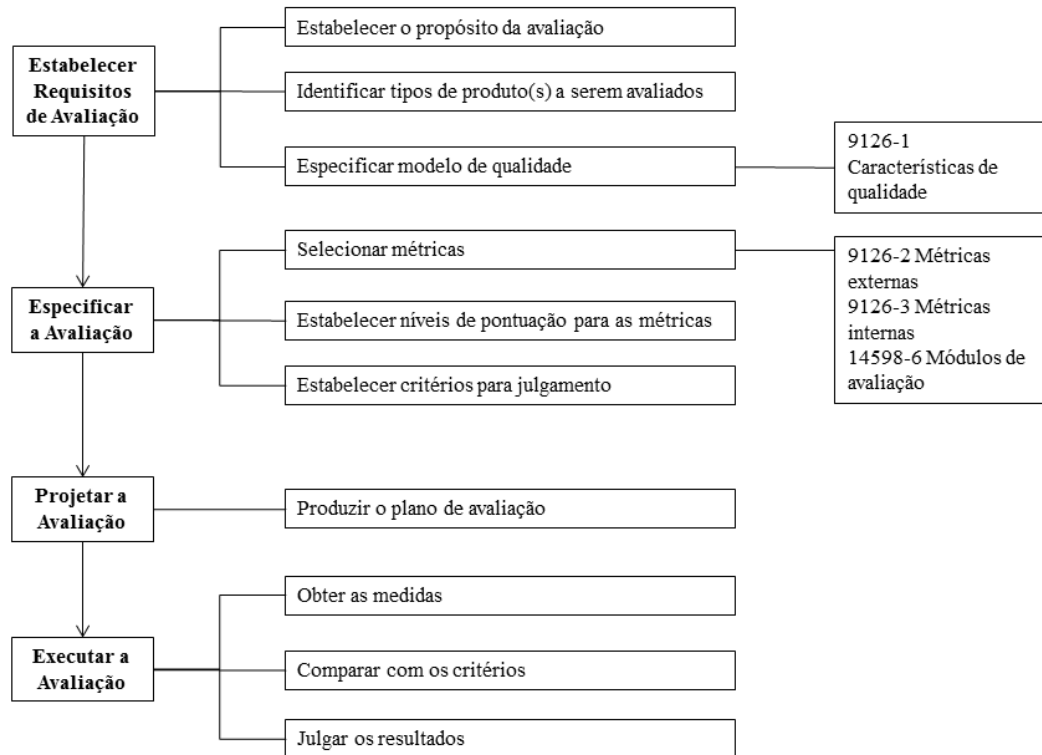


Figura 3 - Processo de avaliação segundo a ISO/IEC 14598-1, sendo subsidiada pela ISO/IEC 9126 (ISO/IEC 14598, 1999).

Na primeira etapa “Estabelecer Requisitos de Avaliação”, as atividades consistem em estabelecer o propósito da avaliação (explicando quais os objetivos da avaliação), identificar os tipos de produtos (definindo qual o tipo do produto a ser avaliado, se são produtos intermediários ou finais) e especificar o modelo de qualidade, selecionando as características de qualidade relevantes e utilizando um modelo de qualidade que consiga apresentar as diferentes características do software.

Na segunda etapa, “Especificar a avaliação”, as atividades consistem em selecionar as métricas a serem utilizadas de acordo com as características do produto de software (de acordo com as normas ISO/IEC 9126-2, 9126-3 e 14598-6), estabelecer os níveis de pontuação para as métricas de acordo com uma escala determinada, e estabelecer os critérios para o julgamento da qualidade do produto, lembrando que cada resultado referente a cada característica deve estar sintetizado.

Na terceira etapa, “Projetar a avaliação”, é onde é produzido o plano de avaliação onde estará descrito os recursos e suas respectivas distribuições dentre as existentes para que a avaliação seja realizada.

Na quarta etapa, “Executar a avaliação”, ocorrerá a avaliação propriamente dita, onde, de acordo com os requisitos de avaliação estabelecidos nas fases anteriores, as métricas selecionadas serão aplicadas, disponibilizando os valores das escalas das métricas, o valor observado, será comparado ao especificado na avaliação, onde ao fim, após tal julgamento, será disponibilizada uma declaração sobre o quanto o software atende aos requisitos de qualidade.

2.5 RELAÇÃO ENTRE AS NORMAS ISO/IEC 9126 (9126-1, 9126-2, 9126-3 E 9126-4) E ISO/IEC 14598 (14598-1, 14598-2, 14598-3, 14598-4 E 14598-5)

A família de normas ISO/IEC 9126 trata da qualidade do produto de software, sendo analisados os principais itens que possam confirmar a acurácia do software quando este for aplicado. Já a família de normas ISO/IEC 14598, aborda a avaliação propriamente dita, de acordo com os critérios estabelecidos na primeira família de normas citada.

A associação entre as duas famílias de normas a ISO/IEC 9126 (9126-1, 9126-2, 9126-3 e 9126-4) e a ISO/IEC 14598 (14598-1, 14598-2, 14598-3, 14598-4 e 14598-5), está descrita no diagrama abaixo.

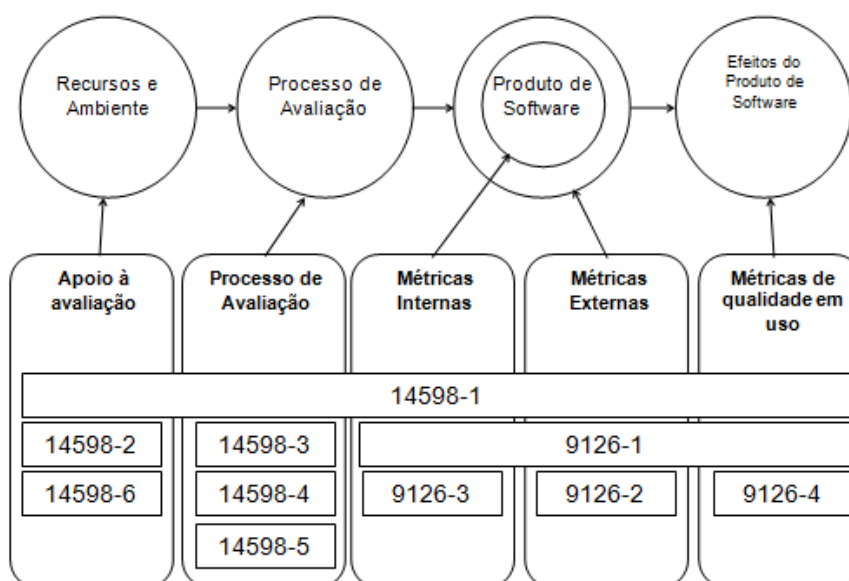


Figura 4 - Relação entre a ISO/IEC 9126 e ISO/IEC 14598 (NETO, 2005).

2.6 SOFTWARE EDUCACIONAL

Software educacional é definido como sendo um sistema computacional e interativo, intencionalmente concebido para facilitar a aprendizagem de conceitos específicos (KELLY, 2008). Segundo citado em (TEIXEIRA, acessado em 20/02/2014), todo software pode ser considerado um software educacional, desde que este, utilize metodologias que o contextualizem no processo de ensino-aprendizagem, ou seja, dependendo diretamente da criatividade dos seus usuários (nesse caso os professores).

De acordo com (GALVIS, 1998), os softwares educativos, podem ter diversas classificações, por exemplo, quando se trata do tipo de utilização, pode-se relatar a categoria Tutorial (guia o aluno por diferentes fases de aprendizagem, de forma sequencial), Exercitação e Prática (se destaca por oferecer subsídios suficientes para memorização e revisão dos conteúdos já estudados) e Simuladores e Jogos Educativos (permitindo aproximação da realidade sem necessitar da exploração in loco).

É também abordado em (GALVIS, 1998) as categorias existentes quando se trata da atividade do sujeito de aprendizagem, sendo dos tipos Algoritmo e Heurístico. O tipo

Algoritmo, é centrado apenas na transmissão do conhecimento, por meio de atividades sequenciadas, já no tipo Heurístico, predomina a aprendizagem experimental, onde o aluno explora um ambiente com situações-problema sobre o assunto o qual deseja estudar, e por meio dessas, constrói seu conhecimento.

Também existe a classificação feita por (REGGINI 1990), que trata das categorias existentes quanto se trata do uso do computador, sendo as modalidades Dura e Branda, onde a Dura direciona o usuário apenas a perguntas e respostas, e a modalidade Branda, onde o aluno comanda a atividade e até os erros são aproveitados, como pontos de reflexão.

Porém, mesmo com tantas possibilidades de utilização de softwares educativos no contexto educacional atual, tal prática não é tão trivial. É defendido em (TEXEIRA E BRANDÃO, 2003) que a utilização do computador e sua gama de recursos na educação só é de fato notável a partir do momento em que os professores os utilizam como uma ferramenta de auxílio as suas atividades didático-pedagógicas, também como um instrumento de planejamento e realização de projetos interdisciplinares, um elemento que motiva e desafia o surgimento de novas práticas pedagógicas, transformando o processo de ensino-aprendizagem em algo inovador, dinâmico, e interativo.

Quando considerado por (ROCHA E CAMPOS, 1993) as fases de desenvolvimento do software educacional, estas fases são diferenciadas das fases de desenvolvimento das demais taxonomias existentes. Isso ocorre principalmente pela necessidade existente de características e atributos especiais quando se trata de softwares educacionais. Estas características e atributos especiais são os que farão a diferença diante do processo de ensino-aprendizagem do aluno que irá utilizar tal software.

Também é afirmado em (ROCHA E CAMPOS, 1993) que tais características devem ser definidas crucialmente na fase de requisitos, fase esta onde também deve ser definida o modelo de aprendizagem que será utilizado.

O ponto crucial do software educacional é o impacto que o mesmo irá gerar no seu usuário. Apesar do mesmo não ser considerado um software crítico, pelas suas características principais (contexto, usuário e tarefa), tal software poderá influir positivamente ou negativamente no conhecimento que pretende que seja adquirido pelo alunado, gerando impactos que o mesmo carregará pelo resto da sua vida acadêmica, e até profissional.

2.6.1 TIPOS DE SOFTWARE EDUCACIONAL DE ACORDO COM A FUNÇÃO EDUCATIVA.

Segundo (GIRAFFA, 2009), os softwares também podem ser classificados de acordo com as funções educativas que assumem de acordo com a relação abaixo.

- **Exercício e Prática**
Consiste na repetição sistemática de exercícios por parte do aprendiz. É caracterizado pela realização de questionamentos sobre temas estabelecidos.
- **Tutoriais**
Consiste em informações organizadas de acordo com uma sequência pedagógica particular e apresentada ao estudante. A interação resume-se na leitura de textos e outras informações dispostas.
- **Simulação e Modelagem**
Caracteriza-se pela representação de fenômenos que não são possíveis de realizar devido à possíveis riscos/custos, o que demandam longos tempos de processamento
- **Sistemas Tutores Inteligentes**
Utilizam técnicas de Inteligência Artificial para proporcionar a instrução adaptada ao aluno (em conteúdo e forma). A estratégia de ensino/aprendizagem é ajustada ao conteúdo e as características do aprendiz
- **Jogos Educativos**

Propõem a utilização de recursos visando motivação e geração de desafios ao aprendiz, por meio de competição entre aluno - aluno, ou aluno - máquina.

2.6.2 QUALIDADE DE SOFTWARE EDUCACIONAL

Foi realizada uma análise em 19 (dezenove) artigos científicos, publicados em periódicos e eventos de abrangência nacional e internacional, onde todos abordavam a produção, utilização e aplicabilidade de softwares educacionais. A tabela abaixo visa a apresentação das abordagens de cada artigo.

Tais artigos foram selecionados, principalmente por tratarem de desenvolvimento do software educacional, suas fases, artefatos, papéis entre outras características relevantes à engenharia de software. Bem como, foram selecionados a partir de repositórios de eventos que abordam a produção de software, assim como especificadamente software educacional.

Ord.	Referências	Abordagem/Objetivo	Publicação/Ano
1	[Giraffa, Marczak e Prikładnicki 2005]	Apresentar a necessidade de uma metodologia adequada para desenvolvimento de software educacional.	WIE/2005
2	[Oliveira et. al. 2007]	Apresentar um processo de desenvolvimento de software distribuído, com garantia de qualidade, baixo custo e agilidade.	WDDS/2007
3	[Ivan e Josue 2007]	Introduzir práticas efetivas em uma metodologia alternativa de desenvolvimento de software educacional, podendo ser utilizado tanto de forma presencial, como em EAD	IEEE/2007
4	[Rangel, Cury e Menezes 2011]	Apresentar contribuições no que se diz respeito a modelagem de ambientes educacionais virtuais, utilizando-se dos recursos tecnológicos disponíveis	SBIE/2011
5	[Fioravanti, Nakagawa e Barbosa 2010]	Apresentar arquitetura de referência para um ambiente educacional baseado nas funções de apresentar e disponibilizar material, comunicar e conduzir atividades, avaliar aprendizes e gerenciar e coordenar cursos.	SBIE/2010
6	[Cavalcante e Domingues 1998]	Apresentar a aplicabilidade de uma metodologia para o desenvolvimento de softwares educativos.	RBIE/1998
7	[Kam et. al. 2008]	Apresentar as etapas de	ACM/2008

		desenvolvimento de aplicativo educacional para ensino de idioma, destinado a dispositivos móveis em áreas carentes da Índia.	
8	[Ehteshami et. al. 2013]	Apresentar a utilização de softwares educativos em dispositivos móveis junto à alunos do curso de medicina.	ACTA/InforMED/2013
9	[Casali et. al. 2012]	Apresenta metodologia de escrita de livros-texto colaborativa, utilizando ambientes virtuais.	
10	[Anderson e Dron 2011]	Apresenta as três gerações da pedagogia da Educação à Distância, seus respectivos focos, metodologias de desenvolvimento e aplicabilidade dos produtos.	International Review of Research in Open and Distance Learning/2011
11	[Das 2007]	Apresentar pesquisa feita com profissionais de desenvolvimento de software acerca do envolvimento dos usuários na área de engenharia de requisitos.	INCT/2007
12	[Dantas et. al. 2012]	Apresentar avaliação de software destinado ao ensino de Programação baseado nas normas de Reeves e ISO/IEC 9126.	CONNEDI/2012
13	[Valle et. al. 2013]	Apresentar metodologia de avaliação de Jogos Educacionais Digitais (JED) baseado na avaliação heurística, adaptando as normas existentes e criando novas de acordo com a necessidade do contexto.	TISE/2013
14	[Sganzerla e Geller 2013]	Apresentar tecnologias assistivas computacionais que auxiliem no ensino de matemática para cegos	TISE/2013
15	[Meier et. al. 2013]	Proposta de desenvolvimento de objeto de aprendizagem computacional, destinado ao ensino de geometria	TISE/2013
16	[Menezes e Schlemmer 2013]	Buscou compreender como as interações dos participantes com o jogo poderiam favorecer o desenvolvimento da competência comunicativa, bem como a aceitação do uso do jogo usado como entretenimento para fins educacionais	TISE/2013
17	[Chaves et. al. 2013]	Apresentar o MOJO, como ferramenta de automatização do trabalho de correção de atividades provenientes das disciplinas de computação	TISE/2013
18	[Boocchese et. al. 2013]	Apresenta Objeto de Aprendizagem computacional, que aborda temas relacionados ao paralelismo sintático e semântico na Língua Portuguesa, por meio de perguntas condutoras e atividades práticas.	TISE/2013
19	[Melo e Neves 2013]	Busca investigar o potencial dos softwares educacionais livres do projeto	TISE/2013

		KDEdu para a utilização nos anos iniciais do ensino fundamental, além de apresentar contribuições e sugestões para o seu aperfeiçoamento	
--	--	--	--

Tabela 13 - Relação de Trabalhos Científicos analisados e suas respectivas abordagens adaptada a partir de (LIMA, 2013).

Conforme a tabela abaixo é possível observar a quantidade de trabalhos que foram publicados, porém, sem nenhuma metodologia destinada à aferição da qualidade do mesmo.

Referências	Possui avaliação ?	
	Sim	Não
[Giraffa, Marczak e Prikladnicki 2005]		X
[Oliveira et. al. 2007]		X
[Ivan e Josue 2007]	X	
[Rangel, Cury e Menezes 2011]		X
[Fioravanti, Nakagawa e Barbosa 2010]	X	
[Marczak et. al. 2003]		X
[Cavalcante e Domingues 1998]		X
[Kam et. al. 2008]		X
[Ehteshami et. al. 2013]		X
[Casali et. al. 2012]		X
[Anderson e Dron 2011]		X
[Das 2007]	X	
[Dantas et. al. 2012]	X	
[Valle et. al. 2013]	X	
[Sganzerla e Geller 2013]		X
[Meier et. al. 2013]		X
[Menezes e Schlemmer 2013]		X
[Chaves et. al. 2013]		X
[Boocchese et. al. 2013]	X	

[Melo e Neves 2013]	X	
---------------------	---	--

Tabela 14 - Verificação da existência (ou não) de aferição de qualidade nos produtos gerados na relação de trabalhos da Tabela 13.

Dos artigos que dispunham de aferição de qualidade de software, foi destacada a presença das metodologias baseadas na norma ISO 9126-1 (descrita anteriormente). Também houve a utilização da metodologia MOSCA (Modelo Sistemico de Calidad), que é baseado na norma ISO 15504. Foi possível encontrar também, em apenas um trabalho, a utilização do modelo de Reeves.

Além dos artigos descritos na tabela acima, também foram analisados os seguintes trabalhos de conclusão de curso do Centro de Informática da UFPE, Avaliação de Qualidade de um Produto de Software (NASCIMENTO, 2010) e Um Processo para Avaliação de Produtos de Software Através de Análise por Especialista (NETO, 2005), onde ambos também utilizaram a família de normas ISO/IEC 9126, e a associaram com a família de normas ISO/IEC 14598.

Diante da análise da literatura apresentada, observa-se que, muitos dos artigos descreveram que possuem métodos e práticas que conotam a avaliação da qualidade do software, mas muitas vezes, estes, não transparecem qual metodologia foi utilizada, ou até o embasamento que estes possuem para saber da qualidade ou não do software a ser utilizado.

O grande problema existente, é que, mesmo com a gama de normas e padrões que podem auxiliar no processo de avaliação de qualidade de software disponíveis atualmente, estes não são utilizados, pois, para muitos profissionais da área de desenvolvimento, a avaliação da qualidade, seria apenas um adendo, e não um essencial no que se diz respeito ao produto de software propriamente dito. (NETO, 2005)

Quando tratamos de aferição de qualidade de software educacional, observamos uma falta de padronização nos processos.

De acordo com o observado, percebemos que os artigos que utilizaram a norma ISO 9126-1, conseguiram contemplar os principais pontos para avaliação do software, de acordo com o ciclo de vida abaixo apresentado.

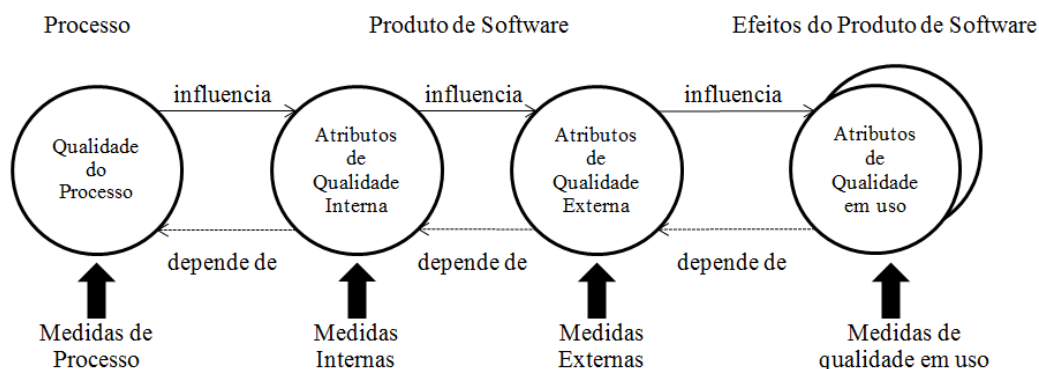


Figura 5 - Ciclo de vida da norma ISO/IEC 9126 – 1.

A principal problemática encontrada em relação à avaliação da qualidade de software baseia-se primeiramente no fato de que, a grande maioria dos artigos analisados, conforme demonstrado na tabela 1, não teve a preocupação de utilizar nenhuma metodologia de avaliação de qualidade, tornando esse um ‘adicional’, ao invés de um ‘essencial’, que de fato, a garantia da qualidade de software, permite, antes de mais nada o desenvolvedor ter a certeza que o produto final, irá atingir as expectativas do cliente, de acordo com os requisitos levantados na fase inicial do projeto de desenvolvimento.

Quando se trata de software educacional, o adendo “educacional” carrega consigo uma carga superior ao que se pensa. Um software educacional influi diretamente no processo de aprendizagem do aluno, onde, muitas vezes o que se espera é que o software ensine ou auxilie o aluno na sua aprendizagem, podendo este software chegar a manter o aluno dependente de uma ferramenta, ou seja, em determinadas situações, o aluno não conseguirá solucionar problemas recorrentes, sem a ferramenta consigo.

Observamos nas metodologias de avaliação de qualidade de softwares, diversas métricas, fatores e sub-fatores, para auxiliar na mensuração quantitativa e qualitativa do produto de

software, tornando-o mais fácil de compreender tanto para o desenvolvedor quanto para o cliente. Porém, pode-se dizer que um dos problemas existentes, se dá pelo fato de, o âmbito educacional ser diferente dos demais (empresarial, comercial, entre outros), pelo fato da quantidade de atores envolvidos. O professor, o aluno, o coordenador pedagógico, o diretor, todos estes (ou parte destes), devem ser consultados quando forem ser adquiridos os dados necessários para avaliação da qualidade do software.

As visões, os conceitos agregados e as expectativas dos stakeholders do software educacional, apesar de objetivarem o mesmo fim, que é a educação de qualidade, são distintos, onde, o que explica muitas vezes, o fracasso da popularidade dos softwares dispostos, a exemplo daqueles disponibilizados pelo governo federal, por meio do Programa Proinfo.

3. QUALI-EDU: UM MODELO DE QUALIDADE DE SOFTWARE EDUCACIONAL

Após pesquisa efetuada em periódicos e anais de eventos, foi constatado conforme Tabelas 13 e 14, a ausência de métodos e práticas que contemplem a aferição da qualidade de software, seja este educacional, ou destinado a outros fins. Foi perceptível que, para grande maioria dos trabalhos apresentados, a quantidade foi mais enfatizada que a qualidade sendo este segundo ponto, não considerado um item primordial (como deveria ser de fato), mas apenas como sendo um adicional ao produto de software finalizado.

Quando tratamos de qualidade de software, buscamos assegurar o grau com que um sistema, componente ou processo cumpre os requisitos especificados e as necessidades ou expectativas do cliente (ROCHA E CAMPOS, 1993). O software em si, deve apresentar características de que foi desenvolvido profissionalmente, adequando-se aos requisitos de desempenho, expondo que os requisitos funcionais estão sendo atingidos e com documentação compreensível e acessível ao usuário.

Quando se trata de software educacional, os cuidados emanados para essa taxonomia devem ser superiores ao que se propõe atualmente. Segundo (VIEIRA, 2000) um software educacional deve ser utilizado de forma responsável, onde os fins pedagógicos devem cumprir com seu dever, não tornando o software apenas uma diversão, e sim uma forma de ensino.

De acordo com (GIRAFFA, 2000), ao mesmo tempo em que este software pode auxiliar no processo de ensino-aprendizagem, pode também prejudica-lo, expondo uma preocupação que deve haver, acerca desse pensamento inicial. Para tal, se faz necessário que as metodologias estejam adequadas, o professor cumpra com seu papel, avaliando os produtos que chegam a ele, considerando os aspectos técnicos, pedagógicos, psicológicos, políticos e sociológicos, quando se trata da informática educativa.

O que é proposto neste capítulo é um processo de avaliação de software que auxilie na avaliação de um software educacional, para que este software possa cumprir com os pressupostos psicopedagógicos que regem o processo de ensino-aprendizagem do ensino convencional, não assistido por computadores. Segundo (GLADCHEFF, 2002), tal tarefa é de grande complexidade, pois a educação comum é composta por várias disciplinas, cada uma destas com suas peculiaridades.

Para suprir tais demandas, não pode-se averiguar corretamente a qualidade do software, apenas tomando por base as qualidades técnicas dispostas nas normas e regras citadas anteriormente. Isto acontece, pois na educação, diversas outras qualidades/contextos devem ser levados em consideração, apenas visando o avanço na qualidade do ensino atual. Tais qualidades e contextos são os aspectos educacionais envolvidos (pedagogia, psicopedagogia, sociologia, cultura, cognição e ludicidade).

Pelo fato de a grande maioria dos professores não deterem conhecimento acerca de aspectos educacionais, uma equipe multidisciplinar é de grande valia, principalmente com a

presença do licenciado em computação, enquanto pessoa competente tanto em termos técnicos, quanto pedagógicos.

Na maioria das metodologias de qualidade de software analisadas, a presença do *stakeholder* principal, ou seja, o cliente final, não foi levada em consideração. A figura do avaliador, nem sempre é descrita como sendo o usuário, ou sequer este é citado no processo de avaliação da qualidade do produto de software.

No meio educacional, pode-se considerar como principais *stakeholders*, os utilizadores finais do produto, ou seja, os personagens do professor e do aluno. Apesar de que, o produto não é adquirido por estes, muitas vezes, por não serem considerados, acabam por não se adequarem/agradarem do produto, tornando-o inutilizável.

Segundo (HINOSTROZA & MELLAR, 2001) a literatura sobre o software educacional tem criticado a qualidade destes, por não atender as expectativas dos profissionais de ensino, pelo fato do impacto epistemológico não condizer com o esperado (SHNEIDERMAN, 1998).

Apesar das famílias de normas ISO/IEC abordadas nesse trabalho (9126 e 14598), apresentarem ampla contemplação dos pontos qualitativos cruciais de um software, quando inseridas no contexto do software educacional, não existem métricas para a aferição da qualidade do lado pedagógico do software.

Para isso, este trabalho propõe o acréscimo, baseados nas famílias das normas trabalhadas, dos pontos que possam sanar a problemática da aferição da qualidade pedagógica do software educacional, visto que, os termos técnicos, já são abordados nestas.

3.1 ESTRUTURA GERAL DO PROCESSO

O processo descrito nessa fase do trabalho busca, principalmente, envolver os usuários finais do Software Educacional propriamente dito, são estes, o professor e o aluno.

Considerando que, nos processos apresentados no Capítulo 2, a ausência destes personagens foi amplamente visível, o que se busca ao envolver estes, é uma melhor aproximação ao contexto de utilização deles, evitando problemas do tipo: insatisfação, falta de motivação para utilização do produto e até a banalização deste produto, que pode e deve ser visto como uma ferramenta de grande apoio ao processo de ensino-aprendizagem.

O processo foi todo elaborado, baseado na família de normas ISO/IEC 9126 e na ISO/IEC 14598, ambas apresentadas nas seções 2.3 e 2.4. As referidas normas conseguem contemplar amplamente, com seus módulos de aplicação, os itens cruciais para o bom funcionamento de um software, porém, não conseguem abordar as questões que possam vir a ser levantadas pelos professores e alunos, quando estiverem sendo aplicadas em condições reais de sala de aula.

Conforme citado na descrição deste processo, o que se busca é a aproximação, o máximo possível, do usuário final (professor/aluno), para que, de fato, as necessidades e expectativas dos mesmos possam ser contempladas. Estas características seguirão as citadas na família de normas 9126 (ISO/IEC 9126, 2002). As características a serem avaliadas serão a Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade, Portabilidade, Eficácia, Produtividade e Satisfação.

Os responsáveis pela avaliação são denominados por este trabalho como Equipe de Desenvolvimento (tratada como E.D.), composta basicamente pelo Programador, Designer Instrucional e Especialista do Domínio (podendo comportar mais componentes dependendo da complexidade do Software Educacional), e pela Equipe de Usuários (trataremos como E.U.), que será composta primordialmente por um grupo de Professores (que estejam em exercício principalmente), alguns alunos destes professores, e um Coordenador Pedagógico. A figura do Licenciado em Computação é de grande valia nesse processo, levando em consideração que o mesmo pode servir como ponte de comunicação entre as equipes citadas.

Também deverá existir a presença do especialista em qualidade de software (Avaliador), este será responsável por guiar o processo entre as partes envolvidas (a E.U. e E.D.), visto que, por ser o melhor entendedor do modelo de qualidade de software que será aplicado, o avaliador, dará o julgamento final, por meio de documento final de avaliação, que poderá ser conferido nos anexos.

Para cada fase haverá um artefato para facilitar o correto acompanhamento dos dados à serem analisados. Estes artefatos serão compostos por questionamentos que indiquem o cumprimento (ou não) de determinadas características. O peso direcionado à esse questionamento, e a atribuição da nota, ocorrerão conforme a descrição nas tabelas abaixo.

Peso	Grau de importância
2	Baixo
4	Médio
6	Alto

Tabela 15 - Pesos e seus respectivos graus de importância (BRATEFIXE, 2013).

Nota	Significado
0	Não cumpriu com o questionamento
1	Cumpriu com o questionamento

Tabela 16 - Notas e seus respectivos significados (BRATEFIXE, 2013).

Os pesos e as notas serão atribuídos pelos avaliadores, de acordo com a necessidade que os mesmos verificarem. Caso o avaliador de determinada sub-característica imagine que aquela é de pouca importância, o mesmo atribuirá o valor “2” para esta, assim como caso o mesmo ache que é de grande importância, atribuirá o valor “6”, e o caso de cumprir ou não o questionamento feito, serão atribuídos respectivamente os valores “1” ou “0”.

As variáveis “S” dizem respeito às notas de cada questionamento (0 ou 1), e as variáveis “P”, dizem respeito aos pesos de cada um destes (2, 4 ou 6). Já no denominador da função, será feito um somatório de todos os pesos estipulados, onde após ser efetuada uma divisão simples, teremos a nota do software diante dos questionamentos respondidos e os pesos estipulados pelos avaliadores.

Os valores foram descritos dessa forma, para a nota final fosse conseguida utilizando a seguinte fórmula, utilizada por (MARÇAL, 2009):

$$\text{Nota Final} = \frac{S_1 * P_1 + S_2 * P_2 + S_3 * P_3 + \dots + S_n * P_n}{\sum_{i=1}^n}$$

De acordo com as notas obtidas, software pode ser classificado da seguinte forma:

Classificação	Valor da Nota Final	Julgamento
Excelente	0,91 a 1,0	Aceito
Bom	0,76 a 0,90	Aceito
Satisfatório	0,66 a 0,75	Aceito com restrições
Regular	0,51 a 0,65	Necessita verificações
Insatisfatório	0,0 à 0,5	Rejeitado

Tabela 17 - Julgamentos e classificações de acordo com valores de notas finais obtidas nos questionários (MARÇAL, 2009).

É notório que, para que o processo de avaliação consiga níveis satisfatórios, se faz necessária uma boa descrição do contexto de aplicação do software, seus usuários e as tarefas que se esperam.

O processo de avaliação também será baseado nas metodologias ágeis de desenvolvimento de software (MANIFESTO FOR AGILE SOFTWARE, 2001), considerando que os processos ágeis, focam na diminuição da distância entre cliente e desenvolvedor, sendo os clientes considerados como parte da equipe de desenvolvimento, contribuindo com informações desde o início do processo. A comunicação deve ser enfatizada, de modo que seja ativa e contínua entre todos os membros que compõe a equipe de desenvolvimento (PRESSMAN, 2006).

A presença do cliente final (no nosso caso, professor e aluno) é de grande valia, principalmente nesta fase da execução da avaliação, pois, ninguém melhor que estes atores para descrever se o software educacional contemplará o que é esperado.

3.2 DESCRIÇÃO DAS CARACTERÍSTICAS A SEREM ANALISADAS

Abaixo será descrito precisamente por meio dos artefatos que serão utilizados o que foi citado no item 3.1, levando em consideração as atividades (características a serem avaliadas), seus respectivos responsáveis.

Deve ser lembrado que o campo “Peso”, estará em branco pelo fato de o avaliador ser o responsável por preenche-lo, de acordo com o que o mesmo observar em termos de pontos relevantes à serem avaliados no software. Da mesma forma, o campo “Nota”, pois esta será também preenchida pelo usuário que avaliar o software.

Nos artefatos que serão apresentados abaixo, caso o responsável pelo preenchimento do mesmo, perceba que as informações que foram descritas por meio das notas e pesos não foram suficientes para expressar seu real julgamento sobre a característica em geral ou uma sub-característica em especial, o mesmo terá o campo “Informações Adicionais”, onde o avaliador poderá descrever algo que achar pertinente que possa vir a auxiliar na conclusão do processo QUALI-EDU.

As características abaixo descritas foram explanadas na seção 2.3, que trata diretamente da família de normas ISO/IEC 9126, mais precisamente no documento ISO/IEC 9126-1.

A) Funcionalidade

- Entrada: Documento de requisitos e Software em condições de uso (finalizado ou protótipo).
- Saída: Artefato preenchido.

Característica Funcionalidade	Sub-característica	Questionamento	Responsável	Peso	Nota
	Adequação	Provê funções apropriadas para as tarefas/objetivos especificados?	E.U.		
	Acurácia	Provê precisamente o resultados corretos de	E.U.		

		acordo com o requisitado?			
	Interoperabilidade	Consegue interagir com outros sistemas?	E.D.		
Informações Adicionais					

Diante das sub-características originais, foram selecionadas estas, devido a importância do software suprir os questionamentos supracitados. Por se tratar de software educacional, o mesmo necessita de coerência no que é apresentado, pois caso os resultados não possuam a exatidão esperada, influirá diretamente na aprendizagem do usuário final (aluno).

B) Confiabilidade

- Entrada: Documento de requisitos e Software em condições de uso (finalizado ou protótipo), diagrama de casos de uso.
- Saída: Artefato preenchido.

	Sub-característica	Questionamento	Responsável	Peso	Nota
Característica Confiabilidade	Maturidade	É capaz de evitar falhas de defeitos no software?	E.D.		
	Tolerância à falhas	É capaz de manter desempenho especificado mesmo com defeito no software?	E.D.		
	Recuperabilidade	É capaz de restabelecer desempenho/recuperar dados após falha?	E.D.		
Informações Adicionais					

Diante das sub-características originais, foram selecionadas estas, devido a necessidade do software ser capaz de evitar falhas, e caso estas existam, o software possa recuperar dados e manter o desempenho de acordo com o estipulado nos requisitos do

software. Para um software educacional, tal característica é de grande importância, pois o usuário final (aluno), precisa se concentrar em como solucionar os problemas educacionais propostos e não os problemas que venham a aparecer no software.

C) Usabilidade

- Entrada: Documento de requisitos e Software em condições de uso (finalizado ou protótipo).
- Saída: Artefato preenchido.

	Sub-característica	Questionamento	Responsável	Peso	Nota
Característica Usabilidade	Inteligibilidade	Permite a compreensão do quão apropriado é o software, e como ele pode ser utilizado para as tarefas especificadas	E.U.		
	Apreensibilidade	É capaz de possibilitar ao usuário aprender sua aplicação?	E.U.		
	Operacionalidade	É capaz de possibilitar o usuário operá-lo e controlá-lo?	E.U.		
	Atratividade	É atraente ao usuário?	E.U.		
Informações Adicionais					

Diante das sub-características originais, foram selecionadas estas, devido à obrigação do software educacional ser atrativo ao aluno e também apreensível. A partir do momento que o aluno se sente atraído pelo software, o mesmo terá interesse em aprender sobre, e dessa forma, o processo de aprendizagem fluirá naturalmente. Deve-se também, ser inteligível, ou seja, que permita que o professor possa verificar o quão adequado o software é para sua realidade.

D) Eficiência

- Entradas: Documento de requisitos e Software em condições de uso (finalizado ou protótipo).
- Saídas: Artefato preenchido.

	Sub-característica	Questionamento	Responsável	Peso	Nota
Característica Eficiência	Comportamento em relação ao tempo	É capaz de prover tempos de resposta/processamento apropriados quando em condições estabelecidas?	E.D.		
	Utilização de recursos	Utiliza quantidades apropriadas de recursos, quando em condições estabelecidas?	E.D.		
Informações Adicionais					

Diante das sub-características originais, foram selecionadas estas, pois o software deve cumprir com os tempos de resposta/processamento (estipulados nos requisitos), e também utilizar os recursos que foram pré-definidos, possibilitando a correta execução do software, evitando assim possíveis pausas durante a utilização do software pelo usuário final (aluno).

E) Manutenibilidade

- Entrada: Documento de requisitos e Software em condições de uso (finalizado ou protótipo) e possíveis alterações estipuladas pela E.U.
- Saída: Artefato preenchido.

	Sub-característica	Questionamento	Responsável	Peso	Nota
Característica Manutenibilidade	Analisabilidade	Permite diagnóstico de deficiências/falhas devido mudanças no ambiente/requisitos especificações?	E.D.		
	Modificabilidade	Permite modificações sem impactos?	E.D.		
	Estabilidade	É capaz de evitar efeitos inesperados devido às modificações?	E.D.		

	Testabilidade	Permite validação após modificações?	E.D.		
Informações Adicionais					

Diante das sub-características originais, foram selecionadas estas, pois na vida útil do software, o mesmo pode necessitar de alterações, independentemente de contexto de utilização. Tais alterações não podem alterar o desempenho nem a integridade do sistema. Deve-se levar em consideração também, a possibilidade de atualizações do sistema, onde ao serem incorporadas ao sistema, não podem gerar falhas.

F) Portabilidade

- Entrada: Documento de requisitos e Software em condições de uso (finalizado ou protótipo).
- Saída: Artefato preenchido.

	Sub-característica	Questionamento	Responsável	Peso	Nota
Característica Portabilidade	Adaptabilidade	É capaz de se adaptar a outros ambientes sem necessidade de outros aplicativos?	E.U.		
	Capacidade de ser instalado	É capaz de ser instalado em determinado ambiente?	E.U.		
	Coexistência	É capaz de compartilhar recursos com outros softwares?	E.D.		
	Capacidade para substituir	Capaz de substituir outro software existente com o mesmo propósito/ambiente?	E.U.		
Informações Adicionais					

G) Aspectos Educacionais

De acordo com (MORIN, 2002) o conhecimento contextualizado, pelo fato de ficar mais próximo da realidade dos alunos, torna a adaptação deles em qualquer disciplina mais

fácil. É de suma importância que o aluno esteja na sua zona proximal do conhecimento, que ele possa aliar o “conhecimento novo”, com “conhecimentos velhos” os quais já estão internamente incorporados pelo aluno.

Deve-se considerar ainda segundo (TAPIA E FITA, 2000) que é de grande importância que o software motive o aluno à utilizá-lo, considerando que o processo de motivação é um dos fatores que está intimamente ligado à aprendizagem,

Outro processo ligado à aprendizagem é a motivação intrínseca, que não resulta no treino ou instrução, mais pode ser influenciada pela ação do professor considerando as experiências dos alunos, ou seja, o contexto o qual estes estão inseridos (TAPIA E FITA, 2000).

Diante do que foi defendido pelos teóricos (MORIN, 2002) e (TAPIA E FITA, 2000), e também considerando que o foco do trabalho é avaliação de software educacional, foi adicionada uma nova característica a ser avaliada/verificada que tomará como base a necessidade eminente do foco no cliente, a característica abaixo, será avaliada completamente pela E.U.

- Entrada: Documento de requisitos e Software em condições de uso (finalizado ou protótipo), além de informações sobre o contexto o qual o software será inserido
- Saída: Artefato preenchido.

Característica Aspectos Educacionais	Sub-característica	Questionamento	Responsável	Peso	Nota
	Coerência	O software é coeso, de acordo com a vertente pedagógica pré-determinada?	E.U.		
	Contextualização	O software consegue ser eficiente ante ao contexto que está sendo aplicado?	E.U.		
	Motivação	O software motiva o aluno a aprender o assunto proposto?	E.U.		
	Satisfatoriedade	O software foi bem	E.U.		

		aceito pelos professores/alunos?			
	Adequação à faixa etária	O software é adequado à faixa etária do usuário final?	E.U.		
Informações Adicionais					

3.3 FASES DA AVALIAÇÃO

A avaliação compreenderá três fases principais, conforme figura abaixo. Maiores informações sobre cada fase estarão descritas nos tópicos abaixo.



Figura 6 - Fases da avaliação do Software Educacional descrito no modelo QUALI-EDU.

3.3.1 FASE 1 – MOTIVAÇÃO DA AVALIAÇÃO

Nessa primeira fase, deve ser definido claramente qual o propósito da avaliação. De acordo com a norma ISO/IEC 14598 descrita em (ISO/IEC 14598, 1998), o software pode ser avaliado enquanto ainda em fase de construção (protótipo), e obviamente também quando este está concluído.

Na primeira situação, existe a possibilidade de destacar como principais propósitos:

- Verificar a aceitação do produto;
- Definir o tempo/critério para finalização de determinados processos assim como a entrada em processos sucessores;

- Prever/estimar qualidade do produto quando finalizado.

Na segunda situação, que trata do produto finalizado, existe a possibilidade de destacar como principais propósitos:

- Decidir sobre a aceitação do produto e quando liberá-lo;
- Análise dos efeitos positivos e negativos da utilização do software que está sendo avaliado;
- Decidir sobre a necessidade de aprimoramentos/substituição do software.

Nessa fase, também serão definidos o modelo de qualidade que será utilizado (no nosso caso o QUALI-EDU), além de que tipo de software que será avaliado (no nosso caso, o software educacional).

Vale ressaltar também, que, esses propósitos podem ser selecionados para as mais diversas situações, sendo possível a escolha de um ou de um grupo destes, dependendo principalmente do que se espera ser avaliado.

Haverá ao fim dessa etapa, um documento simples, com modelo nos anexos deste trabalho, preenchido pelo Avaliador geral, diante do coletado em reunião com a E.U. e a E.D., relatando precisamente quais os propósitos da avaliação da qualidade de software.

3.3.2 FASE 2 – ESPECIFICAÇÕES DA AVALIAÇÃO

Essa fase é a responsável por tratar das métricas que serão utilizadas, personagens que atuaram na avaliação e suas respectivas pontuações e critérios de julgamentos destes.

A métrica tipicamente envolve a produção de uma pontuação em alguma escala específica. Cada medida destas métricas contribuem para um julgamento mais geral do produto, claro que sempre de acordo com o contexto do mesmo. (CORTES, 2001)

Deve-se haver um procedimento que seja capaz de sumarizar os resultados da avaliação das características solicitadas, um bom procedimento é a utilização de funções matemáticas para tal tarefa.

O QUALI-EDU, consegue contemplar, de acordo com a referida fase, a designação das atividades aos seus respectivos responsáveis, assim como, os cálculos necessários para o julgamento final da qualidade do software por parte do avaliador geral. Assim como, o QUALI-EDU, demonstra o procedimento matemático que comporta a avaliação das características/sub-características escolhidas para julgamento.

Porém, o QUALI-EDU não possui as métricas necessárias para a total efetivação desta fase, sendo este ponto, destinado à trabalhos futuros, buscando a avaliação mais profunda e sucinta dos produtos de software educacional.

3.3.3 FASE 3 - EXECUÇÃO DA AVALIAÇÃO

Nessa fase, deve-se aplicar o que foi planejado na fase das especificidades da avaliação, para que as medidas sejam obtidas, ou seja, pontuações apropriadas sejam alcançadas, para que, diante disto, estas sejam comparadas aos critérios propostos pelo QUALI-EDU.

Os papéis dessa fase serão a E.U. e a E.D, além do Avaliador. Todos da E.U. e da E.D. preencherão seus artefatos individualmente, claro que, caso estas equipes tenham um grande número de usuários, pode-se selecionar uma parcela destes para cumprir tal fase, conforme será explicitado melhor a seguir.

Esta fase é dividida em três momentos principais, conforme a Figura 7, onde a mesma será descrita detalhadamente abaixo:

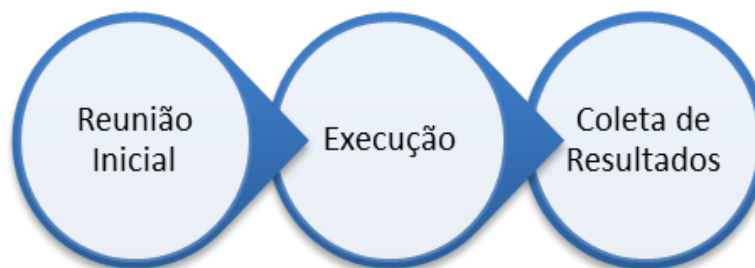


Figura 7 - Momentos da 3ª Fase (Fase de Execução da Avaliação) do QUALI-EDU.

A) Reunião Inicial

Nesse momento inicial, haverá o encontro da E.U., E.D. e do Avaliador, que será o responsável por apresentar o modelo que será utilizado (QUALI-EDU), e explicar o fluxo das informações e sanar qualquer dúvida existente no grupo em geral.

Nesse momento também, ocorrerá a apresentação dos artefatos, para que seja confirmado o entendimento de todos, no que se diz respeito a inserção dos dados nos mesmos. Após o entendimento dos artefatos, o grupo E.U. e E.D., farão a escolha dos pesos para cada sub-característica à ser avaliada, considerando o que for combinado entre estes.

A conclusão dessa reunião se dará com os pesos todos direcionados à cada questionamento dos artefatos. Conforme explicado no início desta seção, cada um dos selecionados responderá seu questionário de avaliação, tendo em comum, apenas os pesos estipulados nesse momento.

B) Execução da Avaliação

Nesse segundo momento, a E.U. e a E.D. já portando seus questionários de avaliação preenchidos, com os pesos determinados no momento anterior, terão acesso ao Software (ou protótipo, dependendo do propósito determinado na Etapa 1 do QUALI-EDU), e também a

algum documento de requisitos que venha explicitar corretamente o que se espera do Software Educacional.

Com o referido material em mãos, os selecionados para a avaliação do software educacional, tanto da E.U. como da E.D., farão o preenchimento das notas do artefato (considerando que os pesos já estarão preenchidos desde a etapa anterior), e o avaliador servirá apenas de orientador neste momento.

Caso os membros da E.U. e da E.D. sintam necessidade maiores informações sobre o que foi preenchido em relação às notas nos artefatos, os mesmos podem informar no campo “Informações Adicionais”, que posteriormente será analisado pelo avaliador.

C) Coleta dos Dados

Neste momento final, o avaliador, munido dos artefatos preenchidos no momento anterior (Execução da Avaliação), fará a análise dos dados, e dará o julgamento final de acordo com a tabela 18, efetuando as operações matemáticas de acordo com a fórmula descrita na seção 3.1.

Para chegar a um resultado final, considerando que o mesmo poderá possuir vários resultados diferentes (diante das notas que foram estipuladas pela E.D. e pela E.U.), o mesmo fará o cálculo da média e do desvio padrão, para que chegue a um resultado final, diante dos vários obtidos.

Após o resultado final obtido, o avaliador, preencherá o Documento final da avaliação da qualidade de software que contemplará a nota final obtida, uma breve justificativa dessa nota (de acordo com as Informações Adicionais preenchidas pelos professores e a visão geral do avaliador).

A conclusão se dará com o julgamento final, ante o estipulado na tabela 18, onde a partir deste julgamento, o software pode ser aceito, ou rejeitado, além de que, pode ser (ou

não) liberado para o uso em sala de aula. O modelo deste documento encontra-se nos anexos deste trabalho.

4. AVALIAÇÃO DO QUALI-EDU

4.1 CONFIGURAÇÃO DO CENÁRIO DE AVALIAÇÃO

4.1.1 INSTRUMENTOS

Os artefatos que serão utilizados, serão as tabelas que foram apresentadas na seção 3.2, e considerando que nesse momento inicial não possuiremos a E.D. , apenas as questões destinadas a E.U. serão respondidas e todos os procedimentos matemáticos que acarretam ao julgamento final do software, explicado na seção 3.1, serão realizados.

4.1.2 EQUIPAMENTOS

Para a nossa avaliação os professores utilizaram o laboratório do Proinfo Urbano de uma escola da rede municipal de ensino, na cidade de Sumé-PB, com computadores multi-estação com a seguinte configuração: Placa-mãe com barramento PCI de 32 bits ou superior, até 4 GB (quatro gigabytes), padrão DDR2 PC2-6400 ou superior, processador Celeron, unidade de disco rígido 320GB, unidade de CD-ROM.

4.1.3 PROFESSORES

Os professores que preencheram os artefatos descritos na seção 4.1.1, são 5 (cinco) professores da rede municipal de ensino da cidade de Sumé-PB. Estes relataram ter conhecimento básico sobre informática, sendo conhecedores de funções corriqueiras de editores de texto e planilhas eletrônicas, além de que dos 5 (cinco), 3 (três) possuem perfis em redes sociais atualmente. Todos relataram possuir computadores em casa, e a quantidade de horas de uso do computador, por semana, oscilou entre 5 horas e 20 horas.

Destes professores, 2 (dois) lecionam matemática no ensino fundamental II e possuem graduação em Licenciatura Plena em Matemática, e 3 (três) possuem graduação em Licenciatura Plena em Pedagogia. Foi relatado por eles também que do grupo, apenas 2 (dois) possuem menos que 10 (dez) anos de docência.

4.1.4 SOFTWARE A SER AVALIADO

O jogo Tux, of Math Command ou simplesmente Tux Math, foi desenvolvido pela empresa Tux4Kids, utilizando a linguagem C e para os padrões gráficos o SDL (Simple DirectMedia Layer). É um jogo destinado à computadores, inicialmente produzido para sistema operacional Linux, porém hoje, funcionando em sistemas operacionais Windows e OS-X. (TUXMATH, acessado em 25/02/2014).

A maior parte do desenvolvimento foi feita em plataformas Linux, onde todas as ferramentas utilizadas para o desenvolvimento do jogo estão disponíveis em qualquer distribuição Linux recente, porém, atualmente já é possível evoluir o jogo em outras plataformas. O jogo possui licença GNU (General Public License). (TUXMATH, acessado em 25/02/2014).

A versão mais atualizada é a 2.0.0 de 14 de abril de 2011, onde os principais recursos são (TUXMATH, acessado em 25/02/2014):

- Suporte multiusuário;
- Tabela de melhores pontuações;
- Modo de treinamento - mais de 50 lições empacotadas que variam da simples digitação de números até todas as quatro operações aritméticas elementares com números negativos e questões de "número ausente (por exemplo: "4 x ? = 20");
- Suporte multiplataforma (Linux, Windows, Mac OS X, BeOS);
- Localização para mais de trinta idiomas;
- Suporta modo de multijogadores pela rede local;

- Missões ou tarefas;

O jogo é baseado no jogo eletrônico “Missile Command” (TUXMATH, acessado em 25/02/2014), desenvolvido pela Atari, uma das principais precursoras no ramo de Video Games em escala mundial. Nesse jogo, busca-se ensinar os alunos de matemática à resolver problemas matemáticos com mais facilidade, com menos abstração e maior ludicidade, confirmando assim a característica de um JED, de acordo o descrito na seção 2.6 deste trabalho.

No jogo, o principal objetivo é que o jogador proteja os seus iglus (levando em consideração o tema Tux), de possíveis cometas que sempre aparecem na tela, caindo em direção destes iglus. Para que o cometa seja destruído, o usuário necessitará verificar quais questões estão aparecendo na tela, para que o mesmo digite a solução e aperte a tecla “enter”.

Após essa ação por parte do usuário, caso a solução esteja correta, o cometa será destruído, caso contrário o disparo do laser, não atingirá nenhum dos cometas que estão caindo, denotando o erro por parte do aluno.

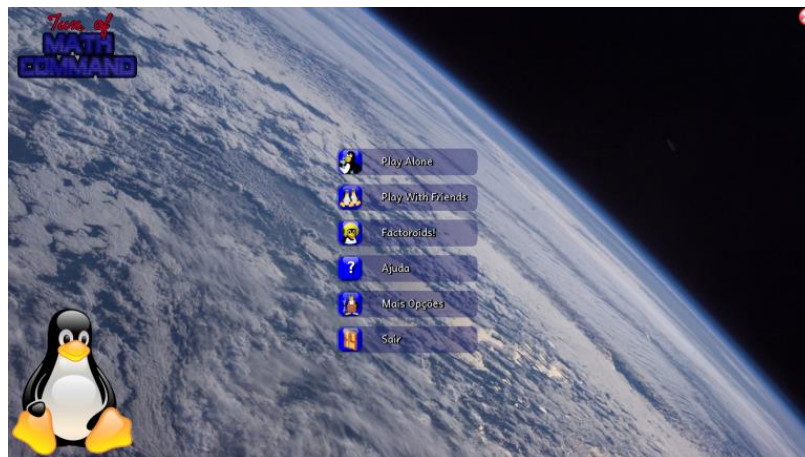


Figura 8 - Menu Principal do Jogo



Figura 9 - Tela Inicial do Jogo



Figura 10 - Jogo em Execução

A primeira versão do jogo foi lançada em 2001 e veio passando por diversas atualizações, a exemplo da versão do 1.7.0. que incluiu o modo multijogador.

4.2 RELATO

Conforme o citado na seção 3.2.3, na fase de execução do QUALI-EDU, haveria uma reunião inicial, onde o modelo seria apresentado, com suas características e artefatos. Dessa forma ocorreu, onde foi apresentado o modelo, e os artefatos.

O propósito, da avaliação foi escolhido pelos professores, e considerando que o software já está pronto, o propósito desta avaliação foi “Decidir sobre a necessidade aprimoramentos/substituição no software”.

Após a decisão do propósito, os 5 (cinco) professores munidos dos artefatos apresentados na seção 3.1, iniciaram a discussão sobre os pesos das questões as quais eram de responsabilidade destes investigar. O resultado da discussão encontra-se na tabela abaixo.

Característica	Questão	Peso
Funcionalidade	Q1	6
	Q2	6
Usabilidade	Q7	6
	Q8	4
	Q9	6
	Q10	4
Portabilidade	Q18	4
	Q20	2
Aspectos Educacionais	Q21	6
	Q22	6
	Q23	6
	Q24	6
	Q25	6
Somatório dos Pesos		68

Tabela 18 - Atribuição de pesos às questões sob a responsabilidade da E.U.

Após a atribuição dos pesos, os professores (ou seja E.U.), já com o conhecimento necessário acerca do modelo QUALI-EDU, individualmente dirigiram-se a cada computador para utilizar o JED TuxMath, e a partir disto, pontuar as notas para concluir a avaliação do software.

Após a fase da execução da avaliação propriamente dita, inicia-se a fase da coleta de dados, para que seja extraído o máximo de informações possíveis de acordo com as respostas dos questionamentos feitos. As pontuações dadas por cada um dos usuários, está descrita na tabela abaixo, onde a variável “U” corresponde ao usuário e as variáveis “Q” correspondem às questões sob a responsabilidade da E.U.:

	Q1	Q2	Q7	Q8	Q9	Q10	Q18	Q20	Q21	Q22	Q23	Q24	Q25
U1	0	0	1	0	1	0	1	0	0	1	0	1	1
U2	1	1	0	1	0	0	0	0	1	0	1	0	0
U3	0	0	0	1	0	1	1	1	1	0	0	1	0
U4	1	0	1	1	0	1	0	1	0	1	1	1	1
U5	1	1	0	0	1	1	0	1	0	1	0	0	1

Tabela 19 - Notas inseridas nos artefatos por cada um dos professores da E.U.

Os resultados finais foram obtidos de acordo com a fórmula matemática descrita em (PRESSMAN, 2006), sendo então os seguintes:

	Resultado final
U1	0,51
U2	0,34
U3	0,29
U4	0,54
U5	0,43

Tabela 20 - Resultados Finais da Avaliação de acordo com (MARÇAL, 2009).

Porém, para o documento final da avaliação da qualidade de software, será realizado o cálculo do desvio padrão, conforme citado na seção 3.2.3. Tal cálculo se faz de suma importância, para que se tenha com precisão e atomicidade um valor que retrate realmente o que foi informado nos questionários pela E.U.

Após o cálculo do desvio padrão, foi obtido o valor 0,56, onde de acordo com os julgamentos descritos na tabela 17, o JED TuxMath, receberá a classificação de “Regular” necessitando assim de verificações antes de chegar ao mercado/salas de aula.

Foi possível observar também, na seção de “Informações Adicionais”, comentários do tipo:

- “O programa é muito simples de usar, o que mais complica é a mistura de idiomas, coisas em inglês e coisas em português podem confundir os alunos”.
- “O programa não é aplicável à nossa realidade, considerando o contexto de interior que possuímos, as crianças praticamente não se concentram nas aulas, e se forem para os

computadores, e verem um programa educacional em inglês, se sentirão desmotivados e procurarão outra opção”.

- “O ‘joguinho’ é bom o suficiente para auxiliar o que o aluno aprende em sala de aula, digamos que como uma tarefa extra-classe, porém fica complicado para eles terem que escolher, se preocupar em entender o jogo, ou se preocuparem em entender como sequer começar o jogo”.
- “Até então não consegui sair do jogo, não é explicativo, não é de fácil compreensão, e no meu ponto de vista, se nós não estamos conseguindo, quem dirá as crianças em sala de aula?”

Portanto, de acordo com o pequeno grupo que avaliou o software, o mesmo pode sim, auxiliar no processo de ensino-aprendizagem de matemática, principalmente considerando a realidade atual, de grande déficit nesse componente curricular. Porém, o TuxMath, necessita de melhorias, para que possa melhor atender a sociedade brasileira, mais precisamente os inseridos no contexto do Nordeste Brasileiro.

5. CONSIDERAÇÕES FINAIS

Foi observado no decorrer do trabalho, que existem disponíveis diversos modelos de avaliação de qualidade de software, reconhecidos internacionalmente, por entidades oficiais (à exemplo da ISO).

Porém, de todos estes encontrados, pouquíssimos possuíam nas suas etapas a inclusão de prerrogativas que permitissem a avaliação da qualidade do software educacional. O software educacional, conforme citamos na seção 2.6, requer diversos cuidados, que não são observados quando se trata do desenvolvimento de outras taxonomias.

Portanto, ao adicionar às características existentes na norma ISO/IEC 9126 a característica “Aspectos Educacionais”, e levar o processo à aplicação, observamos que se a

avaliação de determinados softwares educacionais, que estão disponibilizados a nível nacional por meio do programa PROINFO do Governo Federal, fosse de fato realizada, conforme citamos na seção 1.1, poderíamos ter softwares de maior qualidade e o que poderia resultar em um melhor aproveitamento dos laboratórios de informática disponibilizados nas escolas.

Considerando que os softwares educacionais após a avaliação contemplariam as características inerentes aos aspectos educacionais e ao contexto a ser aplicado, a aceitação destes seria superior à existente hoje, permitindo assim, que o software educacional, realmente exercesse seu papel educacional, de forma eficiente e natural, propiciando ao aluno uma nova possibilidade de aprender e de fazer conhecimento.

A prova disto, foram as notas obtidas pelo TuxMath, quando aplicado o QUALI-EDU em condições reais, que foi julgado como “Regular”, ou seja, que necessita de verificações, para que possa ser aceito.

É de suma importância que os softwares (não só os educacionais), sigam determinado modelo de avaliação de qualidade, para que, o usuário final, tenha a segurança de que o produto o qual o mesmo está adquirindo, possui qualidade suficiente, para suprir as necessidades do mesmo.

5.1 TRABALHOS FUTUROS

Como trabalhos futuros, podemos citar principalmente a criação de métricas para completar o que foi proposto na seção 3.3.2 deste trabalho, permitindo assim, controlar melhor a avaliação, a partir do momento em que existe um padrão oficial à ser seguido e almejado. . Um outro trabalho futuro direto que pode ser realizado é a avaliação do processo QUALI-EDU, no sentido de averiguar se este atende critérios de completude e corretude na avaliação de softwares educacionais.

Também para aprimorar o processo, incluir nas pesquisas a norma ISO 25010, mais conhecida como SQuaRE, que trata diretamente dos requisitos de qualidade e avaliação de softwares. Vale ressaltar que a norma ISO 25010, é a sucessora da norma ISO/IEC 9126.

A partir disto, busca-se concluir o processo e aplicar em outros níveis de software educacional, com complexidade superior a aplicada neste trabalho.

REFERÊNCIAS

- [1] BENNET, S.; MATON, K.; KERVIN, L. **The ‘digital natives’ debate: A critical review of the evidence.** *British Journal of Educational Technology* 39 (5): 775-786, 2008.
- [2] LOU, Y., ABRAMI, P.C. and D’ APOLLONIA, S. **Small group and individual learning with technology: a meta-analisy,** *Review of Educacional Research*, 2001.
- [3] **Painel MEC** – <<http://painel.mec.gov.br>> Acessado em: 03/02/2014
- [4] COUTINHO, **Sociedade da informação, do conhecimento e da aprendizagem: desafios para educação no século XXI.** In: *Revista de Educação*, Vol XVIII, nº 1, p. 5-22, , 2011.
- [5] SCARDAMALIA, M., & BEREITER, C. **Engaging Students in a Knowledge Society.** *Educational Leadership*, 54(3), 6–10, 1996
- [6] CLARK, R., **Review of Educational Research**, Vol. 53, No. 4., p. 445-459, Winter, 1983.
- [7] VERGARA, S. C, **Métodos de pesquisa em administração.** São Paulo: Atlas, 2005.
- [8] BOGDAN, R., BIKLEN, S. - **Características da investigação qualitativa. In: Investigação qualitativa em educação: uma introdução à teoria e aos métodos.** Porto, Porto Editora, p.47-51, 1994.
- [9] TRIVIÑOS, A., **Introdução à pesquisa em ciências sociais: a pesquisa qualitativa em educação:** 1.ed. São Paulo: Atlas, 1987.
- [10] GIL, A., **Métodos e técnicas de pesquisa social.** São Paulo: Atlas, 1995.
- [11] DIEHL, A., **Pesquisa em ciências sociais aplicadas: métodos e técnicas.** São Paulo: Prentice Hall, 2004.
- [12] ABNT. NBR ISO 8402. **Gestão da qualidade e garantia da qualidade – terminologia.** 1994
- [12] CROSBY, P. , **Quality Is Free,** McGraw-Hill, 1979.
- [13] HUMPHREY, W., **Managing the Software Process,** Addison-Wesley, 1989.
- [14] ISO, **ISO 9000:2005 Quality Management Systems - Fundamentals and Vocabulary,** ed: ISO, 2005.
- [15] FANG, H., **Modeling and Analysis for Educational Software Quality Hierarchy Triangle,** 2008.

- [16] SWEBOK, **Guide to the Software Engineering Body of Knowledge. 2004 Version. A project of the IEEE Computer Society Professional Practices Committee.** Disponível em: <<http://www.swebok.org>> Acessado em: 02/02/2014
- [17] BOTT, F. et al., **Professional Issues in Software Engineering**, 3rd ed. New York: Taylor & Francis, 2000.
- [18] KAN, S. H., **Metrics and Models in Software Quality Engineering**, 2nd ed. Boston: Addison-Wesley, 2002.
- [19] KELLY, D., **Innovative Standards for Innovative Software**. Computer. July, pp. 88-89. 2008.
- [20] ISO, **ISO/IEC 9126 Software Engineering – Product Quality**, 2002.
- [21] CENPRA, **Centro de Pesquisas Renato Archer e DQS – Divisão de Qualidade de Software**. “Descrição do MEDE-PROS”, 2004.
- [22] PERELLI, H., **Avaliação de Software no LAPS – Recife**, 2004.
- [23] NIELSEN, J., **Usability Engineering**. Academic Press. 1994
- [24] WEYUKER, E. J. **Axiomatizing Software Test Data Engineering**, IEEE Transactions on Software Engineering, NY, v.16, n.2, 1986.
- [25] **Software Quality Attributes** - <<http://www.sqa.net/softwarequalityattributes.html>> - Acessado em: 20/01/2014.
- [26] NETO, J., **Um processo para avaliação de produto de software através de análise por especialista**, Recife, 2005.
- [27] LITCHFIELD, A., **Interface communication management: A user centred multimedia design model**. In C. McBeath and R. Atkinson (Eds), Proceedings of the Second International Interactive Multimedia Symposium, 298-303. Perth, Western Australia, 1994. Promaco Conventions. <<http://www.aset.org.au/confs/iims/1994/km/litchfield.html>> Acessado em: 02/02/2014.
- [28] **Definição do Diagrama Pert** <<http://searchsoftwarequality.techtarget.com/definition/PERT-chart>> Acessado em: 02/02/2014.
- [29] **Definição do SCORM** – <<http://pt.wikipedia.org/wiki/SCORM>> - Acessado em: 20/02/2014.
- [30] PUNTER, T., SOLINGER, R., e TRIENEKENS, J., **Software Product Evaluation – Current status and future needs for costumers and industry**, 1997.

- [31] DO NASCIMENTO, T., **Avaliação da Qualidade de um Produto de Software**, Recife, 2010.
- [32] ISO/IEC 14598, **Information Technology – Evaluation of Software Products**, 1998.
- [33] ISO/IEC 14598-1, **International Standard. Information Technology – Software Product Evaluation – Part 1 – General Overview**, 1999.
- [34] ISO/IEC 14598-2, **International Standard. Information Technology – Software Product Evaluation – Part 2 – Planning and Management**, 1999.
- [35] ISO/IEC 14598-3, **International Standard. Information Technology – Software Product Evaluation – Part 3 – Process for developers**, 1999.
- [36] ISO/IEC 14598-4, **International Standard. Information Technology – Software Product Evaluation – Part 4 – Process for Acquirers**, 1999.
- [37] ISO/IEC 12119, **International Standard. Information Technology – Software Packages – Quality requirements and testing**, 1998.
- [38] ISO/IEC 14598-5, **International Standard. Information Technology – Software Product Evaluation – Part 5 – Processo for Evaluators**, 1999.
- [39] ISO/IEC 14598-6, **International Standard. Information Technology – Software Product Evaluation – Part 6 – Evaluation Modules**, 1999.
- [40] TEIXEIRA, J., **Uma Discussão sobre a Classificação de Software Educacional**. <www.revista.unicamp.br/infotec/artigos/jacqueline.html> – Acessado em: 20/02/2014.
- [41] GALVIS, A. H. **Ambientes de Enseñanza-Aprendizaje Enriquecidos con Computador**. Boletín de Informática Educativa. Bogotá, 117-138, Dezembro de 1988.
- [42] REGGINI, H.C. **El pasajero de la góndola: reflexiones en torno a la educación y a LOGO**. Boletín de Informática Educativa, 3 (1): 9-17. Bogotá, abr. 1990.
- [43] TEIXEIRA, A. e BRANDÃO, E., **Software educacional: o difícil começo**; CINTED-UFRGS; v.1 nº1., <http://www.cinted.ufrgs.br/renote/fev2003/artigos/adriano_software.pdf>, publicado em fevereiro de 2003, acessado em 18/02/2014
- [44] ROCHA, A. e CAMPOS, G., **Avaliação da Qualidade de Software Educacional**. Brasília, v. 12, n. 57, p. 32-44, 1993.
- [45] GIRAFFA, L., **Uma odisséia no ciberespaço: O software educacional dos tutoriais aos mundos**. Revista Brasileira de Informática na Educação, Sociedade Brasileira de Computação, vol. 17, n.1, p. 20-30, jan/jul 2009.

- [46] LIMA, M. - **PREÁ- Um processo de Desenvolvimento de Software Educacional Baseado em Metodologias Ágeis**, Campina Grande, 2013.
- [47] VIEIRA, F., **Avaliação de Software Educativo: Reflexões para uma - Análise Criteriosa**. Disponível em <<http://www.edutecnet.com.br>>, acessado em: 18/02/2014.
- [48] GIRAFFA, L., **A Comunidade de Informática Educativa Brasileira: Perspectiva Histórica e Pesquisa**. Painel Integrado de Palestras e Debates. In: VI WORKSHOP DE INFORMÁTICA NA ESCOLA, Curitiba, Brasil, 2000.
- [49] GLADCHEFF, A., SANCHES, R, DA SILVA, D. **Um instrumento de avaliação de qualidade de software educacional: como elaborá-lo**, Rio de Janeiro, VIII Workshop de Qualidade de Software, 2002.
- [50] HINOSTROZA, J., MELLAR, H., **Pedagogy embedded in educational software design: report of a case study**, *Comp. & Education*, 37: 27–40. 2001.
- [51] SHNEIDERMAN, B., **Designing the user interface: strategies for effective human-computer-interaction**, Addison. 1998.
- [52] ALQUIMIM, H., **Proposta de critério para avaliação de Softwares de questionários eletrônicos, com base nas normas ABNT NBR ISO/IEC 9126 e 14598**, PUC - São Paulo-SP, 2013.
- [53] MARÇAL, E. ; BEUREN, I.M. **Auditoria da qualidade de um software de contabilidade**. *Gestão & Regionalidade*, v. 23, Fonte: <http://seer.uscs.edu.br/index.php/revista_gestao/article/download/68/31>, 2009.
- [54] **Agile. Manifesto for Agile Software Development**, <<http://www.agilemanifesto.org>>, Acessado: em 20/02/2014.
- [55] PRESSMAN, R., **Engenharia de Software**. 6ª ed, McGraw-Hill, 2006.
- [56] MORIN, E., **Os sete saberes necessários à educação do futuro**. São Paulo, Cortez, 2002.
- [57] TAPIA, J. e FITA, H.. **A motivação em sala de aula: o que é, como se faz**. Trad. Sandra Garcia. São Paulo: Loyola. 2000.
- [58] CORTES, M e CHIOSSI, T. **Modelos de Qualidade de Software**. Campinas, SP, Editora da UNICAMP, IC, 2001.
- [59] **TuxMath** <<http://pt.wikipedia.org/wiki/TuxMath>> Acessado em: 25/02/2014

ANEXOS

Anexo 1

Relação de Programas disponibilizados no Linux Educacional 3.0

- Ferramentas de Produtividade
 - BrOffice.org 2.4.1
 - Apresentação Eletrônica (Impress)
 - Banco de Dados (Base)
 - Desenho Vetorial (Draw)
 - Planilha Eletrônica (Calc)
 - Processador de Texto (Writer)
 - Visualizador de Arquivos PDF (KPDF)
- Gráficos
 - Gerenciador de Fotos (digikam)
 - Programa de Captura de Tela (Ksnapshot)
 - Programa de Digitalização & OCR (Kooka)
 - Programa de Pintura (Kolourpaint)
- Internet
 - Navegador Web (Firefox 3.0)
- Multimídia
 - Editor de Audio(Audacity)
 - Editor de Vídeo (Kdenlive)
 - Gravação de CD/DVD (K3b)
 - Reprodutor Multimídia (VLC)
- Programas Educacionais
 - Linguagem Logo (Kturtle)
 - Tabela periódica do elementos (kalzium)

- Planetário Virtual (Kstars)
- Treinamento em Geografia (Kgeography)
- Aprender Alfabeto (Klettres)
- Estudo das Formas Verbais do Espanhol (Kverbos)
- Ferramenta de referência/estudo do japonês (Kiten)
- Jogo de Forca (KhangMan)
- Jogo de ordenação de letras (Kanagram)
- Revisor de latim (Klatin)
- Desenho de funções matemáticas (kmpplot)
- Exercício com frações (Kbruch)
- Exercícios de porcentagens (Kpercentage)
- Geometria Interativa (Klg)
- Desenho (Tux paint)
- Editor de Testes e exames (Keduca)
- Jogo Simon Diz (blinKen)
- Treinador de vocabulário (KwordQuiz)
- Treinador de vocabulário (KvocTrain)
- Tutor de Digitação (Ktouch)
- Tux of Math Command (TuxMath)
- Ferramentas de Busca
 - Barra Superior (EduBar)
 - Ferramentas de Acesso a Conteúdos (FBEDu)

Anexo 2

Documento da fase de Motivação da Avaliação

QUALI-EDU

DOCUMENTO DA MOTIVAÇÃO DA AVALIAÇÃO DO SOFTWARE EDUCACIONAL

NOME DO SOFTWARE: _____

STATUS DO SOFTWARE:

CONCLUÍDO EM FASE DE DESENVOLVIMENTO (PROTÓTIPO)

FUNÇÕES PRINCIPAIS DO SOFTWARE: _____

PROPÓSITO(S) DA AVALIAÇÃO: _____

PARTICIPANTES DO 1º ENCONTRO: _____

Local, <dd> de <mm> de <aaaa>.

_____ <Assinatura do avaliador> _____

Nome do Avaliador

Anexo 3
Documento do Julgamento Final do Software

QUALI-EDU

DOCUMENTO DE JULGAMENTO FINAL DA AVALIAÇÃO DO SOFTWARE EDUCACIONAL

NOME DO SOFTWARE: _____

STATUS DO SOFTWARE:

CONCLUÍDO **EM FASE DE DESENVOLVIMENTO (PROTÓTIPO)**

NOTA FINAL OBTIDA PELO SOFTWARE: _____

JULGAMENTO FINAL DO SOFTWARE: _____

CONSIDERAÇÕES DO AVALIADOR: _____

Local, <dd> de <mm> de <aaaa>.

_____ <Assinatura do avaliador> _____

Nome do Avaliador