



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS CAMPINA GRANDE
CENTRO CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO**

ALANNA CAMYLLA COÊLHO MONTEIRO

Requisitos Não-Funcionais para o Desenvolvimento de Software Educacional

CAMPINA GRANDE – PB
2014

ALANNA CAMYLLA COELHO MONTEIRO

Requisitos Não-Funcionais para o Desenvolvimento de Software Educacional

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

Orientadora: Prof^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.

M775r Monteiro, Alanna Camylla Coêlho.

Requisitos não funcionais para o desenvolvimento de software educacional [manuscrito] / Alanna Camylla Coelho Monteiro. - 2014. 40 p. : il. color.

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. Engenharia de Requisitos. 3. Requisitos não-funcionais. I. Título.

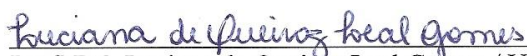
21. ed. CDD 371.33

ALANNA CAMYLLA COELHO MONTEIRO

Requisitos Não-Funcionais para o Desenvolvimento de Software Educativo

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

Aprovada em 14, de março, de 2014



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

Orientadora



Prof.^a Ma. Ana Isabella Muniz Leite/ UEPB

Examinadora



Prof.^a Me. Edson Holanda Cavalcanti Junior/UEPB

Examinador

DEDICATÓRIA

Aos meus pais, Hamilton e Laura, por tudo o que eles são e representam na minha vida, tudo o que sou é fruto dos ensinamentos que eles me passaram.

AGRADECIMENTOS

Agradeço a Deus pela fidelidade para comigo, tudo o que tenho e sou é teu Senhor. Muito obrigada por fazer parte da minha vida. A Nossa Senhora pelo seu manto protetor e interceder a Deus tudo o que preciso.

À minha família, em especial meus pais e minha irmã, por todo amor, broncas, renúncias e por todo apoio indispensável, essa graduação é tão minha quanto de vocês.

À professora Luciana Leal pelo apoio, dedicação, paciência e por todos os ensinamentos passados. Aos colegas Maelson Marques e Jefferson Felipe pelas experiências trocadas durante todo o projeto.

À turma 2010.1 pela oportunidade de ter convivido com vocês, sabemos que sozinhos não somos nada e que juntos podemos fazer mais. Aos trancos e barrancos sobrevivemos.

À todo corpo docente do curso de Licenciatura em Computação pelos ensinamentos que não só contribuíram para minha vida acadêmica, mas também para a minha vida pessoal.

RESUMO

A utilização cada vez mais crescente de ambientes virtuais de aprendizagem faz com que software educacionais procurem ter uma maior qualidade, e que este cumpra com o papel que foi destinado. Para tanto, é de fundamental importância que este tipo de software seja bem desenvolvido, para isso é necessário que tenhamos requisitos bem definidos. Como este software tem uma característica peculiar que é passar conhecimento, deve-se preocupar com o lado pedagógico do mesmo. Dessa forma, levantamento de requisitos funcionais e não funcionais é importante para o bom desempenho do sistema. Este trabalho tem como objetivo obter e avaliar requisitos não funcionais que sejam pertinentes para softwares de caráter educativo. Para tal, nesta pesquisa foi empregado um método de pesquisa qualitativa denominado Teoria Fundamentada nos Dados, ou Grounded Theory. A partir de entrevistas realizadas com estudantes, um grupo de requisitos não funcionais foi obtido e analisado, de acordo com as codificações utilizadas para se aplicar a Teoria Fundamentada nos Dados. Posteriormente, após a obtenção desses requisitos não funcionais, um questionário de avaliação destes requisitos não funcionais foi aplicado a docentes para uma avaliação dos mesmos.

PALAVRAS-CHAVE: Software Educacional, Engenharia de Requisitos, Requisitos não-funcionais, Teoria Fundamentada nos Dados.

A B S T R A C T

The ever increasing use of virtual learning environments makes educational software seek to have a higher quality , and it complies with the paper that was intended . Therefore, it is of fundamental importance that this type of software is well developed , so it is necessary to have well-defined requirements . As this software has a unique feature that is passing knowledge , one must worry about the educational side of it. Thus , survey of the functional and nonfunctional requirements is important for good system performance . This paper aims to obtain and evaluate non-functional requirements relevant to educative software character . To this end , this research has employed a qualitative research method called Grounded Theory , and Grounded Theory . From interviews with students , a group of non-functional requirements was obtained and analyzed according to the encodings used to apply to Grounded Theory . Later , after getting those non-functional requirements , an evaluation questionnaire of these non-functional requirements has been applied to teachers for evaluation.

KEYWORDS: Educational Software, Requirements Engineering, Non-Functional Requirements, Grounded Theory.

LISTA DE TABELAS

TABELA 1 – Requisitos não funcionais obtidos a partir da Codificação Aberta da primeira rodada de entrevistas.	27
TABELA 2 – Grupos de requisitos da codificação axial da primeira rodada	29
TABELA 3 – Grupos de requisitos da codificação axial da segunda rodada.....	29
TABELA 4 – Grupos de requisitos da codificação axial da terceira rodada	30
TABELA 5 – Resultado Predominante na Avaliação dos Docentes	31

LISTA DE FIGURAS

FIGURA 3.1 – Sequência de passos para a execução da Teoria Fundamentada nos Dados.....24

SUMÁRIO

1. INTRODUÇÃO.....	12
1.1 CONTEXTO E MOTIVAÇÃO.....	13
1.2 OBJETIVOS.....	14
1.2.1 OBJETIVO GERAL.....	14
1.2.2 OBJETIVOS ESPECÍFICOS.....	14
1.3 METODOLOGIA DE PEQUISA E DESENHO DE PESQUISA.....	15
1.4 A ESTRUTURA DO TRABALHO.....	15
2. A ENGENHARIA DE REQUISITOS E O DESENVOLVIMENTO DE SOFTWARE EDUCACIONAIS.....	17
2.1 REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS.....	20
2.2 O LEVANTAMENTO DE REQUISITOS NÃO-FUNCIONAIS E O DESENVOLVIMENTO DE SOFTWARE EDUCACIONAL.....	21
3. TEORIA FUNDAMENTADA NOS DADOS E OBTENÇÃO DOS REQUISITOS NÃO-FUNCIONAIS.....	23
3.1 APLICAÇÃO DO MÉTODO.....	25
3.2 RESULTADOS OBTIDOS.....	26
4. AVALIAÇÃO DOS REQUISITOS PROPOSTOS.....	30
5. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....	32
REFERÊNCIAS.....	33
ANEXOS.....	36
APÊNDICE.....	37

1. INTRODUÇÃO

O desenvolvimento de um software é iniciado com o reconhecimento dos requisitos que este deverá possuir. Para isso, existe a engenharia de requisitos que aborda as formas existentes de como coletar este tipo de informação a fim de que não haja erros futuros no sistema.

A Engenharia de Requisitos é o processo pelo qual os requisitos de um produto de software são coletados, analisados, documentados e gerenciados ao longo de todo o ciclo de vida deste software (AURUM; WOHLIN, 2005). A compreensão desse processo facilita a inferência da equipe que irá desenvolver o software em relação ao que será utilizado/necessário.

A utilização da Engenharia de Requisitos é de fundamental importância para os softwares que possuem o fator da aprendizagem. Esta necessita de um processo bem definido para o desenvolvimento de software educacional. Por ser um tipo de software que requer uma análise e uma equipe diferenciada, sua elaboração torna-se mais complexa pelos diferentes papéis que farão parte da concepção do software. Observando a complexidade exigida em um software educacional, recomenda-se a formação de uma equipe multidisciplinar para a construção do mesmo, pois neste, a fase de levantamento de requisitos se mostra ainda mais complexa quando comparada aos softwares convencionais, por envolver *stakeholders* de diferentes áreas de conhecimento (LACERDA, 2007).

Gomes, Tedesco e Castro-Filho (2003), observam uma questão importante envolvendo o desenvolvimento de aplicações educativas que está na especificidade dos requisitos não-funcionais de usuários, que, muitas vezes, parecem pouco integralizados neste desenvolvimento.

Essa pesquisa foi realizada para identificar quais as necessidades dos usuários de softwares educacionais com relação aos requisitos não funcionais que eles devem possuir. O público alvo desta pesquisa são desenvolvedores de software, pedagogos e psicólogos.

Para melhor gerenciar os requisitos de um projeto, é importante que uma ferramenta de apoio à Engenharia de Requisitos, esteja integrada às demais ferramentas utilizadas no desenvolvimento de software (MARTINS, 2007). Logo, para a obtenção dos requisitos não-funcionais de software educacional desta pesquisa foi utilizada a Teoria Fundamentada nos Dados, que possui caráter qualitativo.

1.1 CONTEXTO E MOTIVAÇÃO

O software educacional na sala de aula, como fonte pedagógica para auxiliar a explanação/exemplificação de conteúdos, torna as aulas mais agradáveis e a utilização de recursos tecnológicos prende a atenção do aluno quanto ao assunto abordado. Segundo Valente (1993), há uma mudança profunda no paradigma pedagógico devido à possibilidade de diversificação dos métodos de ensino proporcionados pelos computadores quando estes são usados para melhorar a didática de ensino.

Já existem muitos softwares que são utilizados nos laboratórios de informática das instituições de ensino (SILVEIRA e BARONE, 1999). A crescente inserção das tecnologias em sala de aula para que haja uma maior motivação por parte dos alunos vem estimulando o desenvolvimento e utilização de softwares educacionais.

Software educacional é definido como sendo um sistema computacional e interativo, intencionalmente concebido para facilitar a aprendizagem de conceitos específicos (GOMES e PADOVANI, 2005). Desta forma, a busca por este tipo de sistema tem se tornado cada vez maior pelas suas facilidades.

A produção de software educacional sem a participação ativa de profissionais especialistas nessas áreas como pedagogos, professores e é claro os próprios alunos que também fazem parte deste ciclo (SCHNEIDER e VASA, 2006) (DAS, 2007). Portanto, para este trabalho o foco maior foi identificar qual é a necessidade dos usuários de software educacional, no caso específico deste trabalho dos alunos, e o que eles procuram em um software que possui esse caráter educativo.

Com o aumento da complexidade dos softwares educacionais em relação aos sistemas produzidos anos atrás, da diversidade de tecnologias adotadas e do número de pessoas envolvidas, tornou-se inviável projetá-lo sem utilizar um processo bem definido para orientar o seu desenvolvimento (GIRAFFA, MARCZAK e PRIKLADNICKI, 2005). Assim, é essencial que se tenha uma análise de requisitos eficiente e não ocorra grandes mudanças, análise de requisitos para que não haja grandes mudanças nas etapas subsequentes.

Este trabalho está especificado em quais requisitos não-funcionais são mais relevantes para software educacional, tendo como base os dados coletados das entrevistas realizadas.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

Este trabalho tem como principal objetivo avaliar quais requisitos não-funcionais são necessários para software educacional. Já que este é um tipo de software de caráter específico e voltado para o processo da aprendizagem. Essa avaliação foi feita a partir da teoria fundamentada nos dados.

1.2.2 OBJETIVOS ESPECÍFICOS

- Verificar a prevalência dos requisitos não-funcionais para estudantes a partir de uma entrevista.

- Comprovar quais dos requisitos são relevantes para software educacional com uma avaliação por parte de professores.

1.3 METODOLOGIA DE PESQUISA E DESENHO DE PESQUISA

Esta pesquisa realizou um estudo qualitativo, pois procurou-se entender os fenômenos da Engenharia de Requisitos na perspectiva das inquietações dos entrevistados a respeito sobre os softwares educacionais. Esta pesquisa foi desenvolvida no projeto de pesquisa em Engenharia de Software da UEPB, o Propesq, este era parte do projeto que era intitulado como Uma Revisão Sistemática das Técnicas e Ferramentas usadas em Desenvolvimento de Software Educativo. Realizado no período de agosto de 2012, até fevereiro de 2014. Foram analisadas entrevistas realizadas com estudantes do curso de Licenciatura em Computação, do Campus I da UEPB, e alguns estudantes de diversas áreas. Mais sobre a metodologia de pesquisa encontra-se no Capítulo 3.

A primeira etapa desse trabalho foi o estudo da bibliografia e o embasamento teórico. A segunda etapa foi a elaboração das entrevistas para compor a aplicação da Teoria Fundamentada nos Dados. A terceira etapa foi a aplicação da entrevista com os estudantes, e a última etapa se deu a partir da avaliação de professores universitários sobre a importância dos requisitos não funcionais obtidos para o desenvolvimento de software educacional.

1.4 ESTRUTURA DO TRABALHO

A estrutura deste trabalho de conclusão de curso é apresentada a seguir. Este capítulo introdutório traz a contextualização e motivação para o trabalho realizado, além de apresentar os objetivos pretendidos, a pergunta de pesquisa e o desenho utilizado na realização da mesma. No Capítulo 2, se encontra a fundamentação teórica, onde se tem um breve levantamento bibliográfico com o conceito da engenharia de requisitos; os requisitos

funcionais e não-funcionais, e a relação entre eles; e os requisitos não-funcionais em relação aos software educacionais que é o foco desta pesquisa.

No Capítulo 3, apresenta-se a teoria fundamentada nos dados, conceituando-a, e a obtenção de requisitos não-funcionais a partir da teoria. O Capítulo 4 apresenta a avaliação dos requisitos que foi realizada mediante pesquisa através de um questionário para docentes de diversas instituições de ensino superior. Finalmente no Capítulo 5, as considerações finais e sobre o trabalho são apresentadas e são sugeridos trabalhos futuros.

2. A ENGENHARIA DE REQUISITOS E O DESENVOLVIMENTO DE SOFTWARE EDUCACIONAIS

A Engenharia de Requisitos é uma das fases mais complexas na Engenharia de Software, pois ela trata da compreensão das necessidades do cliente. Para Kotonya e Sommerville - (1998), definem requisitos como a descrição de uma funcionalidade no nível do usuário, da propriedade geral do sistema, da restrição específica sobre o sistema, da especificação de um algoritmo particular que deva ser empregado em cálculos particulares e em uma restrição aplicável ao processo de desenvolvimento do sistema.

A definição dos requisitos não é um processo matemático e há fatores organizacionais, técnicos e sociais envolvidos (BATISTA, 2003). Neste ponto, a Engenharia de Requisitos se enquadra, no intuito de apontar as características que o sistema deverá possuir.

A Engenharia de Requisitos é um processo que envolve todas as atividades exigidas para criar e manter o documento de requisitos de sistema (SOMMERVILLE, 2004). Essas atividades são divididas em algumas etapas que podem adequar-se ao tipo de software que será desenvolvido. É oportuna a identificação dos requisitos para que haja a implementação do sistema.

A Engenharia de Requisitos fornece o mecanismo apropriado para entender o que o cliente deseja, analisando as necessidades, avaliando a exequibilidade, negociando uma condição razoável, especificando a solução de modo não ambíguo, validando a especificação e gerindo os requisitos à medida que eles são transformados em um sistema operacional (THAYER, 1997).

Um escopo do projeto com etapas bem definidas e detalhadas facilita a busca de solucionar todos os questionamentos que poderão ocorrer. Estas etapas são apresentadas a seguir:

Concepção é a compreensão do problema, da forma mais básica e objetiva possível. Um questionário com perguntas diretas e simples tornam o resultado dessa fase mais palpável. Para a criação de softwares educacionais essa fase é de extrema importância, com a verbalização do problema tornando mais fácil o caminho para se chegar à solução.

Levantamento é o papel de descoberta dos requisitos que o cliente/usuário deseja. Segundo Sommerville (2004), nesta etapa os membros da equipe técnica de desenvolvimento de software trabalham com o cliente e os usuários finais do sistema para descobrir mais informações sobre o domínio da aplicação, que serviços o sistema deve fornecer, o desempenho exigido do sistema, as restrições de hardware.

Elaboração é a fase onde ocorre a modelagem do projeto, um exemplo prático é a criação de uma UML. Nesta fase, as interações do sistema se tornarão mais visíveis para o cliente, tornando os papéis mais definidos, onde os professores, psicólogos e alunos farão parte do sistema, fazendo parte do projeto os aspectos metodológicos e pedagógicos.

A **negociação** é feita entre cliente e usuário. Ambos têm perspectivas e ideias convergentes de como deverá ficar o software. Para a educação é necessário que o usuário seja um professor, devido este conhecer as dificuldades e prioridades dos seus alunos. Para Vasconcelos et. al (2006), a qualidade da negociação depende de um bom entendimento e de uma análise rigorosa.

Especificação é uma forma de mapear como deverá estar nas fases posteriores. Essa especificação é variada de acordo com a dimensão e a finalidade do projeto. As tarefas associadas com a especificação de requisitos existem para prover uma representação do software que possa ser revisada e aprovada pelo usuário (VASCONCELOS et al, 2006). Para softwares educacionais é mais conveniente a criação de um protótipo, porque ainda se está na

fase de testes e planejamentos, e terá uma noção mais real de como está caminhando o projeto.

Validar a especificação significa fazer uma análise direta para que não haja no produto final os defeitos, ambiguidades e o que não tem necessidade de ser projetado ao final do software. Segundo Falbo (2005), o objetivo da validação é assegurar que o software que está sendo desenvolvido é o software correto, ou seja, se os requisitos e o software dele derivado atendem ao uso específico proposto.

Sem dúvidas, a tarefa mais difícil em um projeto de software é a responsável pelo levantamento e análise de requisitos, uma vez que é a partir dos resultados dela, que todo o software será desenvolvido. Segundo Pfleeger (2004), mais de 30% dos problemas na construção dos softwares estão relacionados a problemas com requisitos, seja no seu levantamento, nas mudanças das especificações ou na falta de envolvimento do usuário.

Essa atividade torna-se ainda mais complicada quando se trata em softwares voltados para a área educacional. Um software educacional pode ser usado em diversos tipos de contexto, desde lugares distintos, a usuários de diversas idades e conhecimentos. Estes podem encontrar-se lado a lado ou estarem distantes uns dos outros interagindo através de ambientes virtuais (BRICKER, 1993).

As indecisões do cliente com relação ao que ele quer ver e ter no seu software tornam ainda mais difícil o levantamento dos requisitos. O que não quer dizer, que um cliente bem decidido não poderá fazer mudanças futuras.

Um dos principais pontos a ser destacado no desenvolvimento de software, preconizado pelos métodos ágeis de desenvolvimento (AGILE, 2011), é o trabalho conjunto do cliente com a equipe de desenvolvimento, para que ambos evoluam o seu entendimento de como solucionar o problema.

Um dos motivos da dificuldade em produzir software educacional de qualidade parece estar ligado ao fato que no processo de concepção há uma diferença significativa entre as representações que designers, programadores e professores têm acerca dos processos de ensino e aprendizagem (MANDEL, 1997). Coletar requisitos para a elaboração de software educacional é uma tarefa complexa, por ter uma equipe diferenciada que faça parte dessa etapa. Aluno, professores, pedagogos e psicólogos fazem parte desta etapa, além do cliente e do usuário.

2.1 REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS

Requisitos funcionais dizem respeito às funcionalidades que o software deverá possuir e como este software deve responder a determinadas situações. Todo sistema possui características peculiares, logo é preciso que os requisitos para o mesmo sejam bem definidos.

Alguns atributos são definidos para os requisitos funcionais, estes são a completude e consistência do sistema. Completude diz respeito à definição de todos os serviços que o sistema irá oferecer. A consistência significa que os requisitos não podem ter ambiguidade para que não se tenha definições contraditórias.

Os requisitos não funcionais definem qualidades gerais do sistema, e por estarem diretamente relacionados com a satisfação do usuário são determinantes de sucesso para o sistema. Eles são subjetivos, podem ter diferentes interpretações para cada pessoa; relativos, dependem diretamente do sistema a ser implementado; e são interativos, interagem entre si, podendo afetar negativamente ou positivamente outros requisitos (LEITE & SANTA CRUZ, 2009). Requisitos não funcionais definem o sistema, suas restrições e propriedades. Existem diversos tipos de requisitos não-funcionais que dependem da necessidade do cliente/usuário tendo em vista as características qualitativas do software.

A complexidade de um software é determinada em parte por sua funcionalidade, ou seja, o que o sistema faz, e em parte por requisitos gerais que fazem parte do desenvolvimento

do software como custo, performance, confiabilidade, manutenibilidade, portabilidade, custos operacionais entre outros (CHUNG, 1999). Esses são alguns tipos de requisitos não-funcionais existentes.

2.2 O LEVANTAMENTO DE REQUISITOS NÃO-FUNCIONAIS E O DESENVOLVIMENTO DE SOFTWARE EDUCACIONAL

As técnicas de levantamento de requisitos permitem identificar uma ou mais maneiras que possibilitem a construção de um modelo que será utilizado para desenvolvimento do software. Com as diversas técnicas existentes escolher a que melhor se adequa ao desenvolvimento de um determinado software pode ser uma tarefa difícil.

Os desenvolvedores de software têm procurado a melhor forma de levantar os requisitos de um sistema e, dessa forma, diversas técnicas e métodos foram criados para garantir que um sistema atenda às necessidades e expectativas dos clientes (BATISTA, 2003), (PRESSMAN, 2002), (NETO, GOMES E TEDESCO, 2003), (BORTOLI E PRICE, 2000).

Segundo Silva (2005), existem muitas técnicas de modelagem de requisitos, sendo as seguintes as mais expressivas:

- Casos de Uso: conjunto de seqüências de ações, resultantes da interação do sistema com um ator. (BOOCH, RUMBAUGH E JACOBSON, 2000);
- Viewpoints: considera aspectos do sistema percebidos por diferentes requerentes. (FINKELSTEIN, et al. 1992);
- Expressão Controlada de Requisitos – CORE: as perspectivas podem ser físicas ou funcionais, representando organizações, homens, entidades de hardware ou software e processos, no final combinadas representam as atividades. (FINKELSTEIN, et al. 1992);

- Diagrama de Fluxo de Dados: é uma representação gráfica que mostra o fluxo de informação e as transformações que são aplicadas à medida que os dados se movem da entrada para a saída (PRESSMAN, 2002);
- Processador de Linguagem de Requisitos – RLP: desenvolvimento de linguagens formais para os requisitos de especificação das aplicações (SILVA, BONIN E PALUDO, 2005);
- Metodologia da Engenharia de Requisitos de Software -- SREM: os caminhos dos dados consistem nas mensagens de entrada, na sequência das tarefas processadas que envolvem o fluxo do controle, e nas mensagens de saída. (FINKELSTEIN et al., 1992);
- Volere: é um método completo de obtenção de requisitos, baseado nos casos de uso. (FISCHER, 2001).

Tratando-se de software educacional outras técnicas foram adaptadas como o mapa mental, ou desenvolvidas, como a árvores de características. A **árvore de características** estimula o detalhamento dos requisitos, uma vez que ao se abrir um ramo da árvore, os modeladores se sentem impulsionados a criar folhas, aumentando a árvore e, por fim, criando uma análise mais fina do sistema (OLIVEIRA et al, 2010). A partir do crescimento da árvore fica visível o que o cliente deseja em cada parte do projeto. Segundo Vergnaud (1997), o **mapa mental** permite descrever a organização das ações e identificar, por meio de inferências, os elementos e a dinâmica interna de tais esquemas mentais. Mapa mental é adequado à modelagem da aprendizagem que ocorre no uso de uma interface.

Devido às semelhanças entre mapa mental e árvore de características estes podem ser confundidos. Mas o mapa mental é mais superficial em relação a levantamento de requisitos, já a árvore possui detalhes que podem ser levados em conta durante todo o processo de desenvolvimento.

A árvore de característica é uma expressiva técnica para o levantamento dos requisitos não-funcionais, pois trata de um modelo gráfico de como o software ficará no final. Com a maior quantidade de detalhes de cada requisito que o sistema irá possuir facilita o entendimento por parte de toda a equipe. Para Batista (2003), os requisitos do sistema são organizados de forma hierárquica, como também são detalhados, o que pode facilitar a execução das etapas da engenharia de requisitos.

Como mostrado por Silva (2005), casos de uso, diagrama de fluxo de dados, volere e expressão controlada de requisitos – (CORE) são fontes de coleta de requisitos tanto funcionais quanto não-funcionais. Devido à dificuldade de obtenção dos requisitos não funcionais existem poucas técnicas que façam esse tipo de análise de requisitos. Assim, esta pesquisa buscou uma maneira identificar alguns requisitos não-funcionais, a fim de que o software educacional obtido também a partir destes possa refletir as reais necessidades dos seus usuários.

3. TEORIA FUNDAMENTADA NOS DADOS E OBTENÇÃO DE REQUISITOS NÃO-FUNCIONAIS

A Teoria Fundamentada nos Dados é um método de pesquisa qualitativo que utiliza um conjunto de procedimentos sistemáticos de coleta e análise dos dados para gerar, elaborar e validar teorias substantivas sobre fenômenos essencialmente sociais, ou processos sociais abrangentes (BANDEIRA e CUNHA, 2003). Método de pesquisa proposto por Glaser e Strauss (1974). Eles propuseram dois tipos de teorias: as formais e as substanciais. A Teoria Fundamentada nos Dados, como método substancial, propõe que a teoria surja da coleta e análise de dados.

Para que se chegue a uma teoria, ou a conclusões em relação ao fenômeno estudado, é preciso que se chegue a saturação de dados, obtida a partir da análise das respostas quando não existe mais a diversidade dos códigos obtidos, quando não surgem novos códigos a partir

de novas rodadas de entrevistas.. Um conceito (ou código) dá nome a um fenômeno de interesse para o pesquisador; abstrai um evento, objeto, ação, ou interação que tem um significado para o pesquisador (STRAUSS, 1998). O método é finalizado com o refinamento dos códigos obtendo as respostas que seriam necessárias para o fim da teoria. A Teoria Fundamentada nos Dados propõe três etapas de codificação: codificação aberta, codificação axial e codificação seletiva.

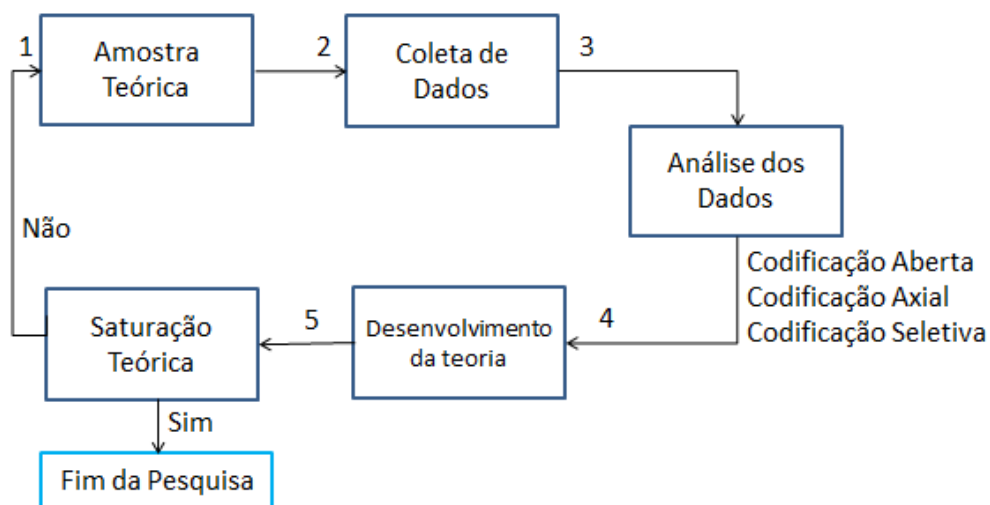


FIGURA 3.1. Sequência de passos para a execução da Teoria Fundamentada nos Dados

É possível notar que o ciclo pode ser executado várias vezes, ou seja, várias etapas de coleta de dados e análise dos dados são realizadas até a concepção da teoria propriamente dita.

Na codificação aberta, tem-se a fase inicial onde os dados são explorados e examinados criteriosamente. Durante a codificação aberta, os dados são separados em partes distintas, examinados e comparados em busca de similaridades e diferenças (STRAUSS e CORBIN, 2008, pg 104). São feitas comparações, análises e a categorização relevante dos dados a partir das leituras dos textos. Nessa fase, o que é relevante para o pesquisador é agrupado formando os códigos.

Na codificação axial é examinada as relações entre as categorias que formam as proposições da teoria substantiva (BANDEIRA e CUNHA, 2006). Segundo Strauss e Corbin (1990), a codificação axial é definida em um conjunto de procedimentos onde os dados são agrupados de novas formas, após a codificação aberta, através das conexões entre as categorias. Isso é feito através do uso de um paradigma de codificação que envolve condições, contexto, estratégias de ação/interação e consequências.

Por fim, a codificação seletiva faz o refinamento de todos os códigos que foram analisados propondo uma teoria central. Esta categoria central pode ser uma categoria existente, ou uma nova categoria pode ser criada. O processo de codificação é finalizado quando nenhum novo dado adiciona novos conhecimentos ao processo de análise da categorização; este momento é denominado “saturação teórica” (BIANCHI et al, 2006).

3.1 APLICAÇÃO DO MÉTODO

Para aplicar o método, foram realizadas entrevistas com o objetivo de identificar as necessidades do cliente ou usuário de um software educacional, ou seja, quais características e restrições são mais importantes para a construção de um software educacional, do ponto de vista do usuário final. O período em que foram realizadas as entrevistas foi de agosto até dezembro de 2013.

A entrevista foi orientada por um conjunto de perguntas elaborado pela pesquisadora e pela orientadora, na qual os entrevistados relataram suas vivências com software educacional. As questões que compõem este roteiro para a entrevista estão no Apêndice.

Foram realizadas três rodadas de entrevistas, para que fosse obtida a saturação dos dados. Ao final de cada rodada, fez-se a codificação aberta e a codificação axial. As duas primeiras rodadas, cada uma com 20 entrevistados, foram feitas com alunos do curso de Licenciatura em Computação do Campus I, da Universidade Estadual da Paraíba. A última

rodada, para consolidar os dados obtidos, foi realizada com 9 entrevistados, estudantes de diversas áreas. Ao todo, nas três rodadas de entrevistas, foram 49 entrevistados.

Para utilizar o método a Teoria Fundamentada nos Dados é necessário examinar cada entrevista, mais de uma vez e por mais de um pesquisador, de maneira independente. Esta atividade é realizada para evitar viés e identificar unidades de análise para a realização da codificação aberta que procura estabelecer categorias conceituais. Ao final da primeira rodada de entrevistas, fez-se a codificação seletiva e elaborou-se uma teoria com resultados presentes em COELHO et. al. (2013).

A aplicação do método tem que ser feita sucessivas vezes em todas as categorias encontradas para a construção efetiva das relações existentes. Nas rodadas posteriores foram feitas também as análises de todas as entrevistas, realizando todas as etapas do método para que chegássemos a saturação dos dados quanto a necessidade dos usuários a respeito de requisitos não-funcionais.

3.2 RESULTADOS OBTIDOS

A Teoria Fundamentada nos Dados foi aplicada a partir da transcrição das entrevistas e análise das mesmas. Dessa forma, cada resposta foi analisada e agrupada de acordo com o requisito não funcional ao qual corresponde. Esta primeira etapa realizada foi a Codificação Aberta.

Na Tabela 1 encontra-se uma exemplificação dos requisitos não-funcionais e as correspondentes unidades de análise. As unidades de análise são categorias conceituais, onde relacionam as respostas com os requisitos não-funcionais adequados.

TABELA 1. Requisitos não funcionais obtidos a partir da Codificação Aberta da primeira rodada de entrevistas

REQUISITOS NÃO-FUNCIONAIS	UNIDADE DE ANÁLISE
1) Usabilidade	“Eu acho que é a usabilidade dele (software), a forma, a interface, a visibilidade de você usar ele, e ele resolver o que propor.”
2) Aprendizagem	“O jogo era um software de multiplicação, eu usei quando paguei a cadeira de prática de ensino, que ele facilitava o aprendizado da multiplicação através do jogo que era um futebol.”
3) Adequação de conteúdo	“Que ele (software) tenha as ferramentas necessárias para resolver meus problemas, independente de interface e de boniteza.”
4) Exemplificação	“Um software que tivesse muitos exemplos, alguma forma que ensinasse várias formas.”
5) Interatividade	“Interativo (componentes de interface), é bom. Porque facilita para entender o que ele (software) faz.”
6) Contexto de uso	“O ruim é que ele (o software) era em inglês então a maioria do pessoal não conseguiu utilizar ele com grande eficiência.”
7) Relação entre a teoria e a prática	“Que o software pudesse auxiliar a gente no sentido de explicar direitinho cada linha de código tava fazendo.”
8) Presença de elementos gráficos	“E é o que eu mais busco em um software educacional ou em qualquer software, é justamente a parte gráfica.”
9) Atratividade	“Pra mim a jogabilidade (do software), a dificuldade, o conteúdo que tem no jogo, a história.”
10) Documentação	“A pessoa tem que aprender como usar o software. Seria interessante um tutorial.”
11) Clareza de conteúdo	“De preferência a compreensão, um software difícil de ser compreendido ele perde a total função primordial que é fazer com que o outro entenda.”
12) Presença de elementos gráficos/Encadeamento de ideias	“Se ele (software) possui muitos componentes de interface, mas tiver bem distribuído como aconteceu na reformulação do Word 2007 com o Word 2003, então é uma vantagem”
13) Adequação pessoal	“Ajudou bastante na disponibilidade no quesito horário, poder me organizar e

	estudar a hora mais conveniente.”
--	-----------------------------------

Os itens e suas definições estão a seguir:

- **Usabilidade** é a medida pela qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com efetividade, eficiência e satisfação. (ISO 9241-11).
- **Documentação** é a fonte de informação que o usuário, em caso de dúvida, poderá utilizar para um melhor entendimento da operação e uso do software. Podendo ter diversos tipos de documentação que podem ser um mediador para o entendimento do software, como por exemplo um tutorial.
- **Contexto de uso** é a associação do que o software possui e o ambiente em que ele está inserido. Está diretamente ligada a qualidade em software, norma NBR ISO/IEC 9126-1.
- **Clareza** está relacionada com a compreensão rápida e correta daquilo que é solicitado pelo software.
- **Aprendizagem** é o conhecimento que será adquirido com o software.
- **Interatividade** é a influência mútua que o software deverá possuir. Onde requer dados de entrada e opções a serem escolhidas pelo usuário.
- **Elementos gráficos** estão relacionados ao uso de mapas, imagens, gráficos, ou outro elemento visual no software.
- **Relações entre teoria e prática** é a relação que deve existir entre o real e o que o software possui.
- **Atratividade** está relacionada com o interesse que o software desperta no usuário.
- **Adequação de conteúdos** é o contexto que o software deverá possuir em relação a realidade do aluno.

- **Encadeamento de Ideias:** coesão e coerência necessárias para que sejam traçadas relações entre os conteúdos apresentados pelo software. É a capacidade que o usuário possui de traçar um raciocínio lógico, a partir do entendimento e da interpretação do conteúdo que o software oferece.

Com a aplicação da codificação axial os dados foram agrupados de acordo com os seus tipos de requisitos correspondentes: os requisitos genéricos dizem respeito aos requisitos que são necessários a todo tipo de software, requisitos educacionais são de requisitos que tem o caráter educativo e tem haver com a aprendizagem do software, e os requisitos lúdicos relacionam-se com o que eu posso realizar de ação no software. Na Tabela 2 encontra-se o resultado do grupo dos requisitos da primeira rodada.

TABELA 2. Requisitos não funcionais, agrupados segundo Codificação Axial realizada a partir da primeira rodada de entrevistas

Grupos de Requisitos	Requisitos não funcionais
Requisitos genéricos	Confiabilidade, Usabilidade, Documentação
Requisitos educacionais	Adequação de conteúdo, clareza de conteúdo, Aprendizagem, Encadeamento de ideias, Relação entre a teoria e a prática
Requisitos lúdicos	Interatividade, Presença de elementos gráficos, Atratividade

Por não haver a certeza da saturação dos dados, foi realizada uma segunda rodada de entrevistas, que possuía o mesmo perfil de entrevistados da primeira rodada. Foi obtido um novo requisitos não-funcional que foi o contexto de uso. Na Tabela 3 temos a segunda rodada de entrevistas.

TABELA 3. Grupos de requisitos da codificação axial da segunda rodada

Grupos de Requisitos	Requisitos não funcionais
Requisitos genéricos	Facilidade de uso, Usabilidade, Documentação
Requisitos educacionais	Contexto de uso, Adequação de conteúdo, clareza de conteúdo, Aprendizagem, Encadeamento de ideias, Relação entre a teoria e a prática, Encadeamento gráfico

Requisitos lúdicos	Interatividade, Presença de elementos gráficos, Atratividade
---------------------------	--

Para a terceira rodada tivemos um perfil distinto das outras entrevistas, estes entrevistados eram de diferentes áreas. Mesmo com um perfil diferente não foram obtidos novos requisitos. Na Tabela 4 temos o grupo de requisitos referentes a terceira rodada.

TABELA 4. Grupos de requisitos da codificação axial da terceira rodada

Grupos de Requisitos	Requisitos não funcionais
Requisitos genéricos	Facilidade de uso, Usabilidade, Documentação
Requisitos educacionais	Contexto de uso, Adequação de conteúdo, Aprendizagem, Adequação Pessoal, Encadeamento de ideias
Requisitos lúdicos	Interatividade, Presença de elementos gráficos, Atratividade

Na codificação seletiva do método a Teoria Fundamentada nos Dados, foi observado uma certa consistência nos grupos de requisitos encontrados. Para o grupo de requisitos genéricos o requisito não-funcional que teve maior prevalência foi a usabilidade. No grupo de requisitos educacionais houve uma saturação com diversos requisitos não-funcionais, os mais expressivos foram: contexto de uso, adequação de conteúdo, clareza de conteúdo, aprendizagem e o encadeamento de ideias. Para o grupo de requisitos referentes aos requisitos lúdicos os requisitos não-funcionais que obtiveram a saturação teórica foram: interatividade, presença de elementos gráficos, atratividade.

4. AVALIAÇÃO DOS REQUISITOS PROPOSTOS

A avaliação dos requisitos foi feita a partir de um questionário enviado para professores de diversas universidades. Dos 23 respondentes, a maioria foi da Universidade Estadual da Paraíba. O questionário foi feito no Google Forms, pela pesquisadora e orientadora. As questões que foram enviadas estão no Apêndice.

Esta avaliação teve uma escala de 0 a 2, onde representava se o conteúdo se adequava ou não como requisito não-funcional para software educacional. Foi avaliado de maneira individual cada item da codificação seletiva.

TABELA 5: Resultado Predominante na Avaliação dos Docentes

Requisito Não-Funcional	Resultado Predominante
Usabilidade	2: O requisito se adequa completamente
Documentação	2: O requisito se adequa completamente
Contexto de uso	1: O requisito se adequa parcialmente
Clareza	2: O requisito se adequa completamente
Aprendizagem	2: O requisito se adequa completamente
Interatividade	1: O requisito se adequa parcialmente
Elementos gráficos	2: O requisito se adequa completamente
Relações entre teoria e prática	2: O requisito se adequa completamente
Atratividade	2: O requisito se adequa completamente
Adequação de conteúdos	1: O requisito se adequa parcialmente
Encadeamento de Ideias	2: O requisito se adequa completamente

Com o resultado da avaliação dos docentes, pudemos constatar que os requisitos levantados com a coleta e análise das entrevistas em sua maioria se adequam completamente a software educacional. Um fator relevante extraído da avaliação dos docentes foi classificação da facilidade de uso e da usabilidade como sendo requisitos não-funcionais distintos, porém com esta avaliação foi observado que estes fatores convergem para um único ponto.

Nos anexos existem alguns comentários realizados pelos professores avaliadores a respeito da pesquisa realizada.

5. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este trabalho apresentou a importância da Engenharia de Requisitos no contexto de software educacional, com foco nos requisitos não-funcionais para este tipo de software. Já que este é um software que tem suas limitações e exigências pelo perfil de aprendizagem que este possui. Com os estudos e análises das entrevistas houve a criação de três grandes grupos de requisitos não-funcionais, que foram os genéricos, educacionais e lúdicos.

Aplicação da Teoria Fundamentada nos Dados, que procura extrair os resultados mais significativos, sendo utilizada para o levantamento dos requisitos não-funcionais. Este levantamento foi realizado com estudantes de perfis diferentes.

Concluído que a aplicação do método quanto aos requisitos não-funcionais coletados, analisados e avaliados, na sua maioria, se adequaram completamente ao contexto de software educacional. Tendo em vista as dependências/necessidades que este tipo de software possui.

Como trabalhos futuros, o estudo deve passar por análise dos outros papéis da equipe de desenvolvimento, como exemplo de psicólogos, analistas, dessa forma estes requisitos não-funcionais sejam referência quando o assunto for o desenvolvimento de software educacional.

REFERÊNCIAS

- AGILE. *Manifesto for Agile Software Development* (2001), disponível em <<http://www.agilemanifesto.org>>.
- AURUM, A.; WOHLIN, C. (Eds.) (2005) “Engineering and Managing Software Requirements”, XVIII, 478 p.
- BANDEIRA-DE-MELLO, R., CUNHA, C. (2003) “Operacionalizando o método da Grounded Theory nas Pesquisas em Estratégia: técnicas e procedimentos de análise com apoio do software ATLAS/TI”. Encontro de Estudos em Estratégia. Curitiba, Brazil.
- BATISTA, E. A. (2003) “Uma Taxonomia Facetada para Técnicas de Elicitação de Requisitos”. Dissertação de Mestrado, UNICAMP.
- BIANCHI, E. M. P. G., IKEDA, A. A. (2006) “Analisando a Grounded Theory em Administração”. IX SEMEAD - Seminários em Administração. São Paulo, Brazil.
- BOOCH, G., RUMBAUGH, J., JACOBSON, I. (2000) “UML: guia do usuário”. Tradução por Fábio Freitas. Rio de Janeiro: Campus, xx, 472 p.
- BORTOLI, L. A. DE., PRICE, A. M. DE A. (2000) “O Uso de Workflow para Apoiar a Elicitação de Requisitos”. WER 2000, disponível em <http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_wer00/bortoli.pdf> Acesso em 05/03/2010.
- BRICKER, L. (1993) “Designs for Computer-Supported Cooperative Learning in Mathematics”. Partial fulfillment of the University of Washington Generals exam.
- COELHO, A. C. M., LEAL, V. Q., GOMES, R. C. M., GOMES, L. Q. L. (2013) “Requisitos Não Funcionais no Desenvolvimento de Software Educacional: Um Estudo Utilizando Grounded Theory”. In TISE, Porto Alegre (RS).
- CHUNG, L., NIXON, B., Yu, E. and MYLOPOULOS, J. (1999) “Non-Functional Requirements in Software Engineering” Kluwer Academic Publishers.
- DAS, V. V. (2007) *Involvement of Users in Software Requirement Engineering*. In: 10th International Conference on Information Technology-INDIA. IN: IEEE.
- FALBO, R. A. (2005) “Engenharia de Software”. UFES - Universidade Federal do Espírito Santo.
- FINKELSTEIN, L, HUANG, J, INKELSTEIN, A, NUSEIBEH, B. (1992) Using Software Specification Methods for Measurement Instrument Systems. Part 1: Structured Methods. London: City University.
- FISCHER, M. C. B. O. (2001) “Estudo de Requisitos para um Software Educativo de Apoio ao Ensino da Introdução à Computação”. Dissertação de Mestrado. Local: USP – Universidade de São Paulo.
- KOTONYA, G.; SOMMERVILLE, I. (1998) “Requirements engineering: processes and techniques”. Chichester, England: John Wiley, 1998.

- LACERDA, Rafael de Alencar (2007) “Proposta de um modelo para análise de requisitos de software educativo”. Dissertação de mestrado. Brasília: UnB, 2007.
- LEITE, A. SANTA CRUZ, S. (2009) “Requisitos Não Funcionais De Usabilidade E Interação Humano Computador”. Pós Graduação Em Ciência Da Computação. Centro De Informática. Universidade Federal De Pernambuco.
- GIRAFFA, L., MARCZAK, S., E PRIKLADNICKI, R. (2005) “PDS-E: Em direção a um processo para desenvolvimento de Software Educacional”. In SBIE.
- GOMES, A. S.; PADOVANI, S. (2005) “Usabilidade no ciclo de desenvolvimento de software educativo.” In: SBIE Simpósio Brasileiro de Informática na Educação, Juiz de Fora (MG).
- PFLEEGER, S. (2004) “Engenharia de Software – Teoria e Prática”, 2ª Edição, Prentice Hall.
- MANDEL, T. (1997) “The Elements of user interface”, John Wiley and Sons: New York.
- MARTINS, A. F. (2007) “Apoio À Engenharia De Requisitos Em Ode”. UFES, Vitória.
- NETO, G.G. da C., GOMES, A. S., TEDESCO, P. (2003) “Elicitação de Requisitos de Sistemas Colaborativos de Aprendizagem Centrada na Atividade de Grupo”. In SBIE, Disponível em <http://br-ie.org/pub/index.php/sbie/article/view/262/0> Acesso em 23/02/2010.
- OLIVEIRA, C. C. et al. (2010) “Árvore de Características de Software Educativo: Uma Proposta para Elicitação de Requisitos pelo Usuário”. Universidade Federal de Uberlândia (UFU) - Minas Gerais, Brasil.
- PRESSMAN, R.S. (2002) “Engenharia de Software”. 5ª ed. Rio de Janeiro: McGraw-Hill, 843p
- SCHNEIDER, J.; VASA, R. (2006) “*Agile Practices in Software Development: Experiences from Student Projects*”. In: Australian Software Engineering Conference (ASWEC’06), IEEE 2006.
- SILVA, S. M. A. da, Bonin, M. R., Paludo, M. A. (2005) “Levantamento de Requisitos Segundo o Método Volere”. Disponível em <http://publica.fesppr.br/index.php/rnti/article/viewFile/v1n1ART2/86> Acesso em: 01/07/2009.
- SILVEIRA, S. R., BARONE, D. A. C. (1999) “Estudo e Construção de uma Ferramenta de Autoria Multimídia para a Elaboração de Jogos Educativos”. Dissertação (Mestrado em Ciência da Computação). Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre. Disponível em: http://www.minerva.uevora.pt/simposio/comunicacoes/html/Sidnei_Brasil/jogos.rtf Acesso em: 1 nov. 2009.
- SOMMERVILLE, I. (2004) “Engenharia de software”. 8ª edição. São Paulo: Pearson Assison Wesley.

- STRAUSS, A., Corbin, J. (1998) “Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory”. 2 ed. London, SAGE Publications
- THAYER, R. H., E DORFMAN, M. (1997) “Software Requirements Engineering”, 2 ed. IEEE Computer Society Press.
- VALENTE, J. A. “Por que o computador na educação?” Disponível em: http://www.ich.pucminas.br/pged/db/wq/wq1_LE/local/txtie9doc.pdf.
- Vasconcelos, A. M. L., Rouiller , A. C., Machado, C. A. F., Medeiros, T. M. M. (2006) “Introdução à Engenharia de Software e à Qualidade de Software”. UFLA/FAEPE.
- VERGNAUD, G. (1997) “The nature of mathematical concepts”. In T. Nunes e P. Bryant (Eds.), Learning and teaching mathematics: An international Perspective, Psychology Press, Hove, pp. 5-28.

ANEXOS

Anexo 1: Alguns comentários feito pelos docentes

- O trabalho é bastante relevante, sugiro apenas cuidado com este tipo de pesquisa (online, questionário), e o intervalo de notas que foi definido. Muitas vezes a sumarização dos resultados podem ser enviesados. Bom trabalho e sucesso.
- Tive certa dificuldade (ou facilidade) em responder o questionário, pois percebo que as dimensões que as questões apresentadas englobam vão desde aspectos didático-pedagógicos até a questões tecnológicas e relacionadas à própria Engenharia de Software e Análise de Requisitos, dimensões essas que apresentam aspectos de extrema importância para o desenvolvimento de software educacional. Talvez, a divisão das perguntas por dimensão, categorizando-se em seguida a importância de cada uma das dimensões abrangidas, pudesse dar uma melhor visão do nível de impacto que cada uma teria no desenvolvimento desse tipo de software.
- Só tenho uma observação com relação ao conteúdo. Na minha opinião a escolha/adequação do conteúdo cabe ao professor, exceto se fosse um tutor inteligente. Do ponto de vista do sistema como um todo são requisitos importantes, mas que, talvez, não caiba ao software essa tarefa...

APÊNDICE

Apêndice 1: Entrevista para os alunos.

- 1. Você já usou um jogo, ou algum outro software, para auxiliar o aprendizado de alguma matéria? Como foi essa experiência?**
- 2. O que é mais importante para você quando vai utilizar um software?**
- 3. Que elementos são essenciais para que um software seja utilizado por você?
Explique o porquê da importância desses elementos.**
- 4. O que falta aos softwares educacionais para que sejam melhor compreendidos?**
- 5. Qual a sua opinião sobre a facilidade de uso de um software com muitos componentes de interface (recursos gráficos, animações, mapas)?**

Apêndice 2: Questionário para os professores.

Requisitos Não-funcionais para Software Educacional

Esta pesquisa tem como objetivo analisar a Engenharia de Requisitos na perspectiva da análise e do desenvolvimento de software educacional. Dessa forma, alguns aspectos de requisitos não funcionais foram levantados, mediante questionamento realizado com alunos de graduação.

Convidamos você, professor, para avaliar os requisitos não funcionais levantados, no sentido de identificar se são úteis ou não e o quão úteis eles são.

Na avaliação, julgue os requisitos com relação à análise e desenvolvimento de software educacional, de acordo com a seguinte escala:

2: O requisito se adequa completamente / O requisito é importante

1: O requisito se adequa parcialmente / O requisito é parcialmente importante

0: O requisito não se adequa / O requisito não tem importância

Agradecemos antecipadamente pela sua colaboração.

Alanna C. Coêlho Monteiro
Aluna do curso de Licenciatura em Computação

Luciana Gomes
Orientadora da pesquisa

* Required

Quão importante é a usabilidade para um usuário de software educacional? *

Requisito: Usabilidade / Definição: é a medida pela qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com efetividade, eficiência e satisfação. (ISO 9241-11)

0 1 2

Select a value from a range of 0 to 2.

O quão uma boa documentação é importante para o uso de software educacional? *

Requisito: documentação / Definição: é a fonte de informação que o usuário, em caso de dúvida, poderá utilizar para um melhor entendimento da operação e uso do software.

0 1 2

Select a value from a range of 0 to 2.

Com relação ao contexto de uso que será inserido, quanto esse aspecto se faz relevante para software educacional? *

Requisito: contexto de uso / Definição: é a associação do que o software possui e o ambiente em que ele está inserido. Está diretamente ligada a qualidade em software, norma NBR ISO/IEC 9126-1.

0 1 2

Select a value from a range of 0 to 2.

Qual a relevância da clareza do conteúdo que será abordado possui nesse tipo de software?*

Requisito: clareza Definição: Clareza está relacionada com a compreensão rápida e correta daquilo que é solicitado pelo software.

0 1 2

Select a value from a range of 0 to 2.

Quão importante é a aprendizagem no aspecto de software educacional? *

Requisito: Aprendizagem Definição: é o conhecimento que será adquirido com o software.

0 1 2

Select a value from a range of 0 to 2.

Interatividade é significativo nesse tipo de software? *

Requisito: interatividade Definição: é a influência mútua que o software deverá possuir. Onde requer dados de entrada e opções a serem escolhidas pelo usuário

0 1 2

Select a value from a range of 0 to 2.

Qual a importância de elementos gráficos em software educacional? *

Requisito: Elementos gráficos Definição: Está relacionado ao uso de mapas, imagens, gráficos, ou outro elemento visual no software.

0 1 2

Select a value from a range of 0 to 2.

É relevante a relação entre teoria e prática na aplicação de software educacional? *

Requisito: Relação entre teoria e prática Definição: é a relação que deve existir entre o real e o que o software possui.

0 1 2

Select a value from a range of 0 to 2.

A atratividade é relevante nesse tipo de software? *

Requisito: Atratividade Definição: Está relacionada com o interesse que o software desperta no usuário.

0 1 2

Select a value from a range of 0 to 2.

Quanto é relevante a adequação dos conteúdos no software? *

Requisito: Adequação dos conteúdos Definição: é o contexto que o software deverá possuir em relação a realidade do aluno.

0 1 2

Select a value from a range of 0 to 2.

Quão importante é o encadeamento de ideias nesse software? *

Requisito: Encadeamento de ideias Definição: Coesão e coerência necessárias para que sejam traçadas relações entre os conteúdos apresentados pelo software. É a capacidade que o usuário possui de traçar um raciocínio lógico, a partir do entendimento e da interpretação do conteúdo que o software oferece.

0 1 2

Select a value from a range of 0 to 2.

Contato (email): *

O que você faz? *

Deixe aqui sua sugestão/crítica: