



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

**MAELSON MARQUES DE LIMA**

**PREÁ- Um Processo de Desenvolvimento de  
Software Educacional Baseado em Metodologias  
Ágeis**

CAMPINA GRANDE – PB  
2013

**MAELSON MARQUES DE LIMA**

# **PREÁ- Um Processo de Desenvolvimento de Software Educacional Baseado em Metodologias Ágeis**

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<sup>a</sup>. Ma. Luciana Queiroz Leal Gomes

CAMPINA GRANDE – PB  
2013

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL-UEPB

L732p Lima, Maelson Marques de.  
PREÁ [manuscrito]: um processo de desenvolvimento de software educacional baseado em metodologias ágeis / Maelson Marques de Lima. – 2013.  
49 f. : il. color.

Digitado

Trabalho de Conclusão de Curso (Graduação em Computação) – Universidade Estadual da Paraíba, Centro de Ciências e Tecnologias, 2013.

“Orientador: Profª. Ma. Luciana Queiroz Leal Gomes, Departamento de Computação”.

1. Engenharia de software. 2. Software educacional. 3. Desenvolvimento de Softwares. I. Título.

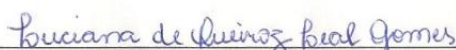
21. ed. CDD 005.1

**MAELSON MARQUES DE LIMA**

**PREÁ- Um processo de Desenvolvimento de Software  
Educativo Baseado em Metodologias Ágeis**

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 04 de outubro de 2013



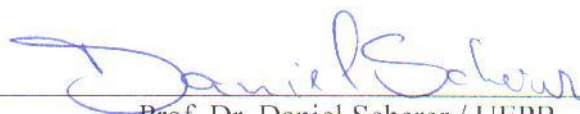
---

Prof. Me. Luciana Queiroz Leal Gomes / UEPB  
Orientadora



---

Prof. Me. Edson Holanda Cavalcante Júnior / UEPB  
Examinador



---

Prof. Dr. Daniel Scherer / UEPB  
Examinador

CAMPINA GRANDE – PB  
2013

## DEDICATÓRIA

A minha mãe, **Marli Marques de Lima**, por todo carinho, apoio e ensinamentos que me direcionaram até aqui, DEDICO.

## AGRADECIMENTOS

À **Deus** por sempre estar presente em minha vida, por me iluminar e me agraciar com paz e sabedoria nos momentos que mais precisei.

À minha querida e amada mãe Marli Marques por todo o seu amor, compreensão, carinho, conselhos e, é claro, os seus carões que me direcionaram no caminho certo.

Ao meu pai Manoel Gomes, ao meu irmão Márcio José, a minhas avós Zumira e M<sup>a</sup> do Socorro, as minhas tias Francisca, Aparecida e Suely, pela compreensão e por todo o apoio que me deram até aqui.

À professora Luciana Queiroz Leal Gomes pelas leituras sugeridas ao longo dessa orientação e pela dedicação.

À minha noiva e futura esposa Jérsica Daiane por todo o apoio, amor, carinho e paciência ao longo desses 6 anos que me mantive distante devido os meus estudos, sei que não foi fácil por isso tenho muito à agradecer.

Ao meu avô (*in memoriam*), embora fisicamente ausente, lembro-me constantemente dos seus conselhos e lições de vida dando-me força para continuar.

Aos professores do Curso de Graduação em Licenciatura em Computação da UEPB, que contribuíram ao longo desses quatro anos e meio, por meio das disciplinas, debates e conselhos, para a minha graduação e para o desenvolvimento desta pesquisa.

Aos meus amigos e primos, Francisco Lima, Caio Marques, John Kennedy, Wesley Lima Haroldo Sales, River Gomes, que me patrocinaram ótimos momentos de distração e alegria fora dos horários acadêmicos.

Aos colegas de classe e de pesquisa pelos momentos de amizade e apoio que foram essenciais para esta pesquisa.

O processo de engenharia de software típico é uma sucessão de problemas de decisão tentando transformar um conjunto de expectativas difusas em requisitos, especificações, projetos e, finalmente, código e documentação. O processo de software em cascata tradicional não conseguiu realizar o seu propósito, pois aplicado um método válido para cenários bem definidos e semi-estática. Esta hipótese está longe de ser a realidade. (NOGUEIRA J., JONES C., LUQI, 2000a, p. 1).

## RESUMO

A evolução tecnológica e as transformações no mundo produtivo têm instigado os diversos setores educacionais a refletirem sobre novos modos de construção curricular. Com o aumento de softwares educacionais no mercado surge a necessidade de se desenvolverem processos capazes de padronizarem a criação desta modalidade de software. Partindo desta conjectura, buscamos especificar um modelo de processo de desenvolvimento de software educacional que leve em consideração a complexidade que é o desenvolvimento desta modalidade de software e que agregue aspectos de avaliação ao software produzido. A metodologia de pesquisa adotada neste trabalho foi dividida em três etapas: Levantamento de requisitos, concepção e modelagem do processo e implementação do processo. Foi realizada uma revisão sistemática da literatura (RSL) a fim de obter informações sobre a evolução dos processos de desenvolvimento de software educativo, bem como as técnicas e ferramentas que são utilizadas para este processo. Esta monografia apresenta o PREÁ, um processo de desenvolvimento de software educacional, com base em metodologias ágeis e no manifesto ágil. O objetivo do PREÁ é contribuir com o desenvolvimento de software educacional, através de boas práticas da Engenharia de Software, bem como das metodologias ágeis, enfocando o fator humano. O resultado desta pesquisa é um processo que procura atender as demandas do desenvolvimento de software educacional. Foram diversas as dificuldades encontradas para o desenvolvimento deste processo, que vão desde precariedade de bases de pesquisa especializadas no assunto até a falta de especificação das técnicas, métodos e ferramentas na documentação dos trabalhos encontrados.

**PALAVRAS-CHAVE:** Engenharia de Software, Processo de desenvolvimento de software, Software Educacional, Modelagem de processos.



## ABSTRACT

The technological evolution and transformations in the productive world have instigated the various educational sectors to think about new ways of curriculum construction. With the increase in the market of educational software, there is a need to develop processes capable of creating standardize this type of software. Based on this conjecture, we seek to specify a process model for the development of educational software that takes into account the complexity that is the development of this type of software that adds features and evaluation the software produced. The research methodology used in this study was divided into three phases: Requirements gathering, design and process modeling and implementation process.

We performed a systematic literature review (SLR) in order to obtain information about the progress of development of educational software, as well as techniques and tools that are used for this process. This monograph presents PREÁ, an educational software development process, based on agile methodologies. The purpose of PREÁ is to contribute to the development of educational software, through good practices of software engineering and agile methods, focusing on the human factor. The result of this research is a process that attempts to meet the demands of the educational software development.

There were several difficulties encountered in the development of this process, ranging from precarious bases of specialized research on the subject until the lack of specification techniques, methods and tools found in the documentation of the work.

**KEYWORDS:** Software Engineering, Software Development Process, Educational Software, Process Modeling.

## LISTA DE QUADROS

<b>QUADRO 1</b> –	Relação dos estudos primários selecionados .....	29
<b>QUADRO 2</b> –	Técnicas Utilizadas no Desenvolvimento de Software Educativo.....	31
<b>QUADRO 3</b> –	Trabalhos que serviram como base para o desenvolvimento do processo proposto.....	33
<b>QUADRO 4</b> –	Dados obtidos com o estudo dos processos de Desenvolvimento de Software Educacional.....	34

## LISTA DE TABELAS

<b>TABELA 1</b> –	Métodos ágeis diminuem custos de desenvolvimento.....	25
<b>TABELA 2</b> –	Produtividade e o Custo de projetos ágeis.....	25

## LISTA DE FIGURAS

Figura 1-Produtividade de equipes ágeis. <i>Fonte: Cohn, 2011</i> .....	24
Figura 2-Seleção dos estudos primários da RSL.....	28
Figura 3-Frequência de Pesquisa Realizada Por Ano.....	29
Figura 4-Quantidade de Estudos por Tópico Pesquisado.....	31
Figura 5-Ciclo de vida do PREÁ modelado em linguagem SPEM.....	38

## LISTA DE SIGLAS

ACM	Associação para Maquinaria da Computação
ACMM	Agile Constructionist Mentoring Methodology
DDJ	Dr. Dobb's Journal - Revista popular de desenvolvimento
EAD	Educação a Distância
EUA	Estados Unidos da América
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
ISO	International Organization for Standardization
MeSoft	Methodology to Develop Educational Software
MSF	Microsoft Solutions Framework
Ph.D	Um Doutor em Filosofia
PSP	Personal Software Process
QSMA	Quantitative Software Management Associates
RBIE	Revista Brasileira de Informática na Educação
RPG	Role-playing game ou Jogo de Interpretação de Personagens
RSL	Revisão Sistemática de Literatura
RUP	Rational Unified Process (ou Processo Unificado Rational)
SBIE	Simpósio Brasileiro de Informática na Educação
SPEM	Software Process Engineering Metamodel
TSP	Team Software Process
UEPB	Universidade Estadual da Paraíba
XP	Programação extrema (do inglês eXtreme Programming)

## SUMÁRIO

1. INTRODUÇÃO.....	15
<b>1.1 Objetivo Geral</b> .....	17
<b>1.2 Objetivos Específicos</b> .....	17
<b>1.3 Principais Contribuições desse Trabalho</b> .....	17
<b>1.4 A Estrutura do Trabalho</b> .....	18
2. SOFTWARE EDUCACIONAL.....	18
3. PROCESSO DE DESENVOLVIMENTO DE SOFTWARE .....	20
<b>3.1 Processos Prescritivos</b> .....	21
<b>3.2 Processos Ágeis</b> .....	23
4. METODOLOGIA DE PESQUISA UTILIZADA.....	25
5. REVISÃO SISTEMÁTICA DA LITERATURA DOS PROCESSOS DE SOFTWARE EDUCACIONAL .....	27
6. PREÁ – PROCESSO DE DESENVOLVIMENTO DE SOFTWARE EDUCACIONAL BASEADO EM METODOLOGIAS ÁGEIS.....	33
<b>6.1 Etapas</b> .....	39
<b>6.1.1 Visão Geral</b> .....	39
<b>6.1.2 Planejamento</b> .....	40
<b>6.1.3 Desenvolvimento</b> .....	42
<b>6.1.4 Avaliação</b> .....	43
<b>6.2 Papéis</b> .....	36
7. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS .....	44
REFERÊNCIAS .....	46

## 1. INTRODUÇÃO

O processo de globalização seguido pela evolução cada vez mais rápida e contínua da tecnologia faz com que a informática esteja cada vez mais presente na vida das pessoas, seja para diversão ou trabalho. O baixo custo e os diferentes tipos de dispositivos encontrados no mercado fazem com que pessoas de baixa renda sejam incluídas no mundo digital, antes restrito a pessoas com condições financeiras mais favoráveis.

Partindo deste ponto de vista e indo em direção à educação, é natural a introdução da tecnologia na vida acadêmica dos alunos, tanto na sala de aula como fora dela. Os professores veem a inserção destas tecnologias em sala de aula como um desafio a ser vencido (Valente, 1993). São diversas as opiniões, posicionamentos e ideias, relacionadas ao tema “informática na educação”, bem como as visões que dela surgem. “A introdução de uma nova tecnologia na sociedade provoca, naturalmente, uma das três posições: ceticismo, indiferença ou otimismo” (Valente, 1993).

O software educacional é um tipo especial de software com a finalidade de ensino ou autoaprendizagem. Por ser voltado à educação, para ser desenvolvido requer o envolvimento de profissionais das mais diversas áreas, como psicólogos, professores, especialistas na área de conhecimento, e além destes, profissionais na área de informática. Além disso, para que um software seja utilizado com finalidade educativa ou em atividades curriculares, é necessário que sua qualidade de interface e pertinência pedagógica sejam previamente avaliadas de modo a atender às áreas de aplicação a que se destina e, principalmente, satisfazer as necessidades dos usuários (Lucena, 2002).

A norma ISO/IEC 9126 regulamenta a forma de avaliação e descreve a qualidade de um produto de software genérico. Assim, para se alcançar qualidade, é necessário implementar um processo de desenvolvimento definitivo e institucionalizado. O desenvolvimento de software educacional experimenta uma etapa onde existe um aumento na

demanda por softwares que atendam às restrições interdisciplinaridade e de qualidade apresentadas e assim a necessidade de um processo que contemple essas restrições (Giraffa, Marczak e Prikladnicki, 2005).

Devido à demanda, uma característica importante que deve ser considerada ao conceber um processo de software é que as entregas que devem ser rápidas. É preciso que sejam desenvolvidos mecanismos que tornem a produção de softwares nas escolas uma atividade economicamente possível, adequada à formação profissional dos professores, coerente com a infra-estrutura tecnológica da escola e, principalmente, passível de realização em tempo hábil (Lucena, 2002).

Em se tratando de software educacional, além das possíveis mudanças características do desenvolvimento de software, deve ser dada a devida atenção ao fator humano. Isto significa que pessoas diferentes, com diferentes perspectivas farão parte do processo de desenvolvimento em alguma ou algumas das suas etapas e suas expectativas devem ser atendidas no software desenvolvido.

O fato da construção do software educacional ser interdisciplinar, dele envolver profissionais de diferentes áreas, e lidar diretamente com o lado cognitivo dos usuários, faz com que um levantamento criterioso dos requisitos, tanto da área educacional quanto do sistema seja realizado. Assim, esse trabalho se propõe a definir um modelo de processo para o desenvolvimento de software educacional com base nas boas práticas do desenvolvimento de software e no manifesto ágil.

Os princípios do manifesto foram utilizados, principalmente, para lidar com os diversos profissionais envolvidos na concepção de software educacional, enfatizando os fatores humanos (Costa, Loureiro e Reis, 2010), (Das, 2007), características específicas do desenvolvimento ágil (Garousi, 2010), (Schneider e Vasa, 2006), indivíduos e suas interações, a colaboração com o cliente e ainda a rápida resposta a mudanças durante o desenvolvimento.



O processo de desenvolvimento de software educacional é algo que precisa ser amadurecido, no sentido de que são poucas as propostas para essa área que são devidamente documentadas e que passam por um processo padrão, ou até mesmo que utilizam alguma técnica ou ferramenta já validada e testada que comprove a sua eficácia.

É necessária a formulação de um processo padrão —processo já instanciado e validado— para o desenvolvimento de software educacional, onde são expressos métodos, técnicas e ferramentas, formas de comunicação entre os membros da equipe, que podem evoluir à medida que forem implementadas em outros projetos.

### **1.1 Objetivo Geral**

Definir um modelo de processo de desenvolvimento de software educacional baseado nas metodologias utilizadas no desenvolvimento ágil, que leve em consideração a complexidade de desenvolvimento deste tipo de software e que agregue aspectos de avaliação em sua definição.

### **1.2 Objetivos Específicos**

- Especificar um metamodelo de processo de software educativo a fim de definir uma terminologia única entre os vários modelos de qualidade de processo de software existentes;
- Descrever etapas que um processo de desenvolvimento de software educacional deve possuir;
- Permitir a modelagem e instanciação deste processo;

### **1.3 Principais Contribuições desse Trabalho**

As principais contribuições que se podem enumerar são a elaboração do **PREÁ**, o processo de desenvolvimento de software educacional baseado em metodologias ágeis, que foi fruto de toda a pesquisa realizada, com o objetivo de padronizar o processo de desenvolvimento de software educacional utilizando as boas práticas da metodologia ágil. E

ainda, a apresentação de uma técnica de levantamento de requisitos “**O levantamento de requisitos baseado em analogia**” que pode contribuir de forma positiva com o processo de desenvolvimento de software educacional de qualidade.

#### **1.4 A Estrutura do Trabalho**

Esta seção apresenta a estrutura utilizada na organização deste trabalho. Este capítulo apresenta a introdução com a motivação para a pesquisa realizada, objetivos e principais contribuições deste trabalho. No capítulo 2 são apresentados os conceitos de software educacional.

No capítulo 3, foi abordado o processo de desenvolvimento de software, o que é e como um processo é formulado, bem como das formas de desenvolvimento já consagradas que são: os processos prescritivos e os processos ágeis.

No capítulo 4 foram expostas as etapas que compuseram a metodologia de pesquisa utilizada neste trabalho, estas que posteriormente foram detalhadas.

O capítulo 5 detalha a revisão sistemática de literatura (RSL) realizada, que se constitui de uma pesquisa em bases de dados confiáveis em Computação. A RSL realizada por esta pesquisa teve o objetivo de captar os processos, técnicas e ferramentas que são utilizadas no desenvolvimento de software educacional.

No capítulo 6, é apresentado o PREÁ— processo de desenvolvimento de software educacional baseado em metodologias ágeis, as etapas e os papéis que compõem o PREÁ.

No capítulo 7, estão as considerações finais, que expõem a opinião do autor sobre este trabalho, além de sugestões de trabalhos futuros que poderão ser desenvolvidos para aprimorar o processo proposto.

## **2. SOFTWARE EDUCACIONAL**

O crescente interesse pela introdução da informática na educação vem sendo objeto de pesquisa em diversos centros educacionais, principalmente devido a rapidez com que os

computadores vêm ganhando espaço no ambiente educacional. A preocupação com o uso dos computadores na educação é voltada principalmente para forma de como fazê-lo, como trabalhar com estas novas tecnologias em sala de aula sem causar danos ao sistema de ensino tradicional. É certo que a inclusão de software educacional nos sistema de ensino causará mudanças no paradigma pedagógico, mas para que isso ocorra de forma positiva é necessário que padrões de desenvolvimento e processos avaliativos de software educacionais sejam amplamente difundidos nos meios educacionais, com o objetivo de melhorar a qualidade dos softwares educacionais que chegam até os sistemas de ensino (Valente, 1993).

Software educacional é definido como sendo um sistema computacional e interativo, intencionalmente concebido para facilitar a aprendizagem de conceitos específicos (Gomes e Padovani, 2005). Ainda pode-se dizer que o software educacional é um artefato que funciona como mediador do processo de ensino/aprendizagem, podendo ser utilizado autonomamente ou de forma colaborativa (Lima et. al, 2012).

O desenvolvimento de software educacional é uma atividade complexa, ela requer habilidades multidisciplinares, para propiciar o processo de ensino/aprendizagem. Baseado nas diversas formas que se pode transmitir conhecimento com o uso de softwares educacionais, estes foram divididos de acordo com a sua taxonomia Galvis (1988). Galvis foi capaz de identificar e classificar os softwares educacionais nas seguintes taxonomias: Tutoriais, Exercícios e Práticas, Simuladores e Jogos Educativos, Programação, Hipertexto/Hipermídia e Software de Autoria. Cada uma das categorias tem características específicas que as diferencia principalmente pela forma como ocorre a interação entre o usuário e máquina, mas também leva em consideração o quanto de animação, entretenimento, liberdade para criação ou desenvolvimento o software oferece para seus usuários.

O software educacional possui características específicas que devem ser levadas em consideração na hora de se pensar num processo para o seu desenvolvimento, tais como:

colaborar com o processo de ensino/aprendizagem, melhorar as formas de comunicação entre professor/aluno e possibilitar a construção de conteúdo.

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 Prikładnicki, 2005). Também se tornou inadequada 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).

### **3. PROCESSO DE DESENVOLVIMENTO DE SOFTWARE**

Um processo de desenvolvimento de software é um conjunto de passos, que fazem o sistema evoluir dos requisitos para um software executável que atenda às necessidades do cliente (Pressman, 2006). Paula Filho (2001) sugere como requisitos para um bom processo de desenvolvimento de software educacional: arquitetura, orientação do time, tempo de ciclo de processo, padrões e práticas, apoio ao aluno e apoio ao instrutor. Contudo, existem outros requisitos a serem levados em consideração, tais como os relacionados à qualidade final do produto de software; à validação do software produzido; à interface; entre outros.

Um modelo de processo de software é uma forma de representação virtual das atividades que ocorrem no mundo real em um processo de desenvolvimento de software. Ele deve ser conduzido de modo a possibilitar o devido entendimento daqueles que dele se utilizarem (Genvigir, Sant'Anna e Borrego Filho 2003).

Muitos modelos de processos para orientar o desenvolvimento de software são definidos na literatura. Modelos tradicionais, como o Modelo Cascata, ou modelos iterativos e incrementais, como o Processo Unificado, estão entre os mais utilizados (Geller, Sobrinho e Araújo, 2009). O problema das perspectivas tradicionais é que elas enfocam nos

procedimentos prescritivos e nos produtos que devem ser criados (Ambler, 2004). Isto significa que são métodos mais rígidos, que podem diminuir a flexibilidade do processo de desenvolvimento de software por não levar em consideração as dificuldades da equipe de desenvolvimento, característica que pode levar a mudanças durante seu desenvolvimento.

O processo educacional é definido como um processo promulgado cuja arquitetura e elementos são adequados para propósitos educacionais (Paula Filho, 2001). É comum clientes de projetos de software só descobrirem o que eles realmente desejam de um produto depois de utilizar alguma versão funcionando. Isto pode significar que os requisitos não foram completamente compreendidos no começo do projeto, e mudam durante o seu desenvolvimento. Motivadas por esta frustração e pela necessidade de uma rotatividade rápida de produtos, as organizações começam a implementar processos de desenvolvimento de software empíricos como as metodologias de desenvolvimento ágeis (Udo e Koppensteiner, 2003).

Um processo de software é um conjunto estruturado de atividades relacionadas, utilizadas para a produção de software. A maioria dos processos de software envolve as seguintes atividades: especificação, projeto e implementação, validação e evolução (Pressman, 2006).

Segundo Sommerville (2011), os processos de software são complexos, dependem de pessoas para tomar decisões e fazer julgamentos, não existe processo ideal. A maioria das organizações adapta os processos de desenvolvimento de software que utiliza.

Quando se fala em Engenharia de Software, duas formas de desenvolvimento já consagradas são citadas: os Modelos Prescritivos de Processo e os Modelos Ágeis de Processo. Estes últimos possuem boas práticas sugeridas pelo Manifesto para o Desenvolvimento Ágil de Software (Pressman, 2006).

### **3.1 Processos Prescritivos**

Para solucionar os inúmeros problemas na construção de software de alta qualidade no prazo e dentro do orçamento estimado é que foram propostos os modelos prescritivos de processos, que podem ser definidos como processos que prescrevem um conjunto de elementos do processo (atividades arcabouço, ações de Engenharia de Software, tarefas, produtos de trabalho) e um fluxo de trabalho (Pressman, 2006).

Existem diversos modelos prescritivos de processo, dentre os principais estão: Cascata, Espiral, Evolucionário, Incremental, cada um sugerindo um fluxo de processo diferente, mas que pode e deve ser adaptado à equipe de trabalho, à situação problema e ao próprio projeto. Em comum, todos os modelos tem o mesmo conjunto de atividades genéricas de arcabouço: concepção, planejamento, modelagem, construção e implantação.

Alguns processos prescritivos são deficientes em um ambiente de negócios mutante, pelo fato de não se adaptarem bem a mudanças e de não se adaptarem às equipes, isto se deve pelo fato deles não levarem em consideração a fragilidade das pessoas que constroem softwares de computador (Pressman, 2006). Conforme Jacobson (2002):

Modificação é aquilo para o qual o desenvolvimento de software está principalmente focado. Modificações no software que está sendo construído, modificações nos membros da equipe, modificações por causa de novas tecnologias, modificações de todas as espécies que podem ter impacto no produto que eles constroem ou no projeto que cria o produto (Jacobson, 2002).

O desenvolvimento de software educacional é um processo complexo que requer um processo flexível a mudanças, com equipes auto-organizadas e que mantenham uma forma de comunicação contínua, pelo fato desta variedade de software necessitar da participação ativa de professores, alunos, e demais interessados no processo que precisam estar trocando informações a todo o momento. Esta troca contínua de informação faz com que surjam modificações durante todo o processo de desenvolvimento, fato que toma muito tempo quando utilizados os processos prescritivos, já que estes processos optam pela entrega do

produto e depois é que avaliam a opinião do cliente para uma provável manutenção e correção de erros.

O processo de software é a estrutura técnica e de gestão estabelecida para a aplicação de ferramentas, métodos e pessoas para a tarefa de desenvolvimento de software, que compartilha experiência a cada nova aplicação em projeto, com base em projetos anteriores. Os principais objetivos que um modelo de processos de software deve atender, segundo Humphrey e Kellner (1989), são:

- Possibilitar a comunicação e o entendimento efetivo do processo;
- Facilitar a reutilização do processo (padronização);
- Apoiar a evolução do processo;
- Facilitar o gerenciamento do processo.

### **3.2 Processos Ágeis**

Para tentar sanar as deficiências nas práticas convencionais de Engenharia de Software e buscando a implementação de novas formas de desenvolvimento que conseguissem uma maior satisfação do cliente, foi elaborado um manifesto para desenvolvimento ágil de software. O desenvolvimento ágil de software surgiu com base nos valores e princípios apresentados no Manifesto Ágil (Agile, 2001)..

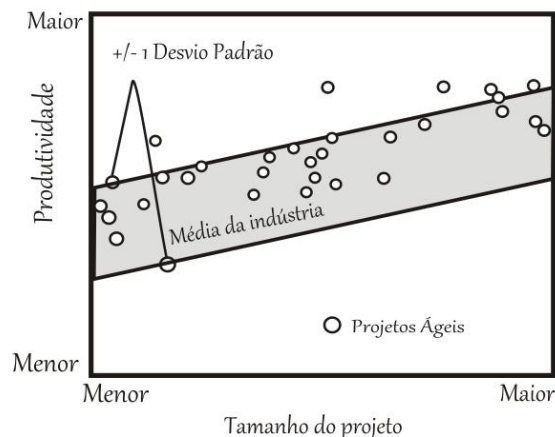
Dessa maneira, os processos ágeis, buscam diminuir a distância entre cliente e desenvolvedor, passando os clientes a serem considerados como parte da equipe de desenvolvimento e contribuindo com informações desde o início do processo. A comunicação deve ser enfatizada, de modo que seja ativa e contínua entre todos os membros que compõe a equipe de desenvolvimento (Pressman, 2006). A engenharia de software ágil sugere mudanças revolucionárias nos processos de desenvolvimento de software, com uma filosofia estruturada em suas diretrizes que priorizam indivíduos e interações ao invés de processos e ferramentas,

software funcionando ao invés de documentação abrangente, colaboração ao cliente em vez de negociação de contratos, resposta a modificações em vez de seguir um plano (Agile, 2001).

Para melhor se utilizar dos processos ágeis, no desenvolvimento de software, a equipe de desenvolvimento precisa estar motivada e bem entrosada. Assim é possível sincronizar bem as tarefas que devem ser desenvolvidas, entregar os incrementos de software operacional dentro do prazo e custo especificados. Além disso, a equipe precisa ser capaz de tolerar qualquer mudança que venha ocorrer seja no software ou na própria equipe.

Embora não haja uma medida de produtividade que seja aceita universalmente, há outros meios de se obter dados confiáveis utilizando algumas métricas para medir os chamados padrões de referência que agem como substitutos de produtividade. Alguns valores que podem ser tidos como padrões de referência são: o número de linhas de código, o número de recursos distribuídos, prazo, esforço, dentre outros que podem variar dependendo da empresa responsável pela pesquisa Cohn (2011).

Cohn (2011) menciona estudos realizado em 2008 pela QSMA, corroborados pelas pesquisas do DDJ e da VersionOne. Cohn enfatiza que nesta pesquisa foi observado que projetos ágeis são 16% mais produtivos quando comparados com projetos tradicionais, como pode ser visto na Figura 1.



**Figura 1-Produtividade de equipes ágeis.** Fonte: Cohn, 2011



Logicamente, quanto mais produtiva for a equipe, menores serão os custos do projeto. Cohn (2011) mostra detalhadamente alguns resultados obtidos com esta pesquisa que confirmam isto, como os estudos da VersionOne (2008), pesquisa online com mais de 3 mil pessoas sobre Desenvolvimento ágil, juntamente com a revista Dr. Dobbs's Journal (Ambler, 2008a). A Tabela 1 apresenta estes resultados.

**Tabela 1** Métodos ágeis diminuem custos de desenvolvimento

<b>Custos de desenvolvimento</b>	<b>DDJ</b>	<b>VersionOne</b>
Melhorou	32%	30%
Melhorou expressivamente	5%	8%

Fonte: RICO, 2008

Cohn (2011) cita uma pesquisa realizada em 2008 por David Rico, Ph.D, que examinou 51 estudos publicados sobre projetos ágeis. Nesta pesquisa é possível observar o impacto do desenvolvimento ágil sobre a produtividade e os custos para o desenvolvimento de projetos. O estudo pode ser entendido melhor ao se observar os dados expressos na Tabela 2.

**Tabela 2** O impacto do desenvolvimento ágil sobre a produtividade e o custo

<b>Categoria</b>	<b>A menor melhoria</b>	<b>Melhoria média</b>	<b>A maior melhoria relatada</b>
Produtividade	14%	88%	384%
Custo	10%	26%	70%

Fonte: RICO, 2008

#### 4. METODOLOGIA DE PESQUISA UTILIZADA

As seguintes etapas foram necessárias durante a realização das atividades:

- a. Levantamento da bibliografia específica;
- b. Concepção e modelagem do modelo de processo;
- c. Implementação do processo.

##### **a. Levantamento da Bibliografia Específica**

Um entendimento detalhado sobre os modelos de processo de desenvolvimento mais utilizados é necessário para permitir a identificação das restrições existentes nos mesmos e com isso verificar as possibilidades de melhoria que podem ser propostas.

Para tal, foi realizado uma revisão sistemática da literatura, com uma análise crítica das principais referências encontradas em cada um dos ramos necessários para a execução da proposta. Além da revisão sistemática, foi realizada uma pesquisa bibliográfica sobre linguagens para representação de processos, modelos de processo de desenvolvimento de software prescritivos e ágeis.

#### **b. Concepção do Modelo de Processo**

Um modelo de processo de software deve ser especificado de acordo com as características levantadas na atividade anterior. Durante a concepção, foram levantados os problemas mais críticos no desenvolvimento de software educacional e a partir deles o processo foi modelado.

As seguintes ações foram necessárias para se alcançar os objetivos desejados nessa etapa:

- Definição do ciclo de vida do processo;
- Definição dos componentes do processo (papéis, atividades, produtos de trabalho);
- Definição de uma técnica específica para desenvolvimento de interfaces para software educacional;
- Definição de um sub-processo para avaliação do processo de desenvolvimento de software educacional. Neste sub-processo serão utilizadas técnicas de avaliação com ferramentas de simulação de processos.

#### **c. Implementação do Processo**

Após a concepção e a modelagem da proposta é necessário realizar a sua implementação ou desenvolvimento de um modelo que possibilitasse a representação de todo o processo proposto, considerando as etapas, atividades e papéis já instanciados que compõem o processo.

O próximo passo se dá com a publicação do processo e posterior avaliação do seu funcionamento. A etapa de avaliação será realizada utilizando ferramentas que simulam todas as etapas e papéis que compõem o processo, sendo que esta etapa ainda está em fase de planejamento. O processo foi representado utilizando uma linguagem de modelagem de processos conhecida como SPEM, que favorece a representação do conjunto de atividades executadas para desenvolver, manter e gerenciar sistemas de software. .

## **5. REVISÃO SISTEMÁTICA DA LITERATURA DOS PROCESSOS DE SOFTWARE EDUCACIONAL**

A revisão sistemática da literatura (RSL) teve como objetivo identificar, na literatura científica de conferências e revistas, quais são os processos de desenvolvimento de software educativo já publicados, além de técnicas, métodos e ferramentas relacionadas. Este trabalho se baseia nas diretrizes para RSL (Kitchenham e Charters, 2007). O protocolo de pesquisa para a RSL descrita neste documento é resumido a seguir.

O primeiro passo foi a formulação de uma pergunta de pesquisa primária. O objetivo desta revisão foi encontrar evidências na literatura para responder à seguinte pergunta:

Questão de Pesquisa: Quais métodos são usados para desenvolver software educativo no contexto da Engenharia de Software?

Da questão de pesquisa, foram derivadas as palavras chave Method = process, tool, technique; e Educational software = educative software. Finalmente, os sinônimos para as palavras-chave e termos booleanos (AND, OR) foram utilizados para formar o seguinte termo de busca: (Method OR Process OR tool OR technique) AND (“educational software” OR “educative software”) AND (“software engineering”).

As bases escolhidas para realizar buscas automáticas foram: ACM Digital Library<sup>1</sup> e IEEE Xplore<sup>2</sup>. Buscas manuais foram realizadas nos arquivos da Revista Brasileira de

---

<sup>1</sup> dl.acm.org

<sup>2</sup> http://ieeexplore.ieee.org

Informática na Educação (RBIE)<sup>3</sup> e do Simpósio Brasileiro de Informática na Educação (SBIE)<sup>4</sup>.

Os critérios utilizados para a inclusão de estudos primários foram: responder à questão de pesquisa, estar escrito em Inglês ou Português, estar disponível integralmente. Foram excluídos estudos fora dos critérios de inclusão, estudos duplicados, e estudos não relevantes (excluídos pelo título, resumo, palavras-chave não relacionados aos objetivos desta RSL).

O processo de seleção dos estudos primários teve como primeiro passo a leitura de títulos e palavras-chave, leitura dos resumos, introduções e conclusões, excluindo estudos não relevantes. Dos estudos selecionados no primeiro passo, os artigos foram lidos na íntegra e selecionados de acordo com os critérios de inclusão especificados. Finalmente, uma lista com os estudos relevantes para a revisão foi criada. A Figura 2 representa o filtro utilizado na pesquisa.

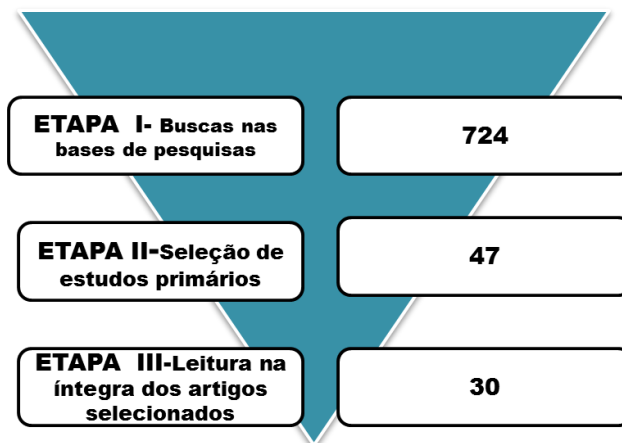
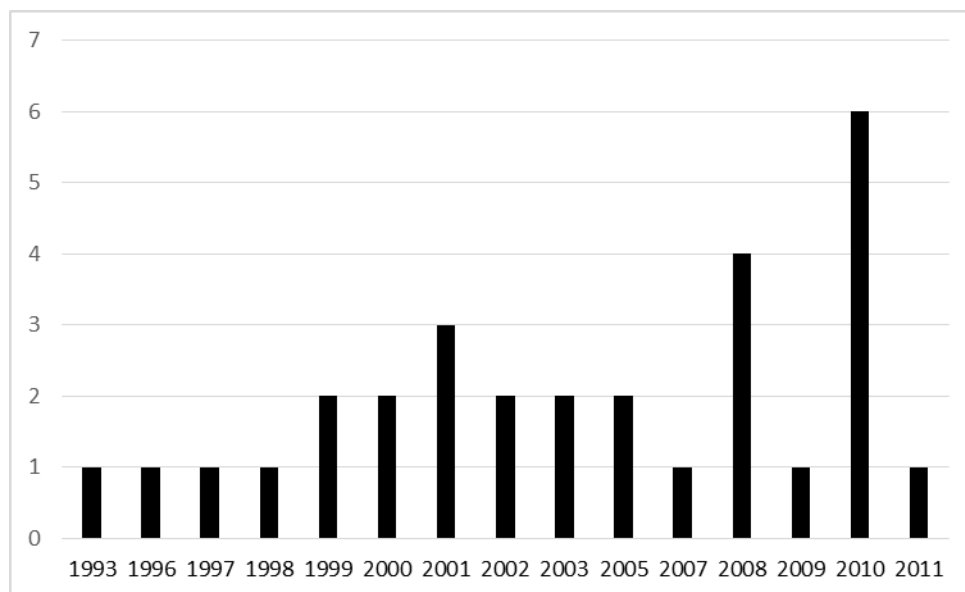


Figura 2-Seleção dos estudos primários da RSL

Como resultado de busca foram identificados 724 estudos relevantes. Destes, foram excluídos alguns com base nos títulos, palavras-chave, abstract, introdução e conclusão, resultando 47 estudos. Finalmente, foram obtidos 30 estudos primários.

<sup>3</sup> <http://www.br-e.org/pub/index.php/rbie/issue/archive>

<sup>4</sup> <http://www.br-e.org/pub/index.php/sbie/issue/archive> e [http://www.ccae.ufpb.br/sbie2010/anais/Artigos\\_Completos.html](http://www.ccae.ufpb.br/sbie2010/anais/Artigos_Completos.html)



**Figura 3-Frequência de Pesquisa Realizada Por Ano**

As pesquisas automáticas foram realizadas em 2011 e atualizadas em 2012. O maior número de estudos foi publicado em 2010 (6 estudos), sendo seguido do ano de 2008 (4 estudos). Não foi estipulado nenhum intervalo de datas nas pesquisas realizadas. A Figura 3 mostra a frequência de pesquisa encontradas por ano.

O Quadro 1 apresenta a relação de estudos primários levantados e o que foi obtido em cada estudo.

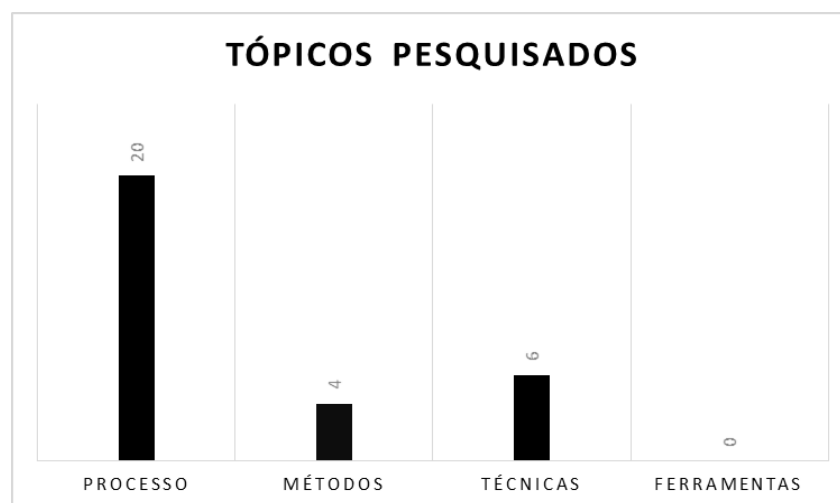
**Quadro 1- Relação dos estudos primários selecionados**

Referência	Fonte	País	O que foi obtido
(Wong, 1993)	IEEE	China	Uso de prototipagem para produção de software educativo.
(Ludi, 1996)	IEEE	EUA	Estudo conduzido com crianças para identificar aspectos relacionados ao desenvolvimento de software educativo.
(Beynon, 1997)	IEEE	Reino Unido	Apresenta a Modelagem Empírica como abordagem para desenvolver software educativo.
(Cavalcante e Domingues, 1998)	RBIE	Brasil	Metodologia tradicional com processos prescritivos de desenvolvimento de software educativo.
(Squires, 1999)	IEEE	Reino Unido	Design volátil (muda com o contexto de uso) de interface de software educativo construtivista.

(Cota et. al., 1999)	IEEE	Espanha	Design e estrutura para o desenvolvimento de software educativo hipermídia e multimídia.
(Bandoh et. al., 2000)	IEEE	Japão	Técnicas de design de software educativo que permitem a sincronização e participação do aluno e do professor.
(Batanov, Dimmitt e Chookittikul, 2000)	IEEE	Tailândia	Utilização do modelo Q&A juntamente com um framework específico para o desenvolvimento de software educativo.
(Paula Filho, 2001)	ACM	Brasil	Levantamento de requisitos, arquitetura, gerência de equipe, ciclo de vida, padrões e práticas de suporte com base em TSP e PSP. (MET)
(Lage, Zubenko e Cataldi, 2001)	IEEE	Argentina	Design, desenvolvimento, avaliação e suporte de softwares educativos com a participação de professores no processo.
(Pizzol e Zanatta, 2001)	SBIE	Brasil	Design de Interface com uso de storyboards na construção de jogos RPG educativos. (TEC)
(Pollard e Duke, 2002)	IEEE	Austrália	Design e modelagem dos processos para facilitar o desenvolvimento de software de matemática.
(Battaiola et. al., 2002)	SBIE	Brasil	Design de interfaces para jogos educativos.
(Gomes e Wanderley, 2003)	SBIE	Brasil	Processo integrado e sistemático de elicitação de requisitos de software educativo. (PRO)
(Marczak et. al., 2003)	SBIE	Brasil	Modelagem de ambientes educacionais com agentes inteligentes. (MET)
(Denis e Jouvelot, 2005)	ACM	França	Boas práticas para o desenvolvimento de jogos educativos envolvendo: design, motivação, musica, e animações.
(Giraffa, Marczak e Prikladnicki, 2005)	SBIE	Brasil	Processo de desenvolvimento de software educativo PDS-E. (PRO)
(Ivan e Josue, 2007)	IEEE	México	Descreve uma prática efetiva de desenvolvimento de softwares educativos voltados para instrumentação eletrônica virtual.
(Rankin et. al., 2008)	ACM	EUA	Design do jogo centrado no usuário.
(Kam et. al., 2008)	ACM	EUA/Índia	Práticas ágeis de design e prototipagem.
(Peres e Meira, 2008)	SBIE	Brasil	Desenvolvimento de interfaces para softwares educativos utilizando a noção de gênero discursivo. Apresenta práticas ágeis. (TEC)

(Braga, Queiroz e Gomes, 2008)	SBIE	Brasil	Design de interface.
(Barba et. al., 2009)	ACM	EUA	Desenvolvimento de jogos educativos com realidade aumentada utilizando técnicas ágeis e evolucionárias de prototipagem e design de jogos.
(Doucet e Srinivasan, 2010)	ACM	EUA	Design de jogos educativos, que simulam ambientes reais, baseado em conceitos educacionais de metáforas.
(Meerbaum-Salant e Hazzan, 2010)	ACM	Israel	Processo Agile Constructionist Mentoring Methodology (ACMM) usado como guias de desenvolvimento de projetos de softwares educativos.
(Costa, Loureiro e Reis, 2010)	IEEE	Portugal	Design de interface baseado no usuário. Usa métodos ágeis com equipes multidisciplinares. (TEC)
(Marçal et. al., 2010)	SBIE	Brasil	Levantamento e análise de requisitos para o desenvolvimento de aplicativos educacionais em dispositivos móveis. (MET)
(Oliveira et. al., 2010)	SBIE	Brasil	Apresenta a técnica de levantamento de requisitos árvore de características. (TEC)
(Fioravanti, Nakagawa e Barbosa, 2010)	SBIE	Brasil	Apresenta uma arquitetura de referência para ambientes educacionais. (EDUCAR)
[Rangel, Cury e Menezes 2011]	SBIE	Brasil	Apresenta o VCom para contribuir com a construção de ambientes educacionais Web

A Figura 4 apresenta a quantidade de estudos por tópicos que foram derivados do termo de busca (Processo, Método, Ferramenta e Técnica).



**Figura 4-Quantidade de Estudos por Tópico Pesquisado**

A maioria dos processos de desenvolvimento obtidos através desta revisão é uma adaptação de processos já existentes. Dentre as técnicas obtidas, a maioria está relacionada ao desenvolvimento de interfaces e ao levantamento de requisitos.

Foram obtidas seis técnicas relacionadas com o desenvolvimento de softwares educativos. A relação entre as técnicas encontradas, seu uso e os países que publicaram encontra-se na Quadro 2.

**Quadro 2- Técnicas Utilizadas no Desenvolvimento de Software Educativo**

Referência	Técnica	Uso	País
[Beynon 1997]	Modelagem empírica	Todo o processo	Reino Unido
[Bandoh et. al. 2000]	Sincronização e participação do aluno e do professor	Projeto (design) e design de interface	Japão
[Pizzol e Zanatta 2001]	RPG (Braistorm e Storyboard)	Levantamento de requisitos e interface	Brasil
[Peres e Meira 2008]	Gênero discursivo	Desenvolvimento de interface	Brasil
[Costa, Loureiro e Reis 2010]	Design de interface baseado no usuário	Desenvolvimento de interface	Portugal
[Oliveira et. al. 2010]	Árvore de Características	Levantamento de requisitos	Brasil

### 5.1 Discussão dos Dados Obtidos com a RSL

No Brasil existe o interesse por ambientes virtuais de aprendizagem, principalmente devido ao aumento significativo de projetos fomentando o ensino à distância (EAD). Logo, surgiram algumas técnicas e modelos que buscam padronizar o processo de desenvolvimento dessa área específica. É o caso de Oliveira et. al. (2010) e Fioravanti, Nakagawa e Barbosa (2010) que sugerem, respectivamente, boas práticas ou técnicas para levantamento de requisitos e arquitetura de referência para o desenvolvimento de ambientes educacionais.

Alguns dos processos encontrados foram propostos para resolver necessidades específicas, outros se apresentaram mais genéricos. Em sua maioria, os processos obtidos apresentam-se como uma adaptação de processos já existentes, citando alguns: prototipagem, desenvolvimento evolucionário e incremental. Outros apresentaram abordagens centradas no usuário, modelo de pergunta/resposta, design baseado em interação, como maneiras de acrescentar melhorias ao processo de software educativo.



Através da pesquisa realizada foi possível verificar a aplicação de técnicas já conhecidas, o uso e a concepção de técnicas novas no desenvolvimento de softwares educativos.

## **6. PREÁ – PROCESSO DE DESENVOLVIMENTO DE SOFTWARE EDUCACIONAL BASEADO EM METODOLOGIAS ÁGEIS**

Este trabalho propõe um modelo de processo de desenvolvimento de software educacional, com base em metodologias ágeis, o PREÁ (PRocesso de desenvolvimento de Software Educacional baseado em metodologias Ágeis). Um processo Desenvolvido utilizando as boas práticas dos métodos ágeis, que tem como foco melhorar o processo de desenvolvimento de software educacional.

A justificativa para o desenvolvimento de um processo de desenvolvimento de software educativo que utilize ágil é o fato da metodologia ágil apresentar características favoráveis para o desenvolvimento de software educacional, como processo mais flexível e adaptável a mudanças. A falta de um processo bem definido, com etapas, atividades, e papéis já instanciados, e que especificasse o uso de metodologia ágil não foi encontrado nas pesquisas realizadas.

O Quadro 3 apresenta os trabalhos que serviram como base para o desenvolvimento do processo proposto. Estes trabalhos são alguns dos estudos primários obtidos através de revisão sistemática da literatura previamente realizada (Lima et. al., 2012).

A partir desses estudos, as características do software educacional (Cota et. al., 1999), metodologia de desenvolvimento (Giraffa, Marczak e Prikladnicki, 2005), (Marczac et. al., 2003), (Cavalcante e Domingues 1998), práticas para desenvolvimento de software educacional (Ivan e Josue, 2007), e práticas ágeis (Meerbaum-Salant e Hazzan 2010), (Schneider e Vasa 2006) foram identificadas. Finalmente, as lacunas encontradas nestes estudos foram utilizadas para o desenvolvimento do PREÁ. Veja o quadro 4.

**Quadro 3- Trabalhos que serviram como base para o desenvolvimento do processo proposto**

<b>Referência</b>	<b>O que foi obtido</b>
[Cavalcante e Domingues 1998]	Metodologia tradicional com processos prescritivos de desenvolvimento de software educacional.
[Cota et. al. 1999]	Design e estrutura para o desenvolvimento de software educacional hipermídia e multimídia
[Paula Filho, 2001]	Elabora uma arquitetura para o processo de desenvolvimento de software Educacional baseado em PSP, TSP, e RUP.
[Marczak et. al. 2003]	Modelagem de ambientes educacionais (PROOGRAMA) com agentes inteligentes (Amigo).
[Giraffa, Marczak e Prikladnicki 2005]	Processo de desenvolvimento de software educacional PDS-E.
[Ivan e Josue 2007]	Descreve uma prática efetiva de desenvolvimento de softwares educacionais voltados para instrumentação eletrônica virtual.
[Kam et. al. 2008]	Propõe um ciclo de desenvolvimento que pode ser modelado para o <i>design</i> de interfaces de jogos educacionais. Baseado em interações diretas entre desenvolvedores, professores e alunos de diferentes classes sociais.
[Fioravanti, Nakagawa e Barbosa 2010]	Apresenta uma arquitetura de referência para ambientes educacionais.
[Meerbaum-Salant e Hazzan 2010]	Processo Agile Constructionist Mentoring Methodology (ACMM) usado como guias de desenvolvimento de projetos de softwares educacionais.
[Rangel, Cury e Menezes 2011]	Apresenta o VCom para contribuir com a construção de ambientes educacionais Web.

Foi elaborada uma tabela com o objetivo de comparar e apresentar o que foi obtido sobre o processo de desenvolvimento de software em cada um dos artigos citado no Quadro 3. Nesta análise, buscaram-se observar quais as etapas que compõem cada processo, quais as técnicas ou métodos de desenvolvimento ágil foram utilizadas em cada um dos processos, bem como o nome ou característica principal do processo em que foram baseados. O Quadro 4 apresenta os dados obtidos com esta pesquisa.

**Quadro 4- Dados obtidos com o estudo dos processos de Desenvolvimento de Software Educacional**

Referência	Etapas do Processo							Técnicas Ágeis						Processo Utilizado	
	Análise	Planejamento	Desenvolvimento	Programação	Implantação	Avaliação-Teste	Revisão	Comunicação direta	Pair-Programming	Releases Curtos	Flexibilidade à mudanças	Iterações <i>Just in Time</i>	<i>User Stories</i>		Timeboxing
[Cavalcante e Domingues 1998]	X	X	X	X	X	X	X								Cascata
[Fioravanti, Nakagawa e Barbosa 2010]	X	X	X			X	X	X							ProSA-RA
[Marczak et. al. 2003]	X	X	X	X	X	X	X			X					MSF Process Model, Espiral + Cascata
[Rangel, Cury e Menezes 2011]	X	X	X		X			X							Vcom - Veículo de Comunicação
[Giraffa, Marczak e Prikladnicki 2005]	X	X	X		X					X					PDS-E (Processo para Desenvolvimento de Software Educacional)
[Ivan e Josue 2007]	X	X	X	X	X	X	X	X							MeSoFT
[Cota et. al. 1999]	X	X	X	X		X		X							Maja
[Paula Filho 2001]	X	X	X		X	X	X				X				RUP
[Meerbaum-Salant e Hazzan 2010]	X	X	X	X	X	X	X								ACMM
[Kam et. al. 2008]	X	X	X	X	X	X	X	X			X		X		Design de interface baseado no usuário
<b>PREÁ</b>	<b>X</b>	<b>X</b>	<b>X</b>			<b>X</b>		<b>X</b>		<b>X</b>	<b>X</b>	<b>X</b>		<b>X</b>	<b>PREÁ</b>

PREÁ é um modelo de processo que utiliza boas práticas de desenvolvimento ágil de software, como as desenvolvidas pelo Scrum e XP, que possibilitam adaptação, com o objetivo de gerenciar e controlar a imprevisibilidades dos projetos de desenvolvimento de software educacional, sejam estes de qualquer nível. Dentre as boas práticas e técnicas

genéricas das metodologias ágeis, o PREÁ utiliza: foco nos usuários, projeto organizado em interações, planos de interação baseados em JIT (*Just In Time*).

O processo proposto agrega algumas características do Scrum, por isto ele deve ser direcionado a equipes entre 07 e 12 pessoas, que se auto-gerenciam, e se auto-organizam. Difundida pelo Scrum também é utilizada a técnica Timebox – período de tempo pré-determinado, inviolável, especificado a alguma cerimônia.

Do processo XP foram adaptadas as técnicas: Releases curtos, integração contínua, cliente presente, testes de integração e validação, produção de protótipos. Sendo estas técnicas necessárias para manter o cliente com software funcionando a todo o tempo e satisfeito com o que é produzido.

O ciclo de vida do PREÁ possui 04 etapas bem definidas: Visão Geral, Planejamento, Desenvolvimento, e Avaliação e Realimentação. Em cada etapa é necessária a participação de indivíduos que tem um papel bem definido dentro do processo, por isto definimos algumas características destes indivíduos e onde estão em cada etapa. Para melhor compreender o processo veja a Figura 5 que foi elaborada com base nos preceitos do SPEM.

## 6.1 Papéis

O PREÁ classifica os seus papéis em perfis. Essa distinção foi pensada para melhor capturar os requisitos que são necessários à construção de um software educacional e focar o aspecto humano envolvido. Os perfis sugeridos são:

- **Técnico:** desenvolvedor, testador, analista, designer.
- **Gerencial:** líder do projeto, gerente de projetos.
- **Especialista**
  - **Nível 1:** professor, profissional que conhece o domínio da aplicação.
  - **Nível 2:** pedagogo, psicólogo.
  - **Nível 3:** aluno, usuário do software.

Foram utilizados níveis diferentes para facilitar o processo de validação das entregas, onde a equipe técnicas só pode avançar com base no que for aprovado por cada nível de especialista. Dessa forma em uma entrega o produto passa pelos professores ou especialista na área de domínio, depois pelo pedagogo e psicólogo e por último pelo teste do usuário ou aluno.

O perfil Especialista representa o lado educacional do processo. Cada um dos níveis corresponde a uma esfera que deve ser levada em consideração no desenvolvimento de softwares educacionais. Ainda, todos os papéis sugeridos para o perfil Especialista representam o lado do cliente no processo. Vale salientar que o PREÁ sugere que o cliente participe de todo o processo como um componente da equipe técnica, conforme aplicado nas metodologias ágeis (Ágile, 2001).

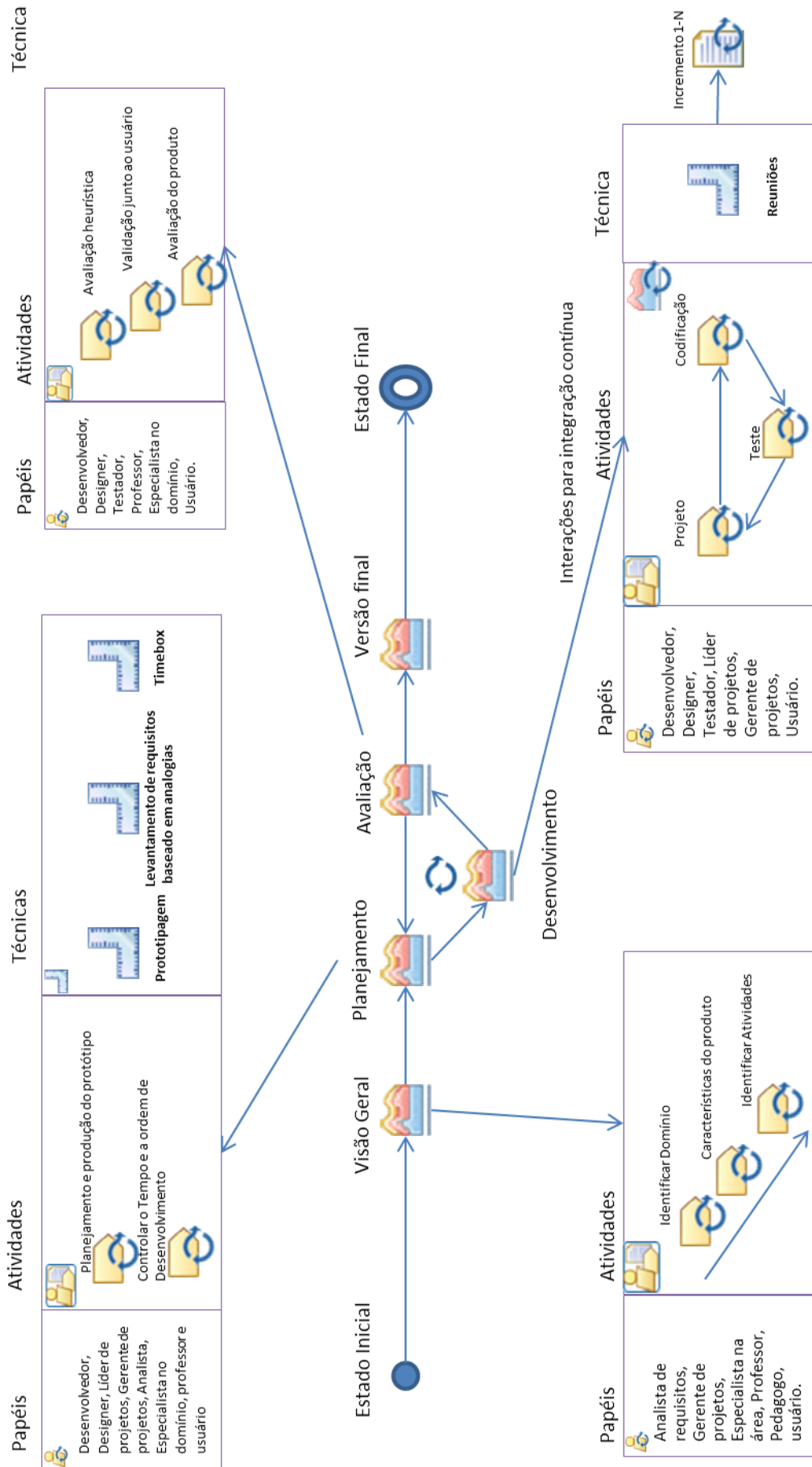


Figura 5-Ciclo de vida do PREA modelado em linguagem SPEM.

## 7.1 Etapas

### 7.1.1 Visão Geral

A etapa descrita como Visão Geral, tem como principal objetivo a identificação e compreensão do problema. Nesta fase, sugere-se que o cliente e os analistas tenham o apoio de um profissional especialista (licenciado em computação, pedagogo, psicólogo) para auxiliá-lo na identificação do problema. Com isto procura-se ponderar tanto o lado pedagógico como o tecnológico.

A conversa entre o cliente e um especialista na área educacional do software e posteriormente entre o especialista e o desenvolvedor, é sugerida para tentar reduzir as previsões incorretas realizadas pelo desenvolvedor, por falta de comunicação. Esta pode ser uma estratégia para evitar o retrabalho e diminuir o custo de desenvolvimento.

Essa forma de análise mais próxima do usuário, onde os analistas têm acesso direto à cultura, e de certa forma ao contexto de uso do produto, facilita o entendimento de qual é a ideia principal do software, o que este produto deverá ter de essencial para manter a satisfação do usuário, e conseqüentemente do cliente (Kam et. al., 2008) (Das, 2007) (Cota et. al., 1999) (Beynon, 1997).

Ao final da etapa de Visão Geral um documento com informações relacionadas ao problema a ser solucionado será gerado, e servirá de entrada para a próxima etapa. O documento a ser gerado será fruto das atividades:

- Identificar o domínio
  - Esta atividade possibilita aos analistas de requisitos compreender em qual área o software educacional solicitado estará direcionado, como história, português, matemática, ou outro ao qual o grupo deverá encontrar especialistas.
- Identificar as características do produto

- Esta atividade permite a equipe de análise entender melhor qual o objetivo que o software deverá alcançar. Quais as características que ele deverá possuir para satisfazer o cliente.
- Identificar as atividades
  - Nesta atividade serão propostas algumas ações, caso sejam necessárias, que a equipe deverá realizar para que o processo de desenvolvimento ocorra como o desejado. São exemplos: buscar especialistas, estudar o domínio, comprar materiais, preparar documentação, dentre outras que poderão surgir.

Os dados obtidos a partir destas atividades compõem o documento que servirá de entrada para a etapa de planejamento.

### 7.1.2 Planejamento

O documento de saída da etapa Visão Geral será entrada para a etapa de **Planejamento**. Para esta etapa a técnica da prototipagem é uma das técnicas indicadas. Com o uso desta técnica, procura-se conseguir uma maior proximidade entre cliente e equipe de desenvolvimento.

A decisão de utilizar protótipos no Planejamento foi tomada com base na necessidade de se trabalhar com um modelo ágil de processo desenvolvimento de software educacional que fosse adaptado por incrementos. Para realizar adaptação incremental, uma equipe ágil requer o *feedback* do cliente, e um catalisador efetivo para este *feedback* é um protótipo operacional ou uma porção do sistema operacional (Pressman, 2006).

Portanto, as atenções devem ser destinadas ao planejamento do protótipo, pois o mesmo servirá como “primeira amostra”, sendo apresentada aos clientes para avaliação. Além disto, o conceito de prototipação deve estar bem claro para o cliente, para que não seja interpretado como produto final.



Na seleção da interface adequada, deverão ser seguidos alguns princípios da Interface Homem Computador, como método de avaliação heurística (mAH) propostos por Jakob Nielsen em 1994, aos quais podemos citar:

- Liberdade e controle ao usuário;
- Flexibilidade e eficiência de uso;
- Prevenção e suporte para reconhecimento, diagnóstico e recuperação de erros;
- Consistência e padrões;
- Visibilidade do estado do sistema.

O tempo será controlado utilizando a técnica de administração de tempo *Timeboxing*, onde o tempo total de execução do projeto, seja total, ou diário, deve ser dividido em “caixas de tempo” (Lima, 2011). Cada requisito tem seu tempo de desenvolvimento estimado e fixado a priori.

Para apoiar a etapa de Planejamento e aproximar ainda mais o cliente da equipe de desenvolvimento, este trabalho propõe a técnica **Levantamento de Requisitos Baseado em Analogias**, que segue a mesma perspectiva da metodologia Adept (Markopoulos, Pycock e Wilson, 1992), utilizada amplamente no Projeto de Interfaces. Esta técnica consiste da pesquisa por softwares ou ferramentas parecidas com o que o cliente deseja. Depois de encontrado o software ou ferramenta, este é apresentado ao cliente para auxiliar a identificação, em termos de funcionalidades e aparência, daquilo que o cliente busca.

O cliente e o usuário, caso sejam pessoas diferentes, devem analisar os softwares ou ferramentas a fim de identificar o que lhes interessa em cada um, quais as funcionalidades que ele objetiva para o seu projeto, excluir aquelas que ele considera não significativas, além é claro, de expor aquelas características que ele sentiu falta durante a análise.

Dessa forma, os responsáveis pelo planejamento —analistas de projeto, designers, analistas de requisitos e desenvolvedores — terão as características principais que o software educacional deverá ter.

Estes dados serão essenciais para o planejamento e produção do protótipo. No processo proposto, o protótipo é apenas uma demonstração, que pode ser descartada, aproveitando apenas as técnicas e tecnologias utilizadas pela equipe de desenvolvimento, evitando possíveis erros no início do desenvolvimento do software (Pressman, 2006).

Sugere-se que o protótipo seja avaliado em condições reais (por exemplo: em sala de aula, por professores e alunos envolvidos no projeto). O usuário testará a navegação do software podendo validar ou não a interface e as formas de navegação utilizadas, de forma a alertar impaciência, falta de avisos, caminhos sem volta, desvio de padrão, lentidão nas respostas. Devem ser anotadas as principais observações feitas pelos usuários. Finalmente, após o levantamento dos requisitos, um modelo de projeto preliminar é elaborado para dar suporte à etapa seguinte.

### **7.1.3 Desenvolvimento**

A etapa de Desenvolvimento conta com a participação da equipe psico-pedagógica auxiliando a equipe técnica. Ao final serão capturadas as características e funcionalidades do protótipo e ele será descartado, para dar início a um ciclo envolvendo: Projeto, Codificação e Testes (ver Figura 5).

O contato direto com os usuários ajuda os desenvolvedores de software a estudar as necessidades reais que foram avaliadas como bem sucedidas pela equipe gerencial (Das, 2007). A Comunicação entre os membros de uma equipe de desenvolvimento é fator primordial no andamento do processo colaborativo e para isto a ligação entre eles tem que ser fácil e produtiva (Castro e Aguiar, 1999).

A rotina de reuniões segue o que as metodologias ágeis preconizam. Assim, são propostas reuniões diárias entre a equipe técnica, com duração de até 20 minutos. Nestas reuniões, ocorrerá: (1) Apresentação de alguma dificuldade encontrada; (2) Apresentação, caso surja, de algum requisito novo; e (3) Apresentação de algum impedimento para o andamento das atividades do projeto.

Além das reuniões diárias, o processo propõe reuniões semanais entre a equipe de desenvolvimento, equipe psico-pedagógica e usuários, com duração de até 1 hora. Nestas reuniões a equipe técnica apresenta as melhorias integradas ao software, e os usuários devem testar as melhorias incrementadas e validá-las, sugerindo outras melhorias ou aprovando o que já foi feito.

Considerando que os usuários são professores e/ou alunos, devem ser analisadas as visões de cada um deles acerca do software. Esta etapa pode ser considerada como um teste Beta, onde os analistas podem preparar um pequeno questionário (entre 3-5 perguntas) sobre alguma funcionalidade incrementada (Cota et. al., 1999).

O ideal é que os ciclos de desenvolvimento sejam curtos, propiciando a cada semana (no mínimo) a implementação de um dos requisitos, priorizando aqueles que na etapa de planejamento foram considerados essenciais para o software, onde o mesmo será integrado ao software. A comunicação deve ser constante, para que todos tenham conhecimento sobre as atividades atuais do projeto, e qual será a ação seguinte.

#### **7.1.4 Avaliação**

Na etapa de **Avaliação**, deve-se avaliar o protótipo e os demais incrementos integrados ao produto de software. Esta avaliação é feita através de três formas distintas:

- **Avaliação heurística** – realizada pelos especialistas: Designers percorrem a interface e alertam sobre algum problema encontrado em desconformidade com os padrões de interface exigidos.

- **Verificação da satisfação do usuário em relação ao produto** – realizada pelos usuários: Verificação feita por professores e alunos que fazem parte da equipe como colaboradores, com os usuários testadores (alunos) observando se o problema foi resolvido e se atende às expectativas dos usuários.
- **Avaliação do uso do produto** – realizada pelos usuários: Os usuários percorrem o software e notificam sobre qualquer dificuldade que tenham, com relação à solução apresentada (aspectos subjetivos do software).

Ao final da etapa de Avaliação devem-se observar todos os questionamentos e problemas levantados pelos usuários e discutir a possibilidade de melhorias e evolução do software. Os resultados desta análise servirão de realimentação para o processo de desenvolvimento do software.

## **7. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS**

O sucesso do sistema é completo somente quando este é implementado, aceito e utilizado pelo usuário (Das, 2007). Tendo como base essa afirmação, este artigo propõe um processo para o desenvolvimento de software educacional baseado em metodologias ágeis. Este processo tem como foco agilidade nas entregas (a cada incremento), e um maior contato com o usuário que é visto como um especialista.

A técnica de levantamento de requisitos baseado em analogias auxilia na difícil tarefa de comunicação entre cliente/desenvolvedor, evitando que sejam desenvolvidas funcionalidades indesejadas no software.

O processo modelado em SPEM encontra-se na URL:  
<http://hoome.net.br/prea/index.htm>.

Foram diversas as dificuldades encontradas para o desenvolvimento deste processo, que vão desde precariedade de bases de pesquisa especializadas no assunto até a falta de especificação das técnicas, métodos e ferramentas na documentação dos processos estudados.

Para que sejam desenvolvidos softwares de qualidade, é necessário que sejam desenvolvidas processos padrões que atentem para técnicas de avaliação e validação de um software considerado educativo.

Como trabalhos futuros, podem-se citar os seguintes: a avaliação do processo de desenvolvimento através de simulação (atividade em andamento) e por especialistas utilizando a técnica de grupo focal. A partir destas avaliações, pretende-se evoluir o processo, realizar um estudo de caso para avaliação desta evolução, e finalmente, aplicá-lo no desenvolvimento de softwares educacionais.

## REFERÊNCIAS

- Agile. *Manifesto for Agile Software Development* 2001, disponível em <<http://www.agilemanifesto.org>>. Acessado: em 21 setembro de 2012.
- Ambler, S. (2004). **Modelagem Ágil: Práticas eficazes para Programação Extrema e o Processo Unificado**. 1ª Edição. Porto Alegre, Ed. Bookman. 352 p.
- Ambler, Scott. (2008). *Agile adoption rate survey*, February. <http://www.ambysoft.com/surveys/agileFebruary2008.html>.
- Associação Brasileira de Normas Técnicas. NBR ISO/IEC9126-1 Engenharia de software - Qualidade de produto - Parte 1: Modelo de qualidade. 2003.
- Bandoh, H., Nemoto, H., Sawada, S., Indurkha, B., e Nakagawa, M. *Development of educational software for whiteboard environment in a classroom*. In IEEE, 2000.
- Barba, E., Xu, Y., MacIntyre, B., e Tseng, T. *Lessons from a Class on Handheld Augmented Reality Game Design*. In ACM, 2009.
- Batanov, D. N., Dimmitt, N. J., e Chookittikul, W. *Q&A Teaching/Learning Model as a New Basis For Developing Educational Software*. In IEEE, 2000.
- Battaiola, A. L., Elias, N. C., Domingues, R. G., Assaf, R., e Ramalho, G. L. **Desenvolvimento de um Software Educacional com base em Conceitos de Jogos de Computador**. In SBIE, 2002.
- Beynon, W. M. *Empirical Modelling for Educational Technology*, In: IEEE, 1997.
- Braga, J. M.. **Aventurando pelos caminhos da leitura e escrita de jogadores de Role Playing Game (RPG)**. Disponível em: [www.anped.org.br/1604t.htm](http://www.anped.org.br/1604t.htm). Acesso em: 15 out. 2000.
- Braga, M. M., Queiroz, A. E. M., e Gomes, A. S. **Design de Software Educacional Baseado na Teoria dos Campos Conceituais**. In SBIE, 2008.
- Cavalcante, R. J., e Domingues, T. S. **Metodologia De Desenvolvimento De Software Educativo: Aplicação Em Educação Ambiental e Microinformática**. In RBIE, 1998.
- Castro, G. M., Aguiar, T. C. (1999). **Engenharia de Software no Desenvolvimento de Software Hiperídia**. In: XXV Conferencia Latino Americana de Informática.
- Cohn, Mike. **Desenvolvimento de Software com Scrum: Aplicando métodos ágeis com sucesso**; Tradução: Aldir José Coelho Corrêa da Silva; revisão técnica: Rafael Prikladnicki. – Porto Alegre: Bookman, 2011.
- Costa, A. P., Loureiro, M. J., e Reis, L. P. *Hybrid User Centered Development Methodology*. In IEEE, 2010.
- Cota, M. P., DamiAn, A. R., Dacosta, J. G., e Boo, S.C. *Maja: A Way To Design Multimedia/Hipermedia Software To Teach Engineering*. In IEEE, 1999.

- Das, V. V. (2007). *Involvement of Users in Software Requirement Engineering*. In: 10th International Conference on Information Technology-INDIA. IN: IEEE.
- Denis, G., e Jouvelot, P. *Motivation Driven Educational Game Design: Applying Best Practices to Music Education*. In ACM, 2005.
- Doucet, L., e Srinivasan, V. *Designing Entertaining Educational Games Using Procedural Rhetoric: A Case Study*. In ACM, 2010.
- Fioravanti, M. L., Nakagawa, E. Y., e Barbosa, E. F. **EDUCAR: Uma Arquitetura de Referência para Ambientes Educacionais**. In SBIE, 2010.
- Galvis, A. H. – **Ambientes de enseñanza aprendizaje enriquecidos con computador**. Boletín de Informatica Educativa, 1(2):117-139. Bogotá, dez 1988.
- Garousi, V. (2010). **Applying Peer Reviews in Software Engineering Education: An Experiment and Lessons Learned**. IEEE Transactions on Education, Vol 53, No. 2, May.
- Geller, M. T. B.; Sobrinho, M. C.; Araújo, C. A. P. (2009). **Proposta de Customização de um Processo para Desenvolvimento de Software Educativo**. XX SBIE, Florianópolis – SC
- Genvigir, E. C.; Sant’Anna, N.; Borrego Filho, L. F. (2003). **Modelagem de Processos de Software Através do SPEM - Software Process Engineering Metamodel - Conceitos e Aplicação**”. In: Worcap – Workshop de Computação Aplicada
- Giraffa, L., Marczak, S., e Prikladnicki, R. **PDS-E: Em direção a um processo para desenvolvimento de Software Educacional**. In SBIE, 2005.
- Gomes, A. S.; Padovani, S. **Usabilidade no ciclo de desenvolvimento de software educativo**. In: SBIE Simpósio Brasileiro de Informática na Educação, Juiz de Fora (MG), 2005.
- Gomes, A. S., Wanderley, E. G. **Elicitando requisitos em projetos de Software Educativo**. In SBIE, 2003.
- Humphrey, W., S.; Kellner, M. (1989). “*Software Process Modeling: Principles of Entity Process Models*. In: (CMU/SEI-89-TR-2).
- Ivan, G., e Josue, G. **Implementing Virtual Practices using an Alternative Methodology to Develop Educational Software**. In IEEE, 2007.
- Jacobson, Ivar. *A Resounding ‘yes’ to Agile Processes-But Also More*. Cutter IT Journal, Vol. 15. nº 1. Jan. 2002.
- Kam, M., Agarwal, A., Kumar, A., Lal, S., Mathur, A., Tewari, A., e Canny, J. (2008). **Designing E-Learning Games for Rural Children in India: A Format for Balancing Learning**. In: ACM, DIS’08, Cape Town, South Africa.
- Kitchenham, B.; Charters, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. Keele University and Durham University Joint Report, 2007.

Klimick, C., e Andrade, F. **RPG & Educação**, São Paulo, 1999. Disponível em: <http://www.akrito.com.br/educ/>. (22 out. 2000).

Lage, F. G., Zubenko, Y., e Cataldi, Z. *An Extended Methodology For Educational Software Desing: Some Critical Points*. In IEEE, 2001.

Lima, E. (2011). **Time Box no Scrum**. Disponível em: <http://blog.myscrumhalf.com/2011/11/time-box-no-scrum/>, Acessado em Agosto de 2013.

Lima, M. M.; Lima, A. R.; Monteiro, A. C. C.; Cavalcante Júnior, E. H.; Gomes, L. Q. L. (2012). **Uma Revisão Sistemática da Literatura dos Processos de Desenvolvimento de Software Educativo**. In: XXIII SBIE, Rio de Janeiro (RJ).

Lucena, M. (2002). **Diretrizes para a Capacitação do Professor na Área de Tecnologia Educacional: Critérios para a Avaliação de Software Educacional**. Disponível em: <[ftp://vpn.fpte.br/cursos/Pos\\_Tecnologia\\_Educacional\\_T1/Aula\\_080510\\_Prof\\_LeonidesJustiano/Crit\\_rios%20avalia\\_\\_o%20software%20educacional.pdf](ftp://vpn.fpte.br/cursos/Pos_Tecnologia_Educacional_T1/Aula_080510_Prof_LeonidesJustiano/Crit_rios%20avalia__o%20software%20educacional.pdf)>, Acessado em agosto de 2013.

Ludi, S. *Children and Educational Software: Software Design Under the Microscope*. In: IEEE, 1996.

Kam, M., Agarwal, A., Kumar, A., Lal, S., Mathur, A., Tewari, A., e Canny, J. *Designing E-Learning Games for Rural Children in India: A Format for Balancing Learning with Fun*. In ACM, 2008.

Luqi. *Software Evolution Through Rapid Prototyping*. IEEE Computer. May, 1989.

Marçal, E., Lima, L., Júnior, M., Viana, W., Andrade, R., e Ribeiro, J. W.. **Da Elicitação de Requisitos ao Desenvolvimento de Aplicações de Mobile Learning em Matemática**. In SBIE, 2010.

Marczak, S., Giraffa, L., Almeida, G., e Blois, M. **Modelando um ambiente de aprendizagem na Web: a importância da formalização do processo de desenvolvimento**. In SBIE, 2003.

Markopoulos, P.; Pycocock, J.; Wilson, S. (1992). *ADEPT - A task based design environment*. Queen Mary and Westfield College, UK.

Moran, J. M. **Saberes e Linguagens de educação e comunicação**, editora da UFPel, Pelotas, 2001, pp. 19-44.

Meerbaum–Salant, O., e Hazzan, O. *An Agile Constructionist Mentoring Methodology for Software Projects in the High School*. In ACM, 2010.

Nielsen, Jakob. 1994. *10 Usability Heuristics for User Interface Design*. Disponível em: <http://www.useit.com/papers/heuristic/>. Acesso em novembro de 2012.

Nogueira, J.C., Luqi, and Berzins, V. *A Formal Risk Assessment Model for Software Evolution*. Paper submitted to SEKE 2000.



Oliveira, C. C., Oliveira, D. C., Oliveira, C. F., Cattelan, R. G., e Souza, J. N. **Árvore de Características de Software Educativo: Uma Proposta para Elicitação de Requisitos pelo Usuário.** In SBIE, 2010.

Paula Filho, W. P. **Requirements for an Educational Software Development Process.** In ACM, 2001.

Peres, F. e Meira, L. **Dialogismo: a idéia de gênero discursivo aplicada ao desenvolvimento de software educativo.** In SBIE, 2008.

Pizzol, C. D., e Zanatta, A. L. **O RPG como técnica na construção de software educacional: A Revolução Farroupilha.** In SBIE, 2001.

Pollard, J., e Duke, R. **A Software Design Process to Facilitate the Teaching of Mathematics.** In IEEE, 2002.

Pressman, Roger S. , **Engenharia de Software.** 6ª ed, McGraw-Hill, 2006, 752 p.

Rangel, V. G.; Cury, D.; e Menezes, C. S. (2011). **VCom: Uma Abordagem para Modelagem de Ambientes Colaborativos para Apoiar a Aprendizagem.** In: XXII SBIE - XVII WIE.

Rankin, Y. A., McNeal, M., Shute, M. W., e Gooch, B. **User Centered Game Design: Evaluating Massive Multiplayer Online Role Playing Games for Second Language Acquisition.** In ACM, 2008.

Ricardo, C. **Educational Software Features: What Do Educators Want?.** In: IEEE, 1984.

Sommerville, Ian. **Engenharia de Software.** 9ª ed, São Paulo: Person Prentice Hall, 2011, 529 p.

Squires, D. **Educational Software and Learning: Subversive Use and Volatile Design.** In IEEE, 1999

Schneider, J.; Vasa, R. (2006). **Agile Practices in Software Development: Experiences from Student Projects.** In: Australian Software Engineering Conference (ASWEC'06), IEEE 2006.

Udo, N.; Koppensteiner, S. (2003). **Will agile development change the way we manage software projects? Agile from a PMBOK Guide perspective.** Projectway.

Valente, J.A. (1993). Por Quê o Computador na Educação. Em J.A. Valente (Org.), Computadores e Conhecimento: repensando a educação (pp. 24-44). Campinas, SP:Gráfica da UNICAMP.

VersionOne, 2008. **The state of agile development: Third anual survey.** Posted as a Downloadable PDF in the Library of White Papers on the VersionOne website. [http://www.Versionone.com/pdf/3rdAnnualStateOfAgile\\_FullDataReport.pdf](http://www.Versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf).

Wong, S. **A Quick Prototyping Approach to Educational Software Development - The Hypermedia Experience.** In: IEEE, 1993.