

YPEDUC: Uma adaptação de Metodologia Ágil para o Desenvolvimento de Software Educativo

Angélica F. Medeiros¹

¹ Universidade Estadual da Paraíba - Campus VII - Patos/PB

¹ Faculdades Integradas de Patos - Patos/PB

angelicafelixx@gmail.com

Abstract. *This work highlights the importance of adopting Software Engineering practices in Educational Software development (SE) projects, analyzing agile methods as possible solutions to problems related to these applications. One of these methods is the EasYProcess Software Development Process (YP), which was created to support the development of academic projects. However, the construction of an SE addresses specific issues that differ from software without pedagogical objectives. In this sense, the objective of this work focuses on listing some criteria to adapt this process to the development of educational software, thus resulting in YPEduc.*

Resumo. *Este trabalho destaca a importância de adotar práticas da Engenharia de Software em projetos de desenvolvimento de Software Educativo (SE), analisando os métodos ágeis como possíveis soluções para os problemas relacionados a estes aplicativos. Um destes métodos é o Processo de Desenvolvimento de Software EasYProcess (YP) que foi criado com o intuito de atender o desenvolvimento de projetos acadêmicos. No entanto, a construção de um SE aborda questões específicas que diferem dos software sem objetivos pedagógicos. Neste sentido, o objetivo deste trabalho se concentra em elencar alguns critérios para adaptar este processo ao desenvolvimento de software educativo surgindo assim o YPEduc.*

1. Introdução

O software educativo é um recurso de imensurável importância na aprendizagem, trazendo de forma lúdica e prazerosa o desenvolvimento prático dos conhecimentos escolares, podendo auxiliar o educador como um recurso a mais em sua prática educativa no processo de ensino-aprendizagem [Coomans and Lacerda, 2015; Da Silva et al., 2016; Almeida et al., 2018].

No entanto, alguns softwares educativos não passam por processos de desenvolvimento que garantam uma qualidade pedagógica efetiva. Lacerda [2007] enfatiza que uma das principais causas para tais problemas é a falta de diretrizes para o processo de desenvolvimento deste tipo de aplicação e ainda neste cenário, [Ferreira and Rodrigues 2015] reforçam a relevância de conceber interfaces educativas alinhadas às estratégias que direcionam ao atingimento dos processos cognitivos.

Devido à especificidade do uso destes produtos, as questões colocadas à equipe de desenvolvimento são muito peculiares [Rodrigues et al., 2016], pois além das dificuldades gerais inerentes ao desenvolvimento de software (por exemplo: incorporação de novos requisitos aos protótipos, manutenção de softwares, etc.) soma-se uma nova dimensão: a educacional [Perry, 2006; Junior and Aguiar 2018]. E apesar disto, estudos apontam que maioria dos profissionais utilizam processos de desenvolvimento de software clássicos e há pouca discussão de ações cognitivas durante o desenvolvimento de SE [Abreu et al., 2012; De Lima et al., 2012; Neto et al., 2015].

A partir da necessidade de processos cada vez mais rápidos, confiáveis e eficientes, com o objetivo de agilizar o desenvolvimento de software, surgiu o Movimento Ágil. Soares [2004] sintetiza a ideia dos métodos ágeis, reforçando o enfoque nas pessoas e não em processos ou algoritmos. Uma característica das metodologias ágeis é que elas são adaptativas ao invés de serem preditivas.

Alguns processos representantes do desenvolvimento ágil são XP (Extreme Programming) e o SCRUM, que são consideradas metodologias compatíveis e complementares, não sendo raro vermos empresas que utilizam ambas as metodologias. O XP é mais focado em práticas integradas de engenharia de software enquanto que o SCRUM foca-se nas práticas de gerenciamento do projeto [Neto, 2008]. Outro exemplo entre as metodologias ágeis é o easYProcess [PET/UFCG, 2007], que foi uma metodologia desenvolvida através de um projeto da Universidade Federal de Campina Grande (UFCG).

O easYProcess (YP) é uma metodologia de desenvolvimento de software criada com o intuito de atender ao desenvolvimento de projetos acadêmicos e por isso é simples e de fácil entendimento pelos integrantes da equipe, porém é completo e robusto a ponto de gerar produtos de qualidade.

Tendo em vista os aspectos observados, levantou-se a seguinte questão: O YP é passível a adaptações a fim de atender as especificidades do desenvolvimento de um software educativo? Neste sentido, o presente trabalho tem como objetivo elencar critérios essenciais para o desenvolvimento de softwares educativos de modo a incorporá-los no processo YP e propõe a utilização desta adaptação denominada YPEDUC.

2. easYProcess (YP)

O easYProcess (YP) é um processo de software simplificado, desenvolvido através de um projeto da Universidade Federal de Campina Grande (PET/UFCG, 2007), amparado em práticas do XP (eXtreme Programming), RUP (Rational Unified Process) e AM (Agile Modeling), para possibilitar o desenvolvimento de software de forma ágil.

O YP foi determinado com base nas particularidades de um projeto para desenvolvimento de software acadêmico, mas mesmo sendo desenvolvido para projetos curtos e simples, é uma metodologia robusta e completa, com processo bem definido e bastante simples de aprender e implantar. As fases do easYProcess são descritas por Garcia et al (2004) como: definição de papéis, conversa com o cliente, inicialização, planejamento, implementação, finalização da iteração e versão do produto, tais fases estão representadas na Figura 1.

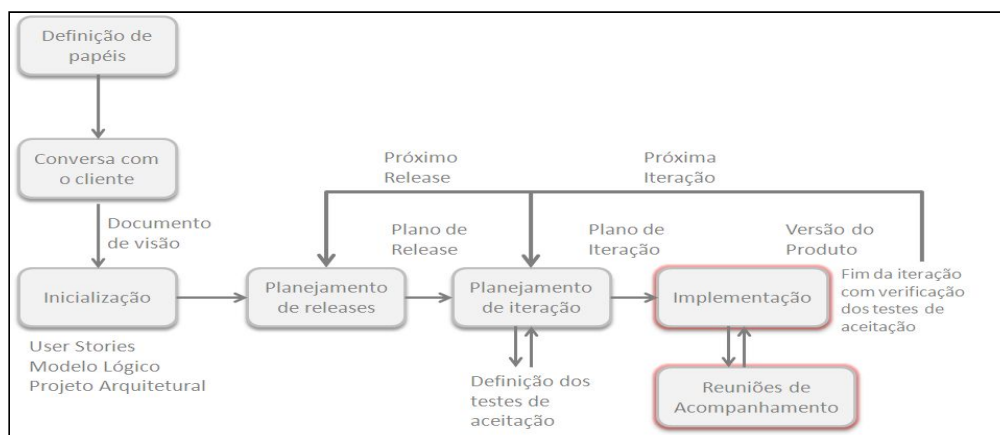


Figure 1. Síntese do Fluxo do Processo de Desenvolvimento YP

Pode-se perceber que essa metodologia possui algumas semelhanças com as outras metodologias apresentadas, como por exemplo, necessidade de produção de poucos artefatos, necessidade de reuniões de acompanhamento, dentre muitas outras. No entanto, como já foi mencionado é uma metodologia oriunda do meio acadêmico e ideal para ser aplicada com equipes ainda menores que o delimitado pelo XP e pela metodologia Scrum, se adaptando melhor, então, ao caso estudado. Por essas razões, foi escolhido como objeto desse estudo.

3. Agilidade no Desenvolvimento de Software Educativo

A metodologia utilizada para o desenvolvimento de SE deve auxiliar a equipe de desenvolvimento a encontrar a melhor forma de atingir o objetivo do software, respeitando as necessidades de aprendizagem e especificações do projeto. Para tanto, metodologias de desenvolvimento têm sido criadas, testadas e adaptadas há décadas [BOYLE et al, 2006].

Para BOYLE et al., [2006] é necessário atualizar os métodos para o desenvolvimento ágil, que é estruturado e adaptável às circunstâncias do contexto educacional. O autor defende ainda que atingir essa flexibilidade é útil para se concentrar inicialmente nas funções-chave do desenvolvimento como análise das necessidades do aluno, desenvolvimento, entrega e avaliação.

Em contrapartida, fica clara a falta de diretrizes superiores para execução de um método de desenvolvimento de SE, sendo necessário que a equipe de projeto opte por uma metodologia simples, que possa ser rapidamente assimilada e aplicada, para que assim facilite e torne ágil a implantação e manutenção do processo.

Tendo em vista os aspectos analisados, desenvolver um software com caráter efetivamente educativo de forma ágil e eficiente significa observar detalhes em cada uma das etapas de produção do mesmo, escolhendo formalismos que potencializam as características educacionais, e produzindo apenas artefatos realmente necessários durante o processo. Por outro lado, constata-se que uma a principal mudança que deve ser realizada neste segmento é de caráter comportamental, tendo em vista que os pontos críticos deste problema envolve essencialmente a maneira como a equipe de projeto utiliza as informações inerentes do contexto educativo.

Uma vez que se tenha conhecimento destas melhorias é importante que sejam elencados elementos essenciais que interferem na concepção deste tipo de software, para que tais elementos possam ser inseridos em uma metodologia de desenvolvimento.

Portanto, seria necessário adaptar uma metodologia para utilização neste contexto, gerando um conjunto de recomendações que pudessem ser inseridas em uma metodologia que além de ágil, fosse utilizada no âmbito acadêmico para que desta forma tais recomendações pudessem ser incorporadas na formação de futuras equipes de projeto, de maneira que viessem a se tornar boas práticas de desenvolvimento de software educativo.

Diante disto, destaca-se o fato do processo de desenvolvimento YP ter sido planejado para academia e idealizado para o uso por equipes com nenhum conhecimento ou prática no uso de metodologias de desenvolvimento e processos. Neste sentido, ele foi desenvolvido para ser simples, mas completo o suficiente para que o aluno tenha um contato com uma metodologia séria de desenvolvimento, o que o torna passível a adaptações para diferentes contextos. Dado o exposto, serão apresentadas algumas recomendações para que o YP possa ser melhor aplicado para o desenvolvimento de software educativos.

4. Trabalhos Correlatos

Alguns estudos já vêm sendo realizados na tentativa de potencializar o desenvolvimento de SE. Alguns trabalhos focam seus estudos em etapas pontuais do processo de desenvolvimento, a exemplo disto [Gomes and Wanderley 2003] e [Marçal et. al., 2010] focam na etapa de levantamento e análise de requisitos e propõe elencar critérios específicos para a construção de software educativos.

Outros trabalhos focam na proposta de metodologias, nesse cenário [Braga et al., 2013] propõe a metodologia INTERA para criação de objetos educacionais que considera em suas etapas todas as atividades pedagógicas que são baseadas em metodologias de design instrucional. No próprio trabalho, a INTERA foi avaliada e apontada como adequada para questão pedagógica, mas devido a complexidade de aplicação necessitava de treinamentos para os profissionais.

Outra vertente de trabalho relacionado a este, destaca [Ferreira e Rodrigues 2015] que defende a ideia de que a concepção das interfaces educativas deve ser guiada por implicações de design que direcionam o atingimento dos processos cognitivos e para tanto, os autores apresentam o processo de design com ênfase na prototipação evolutiva das interfaces do jogo educativo Nicetown.

A partir da análise dos trabalhos relacionados a este, é interessante reforçar a preocupação existente em torno do processo de desenvolvimento de software educativos. tendo em vista que, tal processo influencia fortemente na qualidade do produto entregue e conseqüentemente da qualidade pedagógica do mesmo. No entanto, poucos trabalhos focam na agilidade destes processo.

E é a partir desta lacuna que o presente trabalho traz contribuições ao propor uma adaptação do processo YP que se preocupa com critérios técnico pedagógicos para desenvolver o SE, mas reforça também a necessidade de observar princípios do

manifesto ágil como satisfazer o cliente, através da entrega adiantada e contínua de software de valor, ambiente sustentável de desenvolvimento e principalmente, envolver pessoas relacionadas ao cenário da aplicação.

5. Critérios de Adaptação Do EasyProcess para o Desenvolvimento de Software Educativo

Com a revisão da literatura ficou evidente que alguns pontos são de extrema relevância num processo de desenvolvimento de educativo, surgindo à necessidade de modificar alguns pontos do YP com o objetivo de contemplar características peculiares deste tipo de aplicativo computacional.

Algumas questões específicas foram levantadas como indispensáveis para que o software seja efetivamente educativo e por isso devem ser levadas em consideração no decorrer do projeto de desenvolvimento, dentre elas: o processo de ensino-aprendizagem deve ser um processo dinâmico e ativo; o próprio aluno precisa ser o construtor do seu conhecimento; o professor assume o papel de mediador do processo de aquisição do conhecimento do aluno.

Neste sentido, outro fator em destaque após a revisão da literatura é a necessidade da mudança comportamental por parte da equipe de projeto, tendo em vista que os pontos críticos deste problema envolvem essencialmente a maneira como as informações inerentes ao contexto educativo são utilizadas pela equipe. Para um melhor entendimento, o quadro 1 resume os critérios de adaptação do YP para o desenvolvimento de se, encaixando cada critério nas etapas do processo.

Quadro 1 - Critérios de Adaptação nas Etapas no YP

| ETAPAS | CRITÉRIOS DE ADAPTAÇÃO |
|-----------------------------------|---|
| Definição de Papéis | A constituição de uma equipe multidisciplinar e multiprofissional com especialistas da educação; |
| Conversa com o cliente | <ul style="list-style-type: none"> • A equipe multidisciplinar deve participar da conversa com o cliente; • A análise de requisitos deve respeitar questões como público-alvo, contexto, conteúdo e avaliação; • Além do Perfil do Aluno deve ser levado em consideração o perfil do Professor Mediador; • No Perfil do Usuário deve ser considerado também o conhecimento prévio em relação ao conteúdo abordado pelo software; • Durante a definição dos Objetivos de Usabilidade, o processo cognitivo para a aprendizagem do conteúdo educacional deve ser traçado; • É indispensável que professor e aluno também participem da produção dos objetivos de usabilidade; |
| Inicialização | <ul style="list-style-type: none"> • Considerar o contexto de uso educacional na concepção das User Stories; • Na fase de prototipação da interface, devem ser levadas em consideração as múltiplas inteligências e o perfil dos alunos; • A equipe multidisciplinar deve participar da produção das User Stories e da Prototipação da interface; |
| Planejamento | • Sem alterações; |
| Implementação | • Sem alterações; |
| Reuniões de Acompanhamento | • Sem alterações; |
| Versão do Sistema | • Realizar Teste de Aprendizagem e Teste de Qualidade Pedagógica. |

6. O YP Adaptado Ao Cenário Educacional: Ypeduc

Nesta seção é apresentado o Fluxo de Trabalho do YPEduc, as etapas ou artefatos que sofreram alterações irão aparecer em destaque com letra vermelha na Figura 02. Em seguida, é explicada cada uma das fases com mais detalhes.

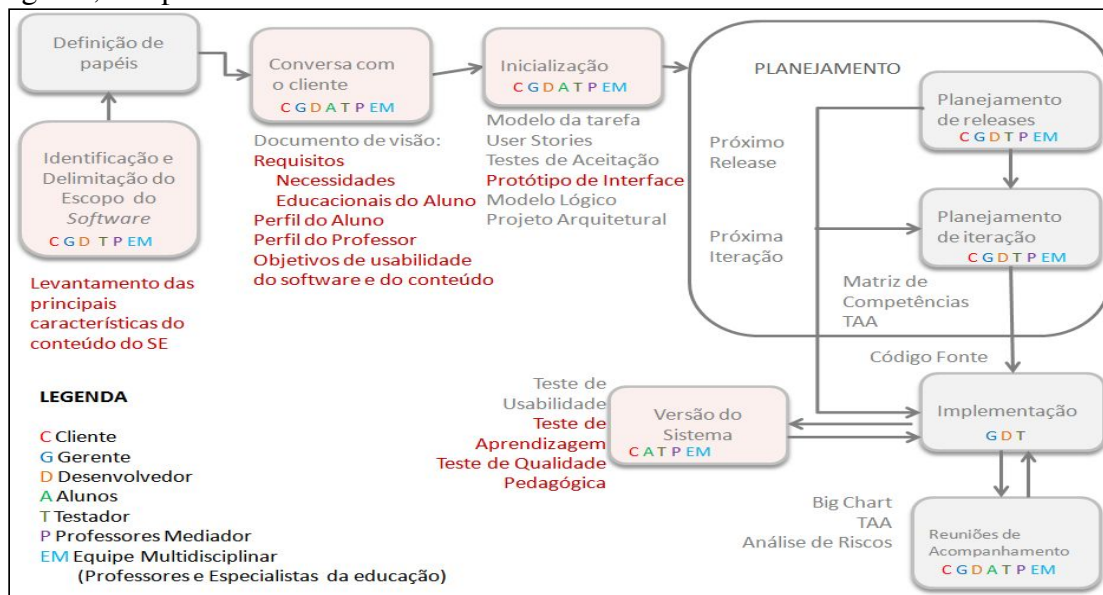


Figura 2. Fluxo do Processo YPEduc

A. Identificação do Escopo do Problema: Além da identificação do Escopo do software, são definidas as características inerentes ao conteúdo que será abordado para que sejam identificados os seus objetivos. Deve-se estabelecer também qual o papel do software educativo no processo de aprendizagem, como ele vai ser utilizado e qual o grau de importância que o mesmo tem em relação a este processo. Toda a equipe deve participar desta etapa.

B. Definição de Papéis: Um papel constitui um conjunto de responsabilidades que determina qual será o comportamento de uma pessoa durante o processo. No YPEduc recomenda-se a presença de: cliente, usuário (aluno), gerente, desenvolvedor, testador, Professor (Mediador) e Equipe Multidisciplinar (Professor Conteudista e Pedagogos ou Especialistas da Educação).

C. Conversa com o Cliente: A primeira conversa com o cliente consiste em uma coleta inicial de informações sobre o sistema que será desenvolvido. Devem participar desta etapa o cliente, gerente, Desenvolvedor, Testador, Aluno, Professor Mediador e Equipe Multidisciplinar. Os artefatos produzidos durante esta etapa são:

- Visão sobre o Software Educativo: A partir dessa conversa o cliente e os desenvolvedores passam a ter um entendimento comum a respeito do sistema.
- Levantamento dos Requisitos Funcionais e Não Funcionais: A análise de requisitos em softwares educativos deve respeitar questões como público-alvo, contexto, conteúdo e avaliação.
- Levantamento do Perfil do Usuário/Aluno: Para o projeto de SE, no Perfil do Usuário deve ser considerado também o conhecimento prévio em relação ao

conteúdo abordado pelo software e diferente do que acontece em projetos de softwares comerciais.

- **Levantamento do Perfil do Professor Mediador:** O perfil do professor mediador deve trazer informações sobre seu conhecimento sobre o conteúdo, sua experiência com tecnologias na educação, as principais dificuldades encontradas para lecionar o conteúdo abordado pelo SE, dentre outras.
- **Levantamento dos Objetivos de Usabilidade:** Durante a definição dos objetivos de usabilidade deve ser levada em conta além da aprendizagem do sistema a aprendizagem do conteúdo educacional.

D. Inicialização: Logo após a equipe de desenvolvimento ter uma ideia geral sobre o problema a ser resolvido, devem ser iniciadas algumas atividades de análise do sistema. Devem participar desta etapa o cliente, gerente, Desenvolvedor, Testador, Aluno, Professor Mediador e Equipe Multidisciplinar. Os artefatos produzidos durante esta etapa são:

- **Modelagem da Tarefa:** É uma representação, hierárquica, de como a tarefa é realizada pelo usuário. A construção do modelo da tarefa é precedida pela análise da tarefa, durante a análise da tarefa aspectos relacionados às características da tarefa, ao perfil do usuário e aos objetivos de usabilidade devem ser considerados.
- **Levantamento das User Stories e Testes de Aceitação:** São especificações feitas pelo cliente a fim de definir o que o sistema deve ter/fazer. O uso no contexto educacional necessita ser considerado e respeitado para a produção das User Stories.
- **Geração do Protótipo da Interface:** Em um SE, na fase de prototipação da interface deve ser levado em consideração às múltiplas inteligências e o perfil dos alunos, a fim de contemplar o máximo de alunos possível com o aprendizado intuitivo.
- **Elaboração do Projeto Arquitetural:** Este artefato tem como objetivo descrever o funcionamento do sistema num alto nível de abstração.
- **Geração do Modelo Lógico de Dados:** Se o sistema a ser desenvolvido envolve um banco de dados, o desenvolvedor deve construir o modelo lógico de dados capaz de representar a estrutura lógica do banco de dados.

E. Planejamento (Release e Iteração): Concluídas as atividades de Inicialização, deve ser iniciado o Planejamento do projeto. Primeiramente, a equipe deve estar ciente do tempo disponível para o desenvolvimento do projeto, e a partir de então, com o cliente, definir o número de releases e iterações necessárias para a conclusão do mesmo. Devem ser realizados pequenos releases na tentativa de garantir uma maior interação entre a equipe de desenvolvimento e o cliente, sendo assim, as User Stories alocadas em cada release só podem ser consideradas como finalizadas após a realização dos testes de aceitação. O plano da iteração consiste em quebrar as User Stories em atividades menores (caso necessário) e em seguida alocar os membros da equipe de desenvolvimento como responsáveis pela realização de cada atividade. A alocação da tarefa ao desenvolvedor pode ser feita com o auxílio da Matriz de Competências, uma tabela simples que informa quais as habilidades dos membros da equipe de desenvolvimento e o tempo que cada um disponibiliza para o projeto. Devem participar

desta etapa o cliente, gerente, Desenvolvedor, Testador, Professor Mediador e Equipe Multidisciplinar.

F. Implementação: É a realização das atividades estabelecidas no plano da iteração. Tem como principal artefato o código do sistema. Para que a implementação do sistema resulte em uma boa codificação é necessário que a equipe de desenvolvimento considere algumas práticas: Integração Contínua, Boas Práticas de Codificação, Propriedade Coletiva de Código, Testes e Pequenos Releases. Devem participar desta etapa o gerente, Desenvolvedor e Testador.

G. Versão do Produto: Nesta fase, é necessário que além dos Testes de Usabilidade também ocorram os Testes de Aprendizagem e de Qualidade Pedagógica, criado pela equipe multidisciplinar para avaliar a aprendizagem dos alunos e o caráter pedagógico do software. Devem participar desta etapa o cliente, Testador, Aluno, Professor Mediador e Equipe Multidisciplinar.

- Testes de Usabilidade: Verificam se os objetivos de usabilidade especificados pelo cliente e pelo usuário foram satisfeitos, além de averiguar como está sendo o processo de interação entre o usuário e o sistema. Os cenários de teste de usabilidade devem buscar avaliar o desempenho de usuários típicos na realização de tarefas comuns.
- Testes de Aprendizagem: Verifica se os alunos estão realmente aprendendo o conteúdo substancial do aplicativo, ou seja, é avaliado o que o aluno retém de conhecimento após a sua utilização. Diversos métodos de avaliação podem ser aplicados para o Teste de Aprendizagem, como provas, questionários, minitestes, dinâmicas, dentre outras.
- Testes de Qualidade Pedagógica do Software Educativo: Este teste verifica os aspectos pedagógicos (facilidade no acesso às informações, adequação a faixa etária, clareza nas informações, tipo de exercícios), do conteúdo (fidelidade ao objeto, coerência de apresentação do conteúdo, correção dos exercícios, promoção da criatividade e motivação dos usuários) e aspectos gerais (alcança os objetos propostos, contribui para a aprendizagem dos conteúdos).

7. Conclusão

Desenvolver um software com caráter efetivamente educativo de forma ágil e eficaz significa observar detalhes em cada uma das etapas de produção do mesmo, escolhendo formalismos que potencializam as características educacionais, e produzindo apenas artefatos realmente necessários durante o processo. E neste cenário surge uma divergência entre a indispensável presença de uma fundamentação pedagógica que permeia todo o desenvolvimento e que define a forma de interação do SE com o professor e aluno e a dificuldade de relacionar esta diversidade de perfis durante o processo de desenvolvimento.

Neste sentido, a proposta do YPEduc marca um avanço e o início de organização do processo de desenvolvimento de aplicações educacionais. Através dela é possível perceber pontos essenciais a este processo que aliadas às revisões de literatura contribuem para a evolução do processo e do conjunto de especificações propostas. É importante observar que o conjunto de critérios elencados tende a amadurecer com o

tempo e a prática do desenvolvimento deste processo será de grande auxílio para a evolução e utilidade da proposta.

Pretende-se, portanto, que este trabalho sirva de contribuição para reflexões que possam tirar proveito dos conhecimentos e dos autores aqui referenciados. E, por fim, que o mesmo possa criar lacunas e suscitar questionamentos que proporcionem incursões no sentido de se chegar entendimento inovador sobre o uso das tecnologias digitais em benefício da educação. E nesse sentido, destaca-se como trabalho futuro a aplicação do YPEDUC no desenvolvimento de SE de diferentes objetivos para consolidar a validação do processo.

References

- ABREU, F. et al. Métodos, Técnicas e Ferramentas para o Desenvolvimento de Software Educacional: um mapeamento sistemático. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2012.
- ALMEIDA, A. et al. Indicadores para Avaliação de Software Educacional com base no guia GDSM (Goal Driven Software Measurement). In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2018. p. 21.
- BOYLE, T. et al.; An Agile method for developing learning objects. Proceedings of the 23rd annual ascilite conference: Who's learning? Whose technology? Sydney, Australia, 2006.
- BRAGA, J. C.; PIMENTEL, E.; DOTTA, S. Metodologia INTERA para o desenvolvimento de Objetos de Aprendizagem. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2013. p. 306.
- COOMANS, S.; LACERDA, G. S. (2015). "PETESE, a Pedagogical Ergonomic Tool for Educational Software Evaluation". *Procedia Manufacturing*, v. 3, [p. 5881-5888], 2015.
- DA SILVA, R. S. T. et al.; (2016) "Aplicação comparativa de diferentes abordagens de avaliação para o software Duolingo - A complexidade de escolher uma abordagem adequada". *Revista Tecnologias na Educação*, V. 8, N. 16
- DE LIMA, M. M. et al. Uma Revisão Sistemática da Literatura dos Processos de Desenvolvimento de Software Educativo. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2012.
- FERREIRA, M. A. D.; RODRIGUES, A. N. (2015). "Interfaces Educativas: Implicações de design e processos cognitivos do jogo Nicetown", In: Congresso brasileiro de informática na educação - CBIE, 2015, Maceió - AL. Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015), 2015. p. 907-916.
- GARCIA et al.; easYProcess: Um Processo de Desenvolvimento para Uso no Ambiente Acadêmico. In: XXII WEI - Workshop de Educação em Computação, XXIV Congresso da Sociedade Brasileira de Computação, 2004, Salvador. SBC, 2004.

- GOMES, A. S.; WANDERLEY, E. G.; Elicitando requisitos em projetos de Software Educativo. In: WIE 2003 Workshop Brasileiro de Informática Educativa, 2003, Campinas. Ciência, Tecnologia e Atalhos para o futuro - Anais do XXIII Congresso da Sociedade Brasileira de Computação. Campinas : SBC, 2003. v. V. p. 227-238.
- JUNIOR, O.; AGUIAR, Y. P. C. Taxonomia de Critérios para Avaliação de Software Educativo-TaCASE. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2018. p. 298.
- LACERDA, R. A.; Proposta de um modelo para análise de requisitos de software educativo; 2007; Universidade de Brasília, UNB, Brasil. (Dissertação).
- MARÇAL et al. “Da Elicitação de Requisitos ao Desenvolvimento de Aplicações de Mobile Learning em Matemática”. In: Anais do XXI Simpósio Brasileiro de Informática na Educação, 2010.
- NETO, E. I. Scrumming - Ferramenta Educacional para Ensino de Práticas de SCRUM. 2008. Trabalho de Conclusão de Curso. (Graduação em Bacharelado em Sistemas de Informação) - Pontifícia Universidade Católica do Rio Grande do Sul.
- NETO, J. C.; REINEHR, S.; MALUCELLI, A. Processo de Desenvolvimento de Software: uma Análise Exploratória com Profissionais que Desenvolvem Jogos Eletrônicos Educacionais. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2015. p. 289.
- PERRY, G. T; Proposta de uma Metodologia Participativa para o Desenvolvimento de Software Educacional; 2006; Universidade Federal do Rio Grande do Sul, UFRGS, Brasil. (Dissertação).
- PET/UFCG. easYProcess: Um processo de desenvolvimento de software. Apostila. 2007.
- RODRIGUES, L. V.; DE FREITAS, S.; MENDES, F. Um estudo sobre o perfil das equipes de desenvolvimento de softwares educacionais. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2016. p. 159
- SOARES, M. S.; Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software. RESI. Revista Eletrônica de Sistemas de Informação, v. 3, p. 1-8, 2004.