

PROPOSTA E AVALIAÇÃO DE UMA ABORDAGEM LÚDICA PARA O ENSINO DE HISTÓRIAS DE USUÁRIO E SCRUM

RESUMO

O processo de aprendizagem está aprimorando constantemente seus métodos de ensino para que se obtenha uma melhoria na educação. Ensinar conceitos relacionados ao levantamento de requisitos, definição de escopo e gerenciamento de projetos demanda alinhamento entre teoria e prática. Esse trabalho apresenta e avalia uma proposta de abordagem para ensinar conceitos de Histórias de Usuários nos moldes do desenvolvimento ágil *Scrum*. Baseado na metodologia ADDIE se deu o desenvolvimento da atividade que foi validada junto a alunos de graduação e pós-graduação da Universidade X. Os resultados encontrados demonstram que a atividade atingiu seus objetivos propostos mesmo entre alunos que já conheciam a técnica e/ou possuíam experiência profissional em métodos ágeis. Uma contribuição importante deste trabalho é disponibilizar os materiais e roteiros para outros professores para que a atividade possa ser facilmente replicada em outras instituições e conseqüentemente constantemente evoluída.

Palavras chave: Histórias de Usuário; Ensino de Métodos Ágeis; *Scrum*.

PROPOSAL AND EVALUATION APPROACH PLAYFUL FOR STORIES OF EDUCATION USER AND SCRUM

ABSTRACT

The learning process is continuously evolving its teaching methods in order to obtain improvements in education. Teaching concepts related to requirements elicitation, scope definition and project management demands alignment between theory and practice. This work presents and assesses a new approach to teaching User Stories concepts following the agile Scrum development method. Based on the ADDIE methodology, the activity was developed and then executed with graduate and undergraduate students at Universidade X. Results demonstrated that the activity proposed achieved its goals even among students that already knew the technique or had professional experience with agile methodologies. An important contribution of this work is to make all materials and scripts available to other teachers so that the activity can be easily replicated in other universities and consequently continuously evolved.

Keywords: User Stories; Teaching Agile Methods; Scrum.

Lorena Adrian Cardoso Carvalho¹
Marcelo Werneck Barbosa²
Vinícius Bernardo Silva³

¹ Pontifícia Universidade Católica de Minas Gerais - PUC/MG. Brasil. E-mail: lorena.adrian@sga.pucminas.br

² Mestre em Ciências da Computação pela Universidade Federal de Minas Gerais - UFMG. Professor da Pontifícia Universidade Católica de Minas Gerais - PUC/MG. Brasil. E-mail: mwerneck@pucminas.br

³ Graduando em Ciências da Computação pela Pontifícia Universidade Católica de Minas Gerais - PUC/MG. Brasil. E-mail: vinicius.bernadosilva@gmail.com

1 INTRODUÇÃO

O cenário vigente no que tange o ensino de conceitos da Engenharia de Software, dentre eles o levantamento de requisitos e a gerência de projetos, demonstra que novas estratégias de ensino vêm sendo desenvolvidas para que a geração atual e as próximas possam experimentar uma abordagem que agrega a teoria e a prática. Expor conceitos sem sua efetiva demonstração prática não é tarefa fácil para professores que lecionam disciplinas dessa área. Mesmo fazendo uso de recursos multimídia, a perda de atenção dos alunos é notória pelos professores no decorrer de sua aula. Conforme Wangenheim (2012), apenas os 15 minutos iniciais de uma aula expositiva são realmente aproveitados pela maioria dos alunos.

Tornar a aula mais atrativa e envolvente é um desafio para professores, principalmente no contexto atual da geração Y, uma geração acostumada com a interatividade. Falkembach (2006) afirma que todo tipo de atividade lúdica agrada, entretém, retém a atenção dos envolvidos, estimula e ensina com maior eficiência, uma vez que atinge mais de um sentido dos participantes ao mesmo tempo sem tornar a atividade cansativa e desinteressante. Quando essa geração encontra-se em cursos de Computação, cujo perfil dos alunos contribui muito para essa demanda por maior interatividade e estímulo aos seus diversos sentidos, a necessidade por explorar diferentes estratégias de ensino aumenta.

É nesse contexto que trabalhos de pesquisa na área da Engenharia de Software vêm sendo desenvolvidos por professores e alunos na forma de elaboração de atividades lúdicas, cujo objetivo principal é criar um ambiente mais atrativo para o ensino e aplicação de conceitos abordados em sala de aula como os trabalhos Isotton (2008), Feitosa e Campos (2010), Silva e Barbosa (2011) e Wangenheim (2012). Silva e Barbosa (2011) realizaram o desenvolvimento e avaliação de um jogo cujo objetivo era aplicar o conhecimento sobre elicitación de requisitos. A avaliação do jogo pelos alunos mostrou que 69% dos envolvidos consideraram como importante a utilização do jogo para aplicação das técnicas explicadas em aula expositiva.

O processo de aprendizagem está aprimorando constantemente seus métodos de ensino para que se obtenha uma melhoria na educação. O lúdico é um desses métodos que está sendo utilizado na prática pedagógica, colaborando para o aprendizado do aluno e possibilitando ao educador o preparo de aulas dinâmicas fazendo com que o aluno interaja mais em sala de aula (QUEIROZ, 2014). Com o método lúdico cresce a vontade de aprender, seu interesse ao conteúdo

aumenta e dessa maneira ele realmente aprende o que foi proposto a ser ensinado, estimulando-o a ser pensador e questionador. Convém ressaltar que o docente deve ter cuidado ao desenvolver uma atividade trabalhando o lúdico, por ser uma atividade dinâmica. O professor fica na condição de estimulador, condutor e avaliador da realização da atividade, no entanto, o educador é o elo entre o lúdico e os alunos (LISBOA, 2013). O lúdico deixou de se referir apenas ao sentido de jogo, fazendo parte da atividade humana e com características espontânea, funcional e satisfatória, não importando somente o resultado, mas a ação e o que se é vivenciado com a atividade (ALMEIDA, 2009). O conhecimento através da ludicidade pode auxiliar a obter um melhor desempenho na aprendizagem. São muitas as vantagens de se educar ludicamente e, dentre elas, podemos citar: a melhoria da capacidade da interatividade, a potencialização da sua capacidade de desenvolver o raciocínio, bem como, da sua capacidade de relacionar-se com seus grupos de colegas (SANTOS, 2011).

Como vários dos trabalhos estudados relatam o uso e avaliação de jogos digitais para ensino de práticas ágeis, entende-se que o diferencial deste trabalho está em apresentar uma proposta de atividade com características lúdicas sem a necessidade de usar recursos computacionais. A atividade é toda baseada no uso de recursos facilmente disponíveis a qualquer professor. O lúdico nesta proposta está, não no sentido próprio de uma brincadeira, mas sim na proposta de uma atividade que seja dinâmica, interativa e que promova a interação entre membros de um mesmo grupo e ainda entre grupos de alunos. Outra grande preocupação deste trabalho é disponibilizar os materiais e roteiros para os professores para que a atividade possa ser facilmente replicada em outras instituições. Assim, esse trabalho tem por objetivo propor e avaliar um mecanismo lúdico de ensino de Histórias de Usuário e *Scrum*. Como contribuição deste trabalho, todo o material proposto e avaliado está disponibilizado como material de apoio para professores de disciplinas do eixo de Engenharia de Software que tratem destes temas.

Este trabalho está dividido da seguinte forma. A Seção 2 estabelece o referencial teórico do trabalho com destaque para os trabalhos relacionados enquanto a Seção 3 apresenta os procedimentos metodológicos utilizados. Na Seção 4 a proposta de atividade lúdica e seus elementos são apresentados. A Seção 5 descreve a análise dos dados obtidos com a realização da atividade. A Seção 6 apresenta os resultados finais enquanto a Seção 7 discorre sobre as conclusões deste trabalho.

2 REFERENCIAL TEÓRICO

2 MÉTODOS ÁGEIS

Métodos ágeis baseiam-se em uma abordagem incremental para a especificação, o desenvolvimento e a entrega do software. Eles são mais adequados ao desenvolvimento de aplicativos nos quais os requisitos do sistema mudam rapidamente durante o processo de desenvolvimento. Destinam-se a entregar o software rapidamente aos clientes, em funcionamento, e estes podem, em seguida, propor alterações e novos requisitos a serem incluídos nas iterações posteriores (SOMMERVILLE, 2011).

Os métodos ágeis possuem o compromisso de atuar de forma rápida e adaptativa às demandas do cliente, reduzindo burocracias e formalidades para melhorar o desempenho final do processo de desenvolvimento de software. Abrantes e Travassos (2007) afirmam que o desenvolvimento ágil não significa simplesmente desenvolvimento rápido de aplicações, mas principalmente capacidade de adaptação com rapidez e flexibilidade a mudanças nos processos, nos produtos e no ambiente.

Vários métodos ágeis têm sido empregados e estudados. O *Extreme Programming* (XP) é baseado em práticas e valores, tais como integração contínua, programação em pares, refatoração de código e programação baseada em testes. A estrutura do processo XP se baseia nos princípios ágeis, realçando o ambiente e os recursos disponíveis na organização e buscando quebrar problemas grandes em um conjunto de vários sub-problemas, com o intuito de aumentar a qualidade das soluções (SANTO et al., 2009).

A família *Crystal* possui uma abordagem voltada à gestão de pessoas. Seus princípios são personalizados para cada formato de projeto tomando por base sua complexidade. Na família *Crystal*, as entregas são frequentes, reduzindo a necessidade de produtos intermediários e a equipe possui bastante proximidade e integração, porém, os

membros têm especialidades distintas. Os projetos têm ciclos de desenvolvimento incremental com liberação de versões de um a quatro meses. Após o término das iterações é comum fazer uma reflexão sobre possíveis melhorias (PESSOA FILHO et al., 2009).

Dos métodos ágeis conhecidos e apresentados, o *Scrum* objeto de estudo deste trabalho, será abordado detalhadamente na próxima seção. A escolha pelo uso do método *Scrum* na proposta de atividade deste trabalho se dá pelo fato de que ele é um dos métodos mais usados, conforme afirma Sommerville (2011).

2.2 SCRUM

(Schwaber e Sutherland, 2011) definem o *Scrum* como um framework estrutural sobre o qual se pode utilizar dos vários processos ou técnicas para se desenvolver produtos complexos. Não é somente um processo ou técnica, mas a relação entre papéis, eventos, artefatos e regras do *Scrum* que viabilizarão um gerenciamento de projetos adaptável, iterativo e incremental. Esse modelo de gerenciamento de projetos advém das teorias empíricas de controle de processo e, portanto, se apoia nos pilares transparência, inspeção e adaptação.

O processo de gerenciamento no *Scrum* baseia-se nas unidades de trabalho denominadas *Sprints*. Schwaber e Sutherland (2011) definem *Sprint* como um *container* para outros eventos e possuem essa estrutura para viabilizar transparência e inspeção criteriosa. Pressman (2006) coloca ainda que as *Sprints* destinam-se para o desenvolvimento do(s) requisito(s) definido na Lista de Pendências (*Product Backlog*) e que será entregue no prazo estipulado (tipicamente até 30 dias). No decorrer da *Sprint*, esse(s) requisito(s) fica(m) congelado(s) e não sofre(m) alteração até a conclusão da *Sprint*. Assim, mesmo que o *Scrum* aceite alterações, durante o tempo da *Sprint*, o ambiente para desenvolvimento é estável. O fluxo do *Scrum* está detalhado na Figura 1.

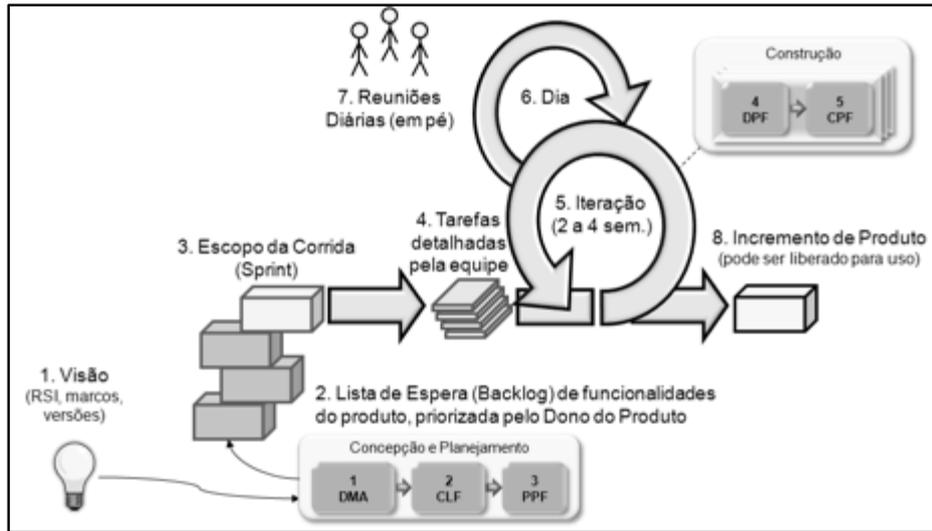


Figura 1 – Fluxo do Scrum.
Fonte: Jenny (2012).

O *Product Owner* define uma lista de funcionalidades a serem desenvolvidas no *Product Backlog*. Após essa definição, o *Scrum Master* estabelece quais dessas serão desenvolvidas no

decorrer da *Sprint*. Os papéis no *Scrum* citados até o momento compõem o *Time Scrum* e são detalhados na Figura 2.

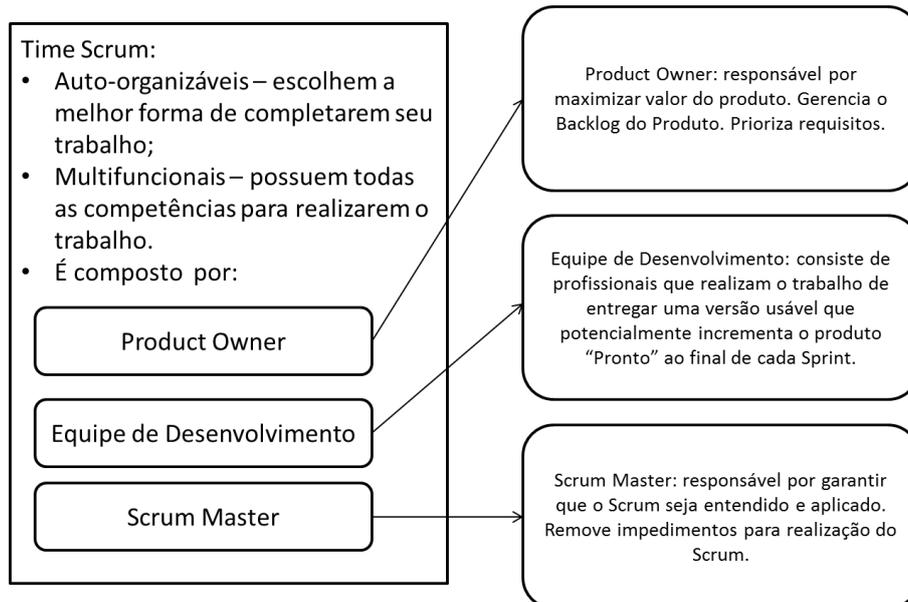


Figura 2- Papéis do Scrum

Fonte: Elaborada pela autora com textos extraídos de Schwaber e Sutherland (2011)

2.3 HISTÓRIAS DE USUÁRIOS

História de Usuário é um método para levantamento de requisitos que descreve de forma funcional os requisitos para o cliente ou comprador do projeto (Cohn, 2004). Três aspectos devem ser observados durante o processo de levantamento de requisitos:

- **Cartão:** meio de registro do requisito, normalmente em texto curto e como forma de lembrete quando na fase de desenvolvimento do software;
- **Conversas:** comunicação existente entre os envolvidos para melhor detalhar a história;

- **Critérios de aceitação:** pormenores a serem validados por meio de testes que servirão para validar e dar por concluída uma história.

Apesar de o cartão conter as funcionalidades, Cohn (2004) ressalta que este não é o mais importante. Por meio das conversas serão tratados todos os detalhes de cada funcionalidade. Uma história, ainda que sintetizada no cartão, que produz histórias ainda menores, é chamada de épico. Assim, um épico contém vários detalhes a serem descritos em cartões distintos.

Algumas características devem ser observadas ao escrever as histórias de usuário. O acrônimo inglês INVEST (*Independent, Negotiable, Valuable to users or customers, Estimatable, Small and Testable*), sugerido por Bill Wake e citado por Cohn (2004) contém os seis atributos indispensáveis para uma história bem escrita. São eles:

- **Independente:** uma história deve poder ser desenvolvida, testada e até entregue isoladamente. A conclusão de uma história deixa o produto em um estado em que ele pode ser entregue;
- **Negociável:** as histórias representam oportunidades de discussão dos requisitos. Deve haver possibilidade de *trade-offs* entre funcionalidade e datas de entrega;
- **de Valor:** proporcionar valor para o cliente é a essência dos métodos ágeis. Trata-se de uma mudança de uma abordagem de decomposição hierárquica funcional para uma abordagem vertical;
- **Estimável:** a equipe deve ser capaz de prover uma estimativa de sua complexidade e quantidade de trabalho necessário para terminar a história. Se equipe não consegue estimar, a história deve ser quebrada em histórias menores;
- **São de tamanho pequeno:** As histórias devem ser completadas em uma iteração. Histórias pequenas provêm maior agilidade;
- **Testável:** a história deve ser testável, pois se a equipe sabe como testar uma história, eles sabem como codificá-la.

Os critérios de aceitação nos testes das histórias servirão à equipe de desenvolvimento como lembrança sobre o que testar na história. Eles auxiliam a equipe no entendimento da história, porém não poderão conter tantos detalhes que dispense a comunicação entre equipe e clientes.

2.4 TRABALHOS RELACIONADOS

Vários trabalhos vêm sendo desenvolvidos sobre o uso de jogos promovendo a ludicidade como ferramenta de auxílio para o ensino de técnicas da Engenharia de Software. Jogos têm sido utilizados para ajudar no ensino de diversas áreas do conhecimento e muitas vezes despertam maior interesse por parte do aluno. Existe uma grande associação entre jogos e aprendizado (PRIKLADNICK E WANGENHEIM, 2008). Nesta seção, serão descritos alguns destes trabalhos e suas ferramentas.

Planager (KIELING E PRIKLADNICK, 2007) tem por objetivo explorar os conceitos relativos às atividades de gerência de software apoiados no PMBOK. Foi desenvolvido explorando os conceitos através da intuição e visualização. Além do aprendizado, a motivação do jogador é adquirir melhor pontuação para estar mais bem posicionado no *ranking*. Por ser um jogo digital, requer infraestrutura como um computador para utilizá-lo.

O *Scrumming* (ISOTTON, 2008) simula parte das práticas do *Scrum*. Trata-se de um jogo mais voltado para o entendimento das *Sprint*, uso dos gráficos de *burndown* e gerenciamento do *backlog*.

Em relação ao uso, num primeiro momento o jogo foi concebido pensando em ser utilizado por profissionais da indústria ou alunos de graduação. Entende-se que, por apresentar os conceitos de forma genérica, é possível utilizá-la com foco na gerência de projetos como um todo e não apenas em projetos de software.

No *SimulES-W* (MONSALVE, 2010), o aluno tem a condição de vivenciar problemas dificilmente abordados em aulas expositivas no papel de gerente de projeto e com isso desenvolver conceitos de computação e engenharia de software. O jogo, acessado pela Web, promete envolver seus jogadores motivando-os a construir e entregar o produto solicitado no menor tempo possível e atendendo aos requisitos de qualidade propostos. A competição torna-se maior quando os participantes, além de trabalhar em seu próprio produto, tem a condição de dificultar seus adversários com a criação de problemas. É nesse contexto que o ambiente desenvolvido se aproxima das condições reais de gerência de projetos e os alunos são desafiados a solucionar os problemas para prosseguirem no jogo.

Em *AprendES* (FEITOSA E CAMPOS, 2010), o jogo, que não é eletrônico e destaca nessa versão a vantagem de poder ser aplicado sem grandes exigências de infraestrutura, é uma releitura do jogo *SimulES*, versão antiga e de tabuleiro de *SimulES-W*, esse último diferenciando-se por ser jogado pela Web. Os jogadores são motivados a construir os artefatos definidos pelo projeto e será ganhador quem os fizer no tempo e orçamento pré-

definidos e cuja qualidade esteja nos limites aceitáveis.

Petri e Marcon Junior (2014) realizaram um experimento aplicando o *Agile Ball Point Game* em uma amostra de 31 alunos formando uma grande equipe. Utilizando o *Agile Ball Point Game* como um jogo educativo permitiu aos alunos vivenciar na prática os conceitos previamente teorizados em sala de aula. Em uma análise inicial dos resultados do experimento, obtidos através de um questionário, fatores como a necessidade de replanejamento, reorganização, comunicação, readaptação e a capacidade de liderança foram os itens identificados como essenciais para alcançar os objetivos do jogo.

Apesar de ser possível encontrar diversas ferramentas gratuitas para o gerenciamento de projetos *Scrum*, tais como *Pango*, *Scrumy*, *Mingle*, *ScrumWorks*, *IceScrum*, *ScrumHalf* (DUARTE, 2014); Borges, Schmitt e Nakle (2014), desenvolveram o *eduScrum*, uma metodologia desenvolvida em uma ferramenta que implementa uma solução virtual para o Kanban. A metodologia *eduScrum* foi utilizada em 3 turmas do curso Superior de Sistemas para Internet, do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS). Por meio de seus experimentos, foi possível comprovar que os princípios da metodologia *Scrum* são aplicáveis em um ambiente de aprendizagem.

Barros e Gestal (2014) propõem o uso de uma ferramenta que auxiliou professores e alunos no ensino e aprendizado do *Scrum* baseado na exposição a situações e problemas simulando um contexto real. O modelo proposto possibilita aos alunos discutir sobre cada decisão e estimativa tomada, revisão de decisões, identificar reações positivas e negativas de clientes e responder a eventos que ocorrem no decorrer de uma *Sprint*.

(Jaqueira, Coelho e Lucena, 2012) apresentaram um relato de experiência da aplicação de um projeto prático multidisciplinar envolvendo as disciplinas Engenharia de Software e Banco de Dados com alunos da graduação. O trabalho foi realizado usando práticas ágeis como a divisão do projeto em 3 *sprints*.

Wangenheim, Kochanski e Savi (2009) realizaram uma revisão sistemática para identificar jogos relacionados ao ensino e aprendizagem de Engenharia de Software no Brasil. O trabalho concluiu que ainda existem poucas publicações sobre jogos educacionais para ensino de Engenharia de Software no Brasil refletindo a necessidade de aperfeiçoamento deste método de ensino na área. Dentre as áreas da Engenharia de Software, a mais abordada pelos jogos é a de Gerenciamento de Projetos.

Como se pode perceber pelos trabalhos descritos, a grande maioria é voltada para o uso e

avaliação de jogos digitais. O diferencial deste trabalho é apresentar uma proposta de atividade com características lúdicas sem a necessidade de usar recursos computacionais. A atividade é toda baseada no uso de recursos facilmente disponíveis a qualquer professor. Outra grande preocupação deste trabalho é disponibilizar os materiais e roteiros para os professores para que a atividade possa ser facilmente replicada em outras instituições.

3 PROCEDIMENTOS METODOLÓGICOS

Este trabalho pode ser caracterizado como uma pesquisa qualitativa que envolveu como participantes alunos de graduação e pós-graduação da Universidade X. Como instrumentos de coleta de dados, foram elaborados e aplicados questionários de satisfação em relação à atividade proposta aos participantes. A elaboração da atividade foi toda baseada na metodologia ADDIE, utilizada principalmente no projeto e desenvolvimento de objetos de aprendizagem.

A metodologia ADDIE (*Analysis, Design, Development, Implementation, Evaluation*) utilizada no *design* instrucional pode ter suas fases descritas assim: a análise corresponde ao processo de definição dos conteúdos (o que deve ser ensinado); o projeto é o processo de determinação de como estes conteúdos devem ser ensinados de acordo com os objetivos educacionais (seqüenciamento, mídias e metodologia de ensino); o desenvolvimento é o processo de autoria e produção dos materiais educacionais; a implementação corresponde ao processo de entrega destes materiais para uso; a avaliação permeia todos os processos e guia as atividades pós-desenvolvimento dos materiais (Molenda, 2014). Por ser uma metodologia sólida neste contexto, este trabalho foi realizado pautado pela metodologia ADDIE e contemplou, de maneira sucinta, as seguintes atividades:

1. Pesquisa sobre trabalhos relacionados ao desenvolvimento de atividades lúdicas e jogos digitais em Engenharia de Software, particularmente voltados para o ensino ou aplicação de métodos ágeis;
2. Estudo, identificação e definição dos conceitos e técnicas que deverão ser desenvolvidos dentro da atividade lúdica a ser proposta.
3. Identificação de uma metodologia para desenvolvimento de modelos lúdicos de ensino sendo a escolhida o processo de design instrucional ADDIE;
4. Elaboração da atividade de ensino com base no processo acima citado;

5. Aplicação e avaliação da atividade desenvolvida com alunos da graduação e pós-graduação da Universidade X.

A seguir, são descritas em maior detalhe as etapas da metodologia ADDIE e como ela foi utilizada neste trabalho.

3.1 O USO DA METODOLOGIA ADDIE PARA A ELABORAÇÃO DA ATIVIDADE

3.1.1 Análise

A primeira fase do modelo ADDIE, a Análise, comporta a motivação principal desse trabalho. Por meio da observação, professores de disciplinas do eixo de Engenharia de Software da Pontifícia Universidade Católica de Minas Gerais identificaram a necessidade de trabalharem também com modelos lúdicos para práticas dos conteúdos ministrados em sala de aula.

3.1.2 Desenho

Na fase de Desenho do modelo ADDIE, definem-se os objetivos e todo o planejamento das atividades para atenderem as necessidades identificadas na fase anterior, a Análise. Foi nessa fase que todo o material para a atividade foi desenvolvido.

A princípio foram listados os aspectos considerados essenciais sobre Histórias de Usuários que deveriam ser praticados pelos alunos durante a atividade. São eles:

- Identificação e registro das histórias em cartões de história;
- Realização de conversas;
- Definição e registro dos critérios de aceitação;
- Realização das estimativas para as histórias seguindo o método *Planning Poker*;
- Definição de um *Sprint Backlog* com base no entendimento e na priorização das histórias;

3.1.3 Desenvolvimento

Na fase de Desenvolvimento, foi produzido o material utilizado na execução da atividade, material de apoio tal como apresentações, ferramentas de avaliação, guias e procedimentos.

Foram desenvolvidos os *templates* dos cartões de história, do *Product Backlog*, *Sprint Backlog* e *critérios de aceitação*. Foi definido o cenário de proposta para elicitación dos requisitos, procedimentos para realização da prática, regras, o método de estimativas para as histórias e material complementar para exposição do conteúdo tema da

atividade. Esse material será apresentado e descrito na Seção 4.

3.1.4 Implementação

A fase de Implementação é a prática da atividade propriamente dita. A atividade foi aplicada em dois momentos distintos. No primeiro momento, para alunos da graduação da Universidade X, em um curso extracurricular com quatro horas de duração. No segundo momento para alunos da pós-graduação da Universidade X dentro da disciplina de Desenvolvimento Ágil de Software com duração total de 3,5 horas. Para os alunos da graduação foi estabelecido como pré-requisito já ter cursado a disciplina de Engenharia de Requisitos e, portanto, os alunos deviam possuir conhecimentos prévios sobre o assunto. Não era necessário que já tivessem qualquer experiência profissional sobre o conteúdo.

3.1.5 Avaliação

Na fase de Avaliação, devem-se analisar e comparar os dados levantados para chegar a resultados que indicam se os objetivos propostos foram atingidos. Para esse trabalho foi elaborado um questionário seguindo o modelo proposto por Savi (2011) que considera medir pela escala de Likert a opinião dos participantes em relação à prática da atividade. Os detalhes sobre avaliação são descritos na Seção 5.

4 DESCRIÇÃO DA ATIVIDADE PROPOSTA

4.1 VISÃO GERAL

A atividade proposta pode ser definida como uma atividade lúdica complementar elaborada para alunos de disciplinas de Engenharia de Software relacionadas a temas como Engenharia de Requisitos de Software ou Gerenciamento de Projetos. A atividade foi planejada para ser executada em sala de aula sem a necessidade de infraestrutura tal como computadores e acesso à internet para uso.

Com a proposta de trabalhar a prática do registro em cartões de história das funcionalidades solicitadas por um cliente hipotético, o aluno é estimulado a exercitar também a comunicação oral estabelecida com usuários por meio das conversas, que na atividade são realizadas com o professor que desempenha o papel de usuário (*Product Owner*). O aluno também passa a conhecer e tem a oportunidade de praticar, ainda que em um formato adaptado para a atividade, diversos estágios do *Scrum*.

A atividade é desenvolvida por meio dos seguintes passos:

1. Os participantes são divididos em grupos (preferencialmente em trios);
2. O material é distribuído e apresentado aos participantes;
3. O contexto e requisitos do sistema, procedimentos e regras da atividade são apresentados aos presentes e, a partir de então, se dá início à atividade propriamente dita, que se encerra com um questionário para avaliação da prática.

Como contexto para o desenvolvimento do sistema, os alunos são apresentados às necessidades de uma biblioteca de uma escola pequena que precisa informatizar seus processos de pesquisa ao acervo, realização de empréstimo e devoluções. Neste contexto, os alunos ficam sabendo que a biblioteca não aplica penalidades aos alunos que atrasam as devoluções, mas mantém em seus arquivos uma restrição vinculada à matrícula do aluno que não devolver o livro até sua formatura e isso implica débitos para o aluno que deseja retirar seu diploma. Sabem ainda que atualmente existe um sistema de gestão acadêmica na escola contendo os dados cadastrais de todos os alunos e ex-alunos e o mesmo disponibiliza em arquivo XML os dados dos alunos. Em relação às prioridades, os alunos tomam conhecimento de que é urgente para a biblioteca realizar a pesquisa em seu acervo seja pelo nome do livro, por autor ou gênero literário. Quando algum usuário realizar a pesquisa, o sistema deverá exibir o nome do livro, nome do autor (ou autores) e quantidade de exemplares contidos na biblioteca. Até que o processo de empréstimo e devolução seja incorporado, não é necessário apresentar a disponibilidade dos itens do acervo. Os empréstimos e devoluções são processos que podem ser entregues

depois. Atribuir restrições a alunos com pendências é o processo de menor grau de prioridade para a biblioteca.

Como pode ser observado, as informações passadas aos alunos permitem que eles possam reconhecer as histórias principais e até mesmo decompor algumas em histórias menores; que eles tenham uma noção da prioridade de cada história de maneira que eles possam identificar aquelas que devem compor a primeira *Sprint*.

Cada grupo inicia a atividade registrando nos cartões de história (Figura 4) aquelas levantadas com base no cenário apresentado. Em seguida, os alunos fixam as histórias no *Product Backlog* representado na Figura 5. O *Product Backlog* é disponibilizado para os alunos como uma folha de papel *Kraft* (ocupando o lado esquerdo da folha). Considerando a prioridade já estabelecida no cenário e também por meio de conversas realizadas com o *Product Owner* (papel desempenhado pelo professor), cada história terá sua prioridade assinalada. Em seguida, as histórias de prioridade alta são transferidas para a *Sprint Backlog*, localizada na metade direita da folha de papel *Kraft* (Figura 5) e estimativas são realizadas usando o método *Planning Poker*. Terminadas estas atividades, os alunos então detalham as tarefas para a primeira história da *Sprint Backlog*, bem como os seus critérios de aceitação também registrados em cartão conforme representado na Figura 6. Por último é elaborado um produto, chamado de “produto acabado” sob a forma de protótipo de tela que refletirá a funcionalidade descrita na história cujas tarefas foram detalhadas e critérios de aceitação escritos. Esse produto precisa evidenciar ao menos um dos critérios de aceitação da história. Essa sequência de tarefas desempenhadas pelos alunos está representada na Figura 3.

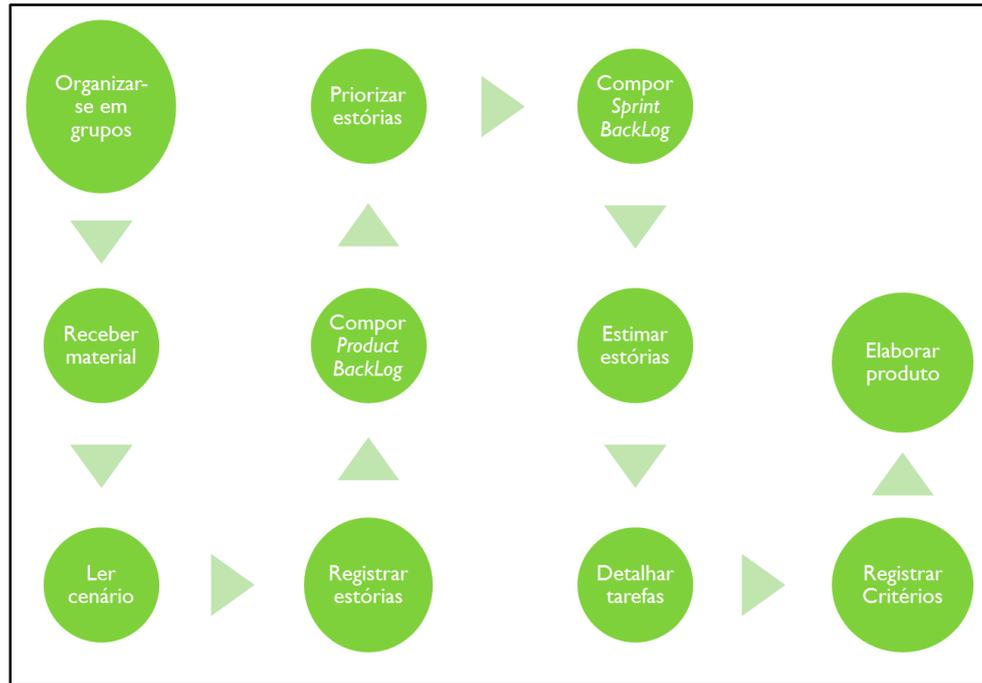


Figura 3- Etapas da atividade na perspectiva dos alunos
 Fonte: Elaborado pelos autores

O método de estimativas *Planning Poker* foi escolhido porque sua compreensão e execução são rápidas e fáceis, além de ele naturalmente envolver em sua execução os recursos já utilizados pela atividade. O material para realizar o *Planning Poker* desenvolvido consistiu em cartões numerados conforme a sequência de Fibonacci de 1 a 13. Todo material necessário para realizar a atividade encontra-se disponível para *download*⁴.

4.2 ARTEFATOS DA ATIVIDADE

Os artefatos da atividade usados no decorrer de sua realização estão representados nas figuras a seguir. Na Figura 4 é apresentado o *template* dos cartões de história. Foi disponibilizado um texto no verso do cartão de história que reforça as características que uma história precisa apresentar. Na Figura 5, são exibidos o *Product Backlog* (esquerda) e *Sprint Backlog* (direita) que cada grupo monta de acordo com as histórias escritas e prioridade das mesmas. Os critérios de aceitação são

registrados em cartões conforme *template* da Figura 6. As tarefas listadas e o produto acabado são escritos em folhas em branco fornecidas junto com os demais materiais.

Além desses, outros materiais foram elaborados como material de apoio. Para os alunos, os materiais disponibilizados são:

- Cartões de história já preenchidos servindo de exemplo para a forma correta da escrita das histórias. Estes cartões de história apresentam requisitos de outro sistema, mas no formato que o aluno deve seguir;
- O texto que descreve o contexto do sistema e os procedimentos para realizar a atividade;
- Um texto explicativo sobre como identificar e escrever os critérios de aceitação;
- Os cartões para realizar a estimativa com o *Planning Poker*;
- Regras e dicas que estão explicadas na próxima seção.

⁴ <https://db.tt/XWrRXXoW>

Como <papel>	ID.:
Eu posso/preciso/devo/quero <funcionalidade>	
FRENTE	
Parceiro valor para o negócio	
Prioridade: BAIXA MEDIA ALTA	Estimativa:

Quando elaboro uma estória, preciso considerar:

- **Independência:** procurar elaborar estórias independentes entre si para evitar confusão quando na priorização e planejamento das atividades de desenvolvimento.
- **Negociável:** a estória não precisa conter todo o detalhamento e com isso promover maior comunicação entre equipe de desenvolvimento e cliente.
- **Útil:** as estórias precisam agregar valor para o cliente e por isso é necessário que os desenvolvedores possam entender seu valor.
- **Estimável:** com a finalidade de possibilitar a estimativa geral do projeto, as estórias, ainda que não totalmente entendidas, precisam ser compreendidas no contexto geral por todos.
- **Pequena:** as estórias precisam ser sintetizadas e assim poderem ser desenvolvidas, mas sem perderem seu valor ao negócio.
- **Testável:** cada estória precisa ser passível de teste para que a equipe de desenvolvimento possa determinar seu término.

Figura 4- Representação do modelo do cartão de história
 Fonte: Elaborado pela autora com textos de Bill Wake e citado por (Cohn, 2004)

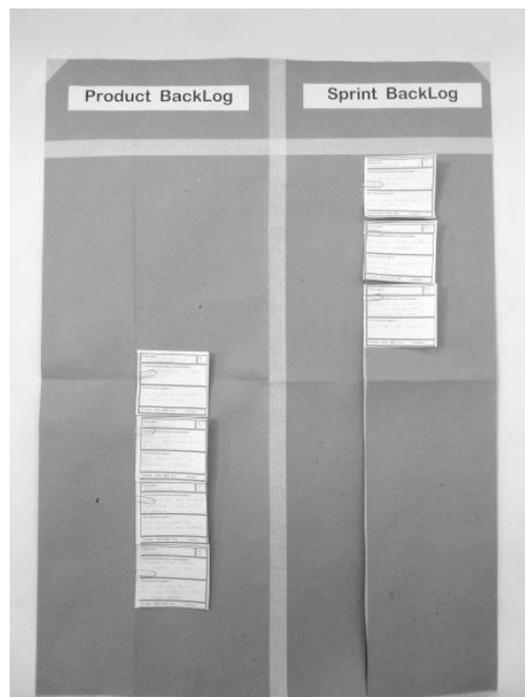


Figura 5- Representação do *Product BackLog* e *Sprint BackLog*
 Fonte: Elaborado pelos autores

4.3 REGRAS E DICAS

Para a atividade foram estabelecidas algumas regras principalmente no atendimento do

professor às dúvidas dos alunos, seja assumindo o papel de *Product Owner* ou *Scrum Master*. São elas:

Critérios de aceitação	ID.:

Figura 6- Representação do *template* para critérios de aceitação
 Fonte: Elaborado pelos autores

1. Sempre que julgar necessário esclarecer dúvidas em relação ao cenário e/ou histórias, o grupo deve solicitar uma conversa com o *Product Owner*. Cada conversa terá a duração máxima de 3 minutos para permitir que o *Product Owner* atenda também outros grupos;
2. Não há limites de número de conversas por grupo, mas existe a seguinte política de atendimento: O *Product Owner* primeiro deverá atender pelo menos uma vez cada grupo conforme ordem de chegada e solicitação dos grupos. Caso haja muitos pedidos de conversa e o tempo não seja suficiente, o *Product Owner* atenderá os grupos de tal forma que a quantidade de conversas com os grupos seja igual ou aproximada;
3. No verso do cartão de história são listadas as características que uma história bem elaborada deve apresentar. Os alunos são incentivados a utilizar o texto sempre que julgarem necessário;
4. Os alunos são informados que no material fornecido há cartões de história preenchidos para auxiliar na elaboração dos cartões de história da atividade;

5 AVALIAÇÃO DA ATIVIDADE

Conforme Savi (2011), avaliar o material educacional utilizado no processo de ensino aprendizagem é essencial, pois é necessário

identificar se o produto educacional conseguiu cumprir com o planejado para ele. A avaliação desenvolvida para a atividade baseou-se no modelo sugerido por Savi (2011) que considera os aspectos Motivação, Experiência de Usuário e Conhecimento analisados em frases que são classificadas de acordo com a escala de Likert variando de 1 a 5 sendo o valor 1 para a expressão “Não concordo totalmente”, 2 para “Não concordo parcialmente”, 3 para “Indiferente”, 4 para “Concordo parcialmente e 5 para “Concordo totalmente”. O Quadro 1 relaciona as frases usadas na avaliação classificadas de acordo com os aspectos acima citados.

Os 47 alunos participantes, sendo 7 da graduação e 40 da pós-graduação, também responderam perguntas relativas ao seu perfil. Nelas era possível saber se estes conheciam o conteúdo do tema Histórias de Usuário tendo vivenciado ou não a prática da técnica e se já possuíam experiência profissional com desenvolvimento ágil. Dos alunos da graduação, 3 declararam possuir apenas conhecimento enquanto 4 afirmaram ter conhecimento e experiência prática com histórias de usuário. Dos alunos da pós-graduação, 13 disseram até antes da disciplina não ter conhecimento sobre histórias de usuário, 16 afirmaram possui somente conhecimento enquanto os 11 restantes tinham tanto conhecimento quanto prática com esta técnica.

Todos os dados coletados foram analisados a partir da mediana das respostas dos alunos. Tal cálculo estatístico foi adotado, como justifica Wangenheim (2013) por considerar a tendência central das respostas da população observada.

MOTIVAÇÃO	Satisfação	Eu aprendi algumas coisas com a atividade que foram surpreendentes ou inesperadas.
	Confiança	As tarefas da atividade foram muito difíceis.
	Relevância	O conteúdo da atividade é relevante para meus interesses.
	Atenção	A proposta da atividade capturou minha atenção.
EXPERIÊNCIA DO USUÁRIO	Divertimento	Fiquei torcendo para que a atividade terminasse logo.
	Interação Social	A colaboração na atividade ajuda a aprendizagem.
	Habilidade	Senti que estava tendo progresso quanto ao conteúdo ensinado e por mim compreendido durante o desenrolar da atividade.
	Desafio	A atividade oferece novos desafios para mim num ritmo apropriado.
	Imersão	Eu não percebi o tempo passar enquanto desenvolvia a atividade.
CONHECIMENTO		Considero que a atividade contribuiu para que eu compreendesse melhor o tema por ela proposta.

Quadro 1 – Formulário para avaliação da atividade segundo modelo de (SAVI, 2011)

6 RESULTADOS E CONCLUSÕES

A Motivação pelo modelo ARCS, acrônimo de termos do inglês que em português são Atenção, Relevância, Confiança e Satisfação descrita por Savi (2011) foca na interação dos alunos com o ambiente de aprendizagem. Essa motivação determina o esforço empregado para realizar a atividade.

Com base no Gráfico 1, pode-se perceber que os alunos sentiram-se motivados a executar a atividade, identificando sua relevância e sem grandes dificuldades. Esse resultado não varia consideravelmente entre as turmas o que significa que mesmo para participantes em níveis mais avançados de conteúdo, a utilização da atividade lúdica é bem recebida.

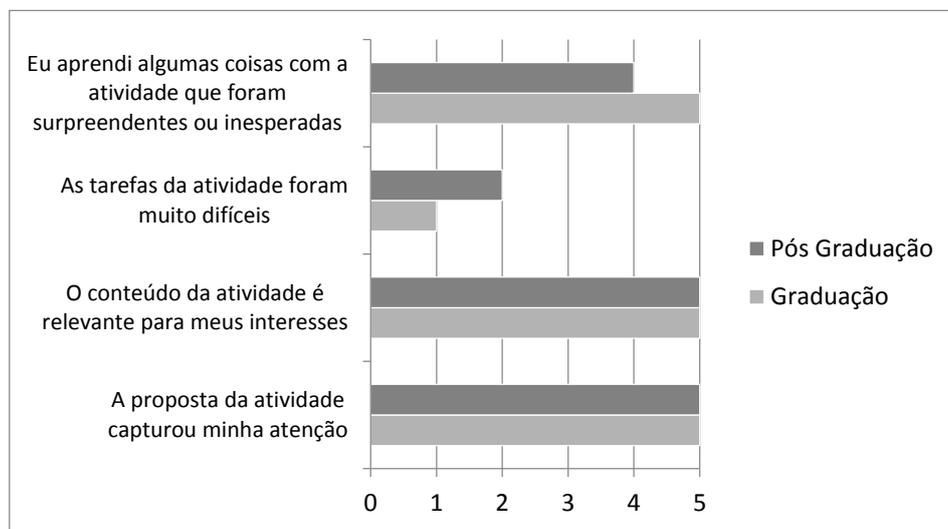


Gráfico 1 - Mediana da avaliação dos aspectos motivacionais.

Fonte: Elaborado pelos autores.

A Experiência de Usuário no modelo de avaliação procura identificar as emoções vividas pelo participante no decorrer da atividade e como essas emoções tornam-se determinantes no comportamento do participante. Nesse aspecto, a atividade conseguiu promover uma boa experiência de usuário, seja na imersão dos participantes na

prática, interação com outros colegas e sentimento de realização notando seu progresso na atividade. O grau de dificuldade para as habilidades dos alunos também é adequada como se vê na avaliação do item que se refere a desafio em ritmo apropriado, conforme pode ser observado no Gráfico 2

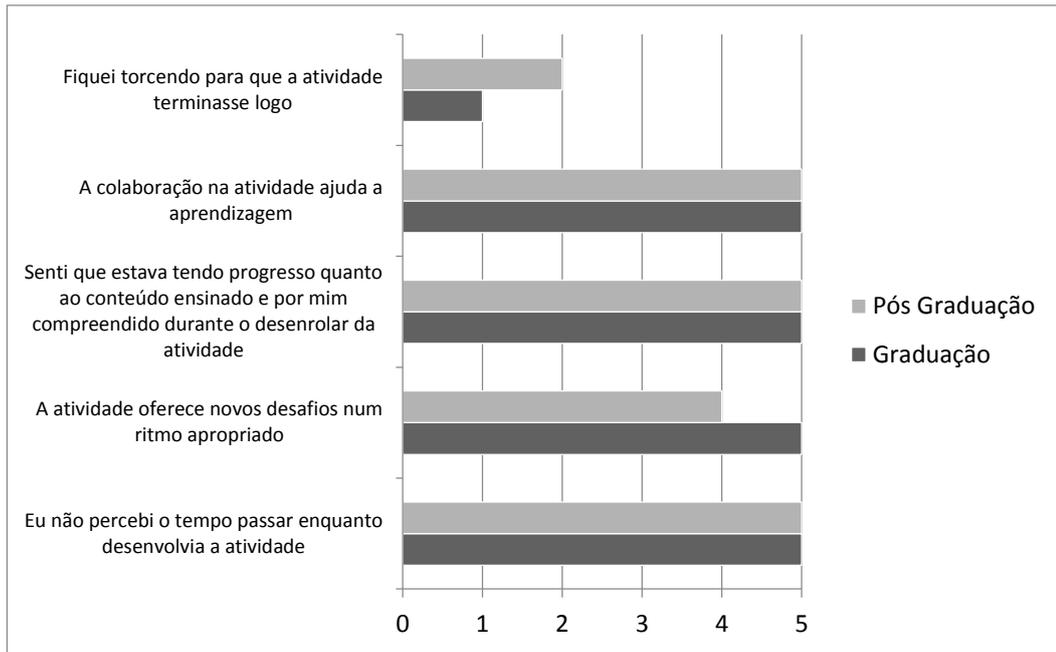


Gráfico 2 - Mediana da Avaliação da Experiência de Usuário na atividade.
Fonte: Elaborado pelos autores.

No modelo de avaliação proposto por Savi (2011), o elemento Conhecimento é definido dentro da Taxonomia de Bloom nos três primeiros níveis: conhecimento, compreensão e aplicação. Na

avaliação da atividade, conforme Gráfico 3, todos aqueles que participaram consideraram que a prática proporcionou melhor compreensão do tema apresentado.

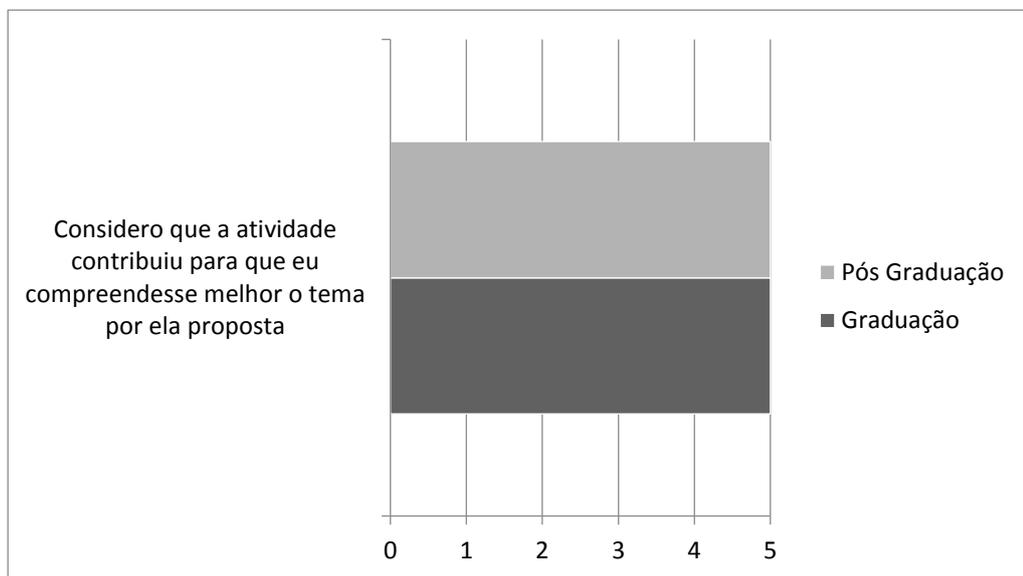


Gráfico 3 - Mediana da avaliação do aspecto Conhecimento da atividade.
Fonte: Elaborado pelos autores.

7 CONCLUSÃO

Esse trabalho apresentou a elaboração, aplicação e avaliação de uma atividade para prática do conhecimento sobre a técnica de Histórias de Usuário. Os resultados obtidos na avaliação são satisfatórios e indicam que a atividade conseguiu atingir os objetivos propostos. Durante a fase de implementação alguns participantes se mostraram interessados em aplicar a prática em seu ambiente de trabalho como proposta de treinamento. Esse interesse demonstra que a atividade não precisa ficar limitada ao contexto acadêmico podendo ser adaptada ao ambiente empresarial.

Dos alunos que participaram da atividade tanto na graduação como na pós-graduação, mais da metade declaram possuir conhecimento ou experiência com desenvolvimento ágil e ainda sim entre esses os resultados se mantem na mesma tendência de opinião identificada na seção de Avaliação, o que indica que a atividade aqui proposta já pode ser utilizada. Contudo, algumas lições aprendidas são sugeridas como observações a serem consideradas. No cenário a ser trabalhado, seja o proposto aqui ou algum outro definido pelo aplicador da atividade, é importante que para quem desempenhar o papel de *Product Owner*, o escopo esteja bem claro e assim nas conversas desenvolvidas na atividade não haja divergências de respostas entre os grupos. Também é importante ressaltar para os alunos que o procedimento definido para realização da atividade foi adaptado para ser desenvolvido em sala de aula e que, portanto alguns passos se diferenciam um pouco da prática adotada nas empresas, como por exemplo, a ordem que as estimativas e decomposição das histórias em tarefas são realizadas. Tais decisões foram tomadas para tornar a atividade passível de ser concluída no tempo estipulado bem como mais dinâmica.

Como trabalhos futuros, pretende-se elaborar propostas de outras atividades lúdicas que compreendam também outras atividades típicas do desenvolvimento ágil com *Scrum*, como por exemplo, testes e integração contínuos. Pretende-se ainda integrar e avaliar as atividades aqui propostas em disciplinas do curso de graduação em Sistemas de Informação desta universidade.

REFERÊNCIAS

- Abrantes, José F.; Travassos, Guilherme H. (2007). Caracterização de métodos ágeis de desenvolvimento de software. 1ª Sessão. In: WDRA, SBQS, Porto de Galinhas, Pernambuco.
- Almeida, A (2009). Ludicidade como instrumento pedagógico. Disponível em: <<http://www.cdof.com.br/recrea22.htm>>. Acesso em: 12 de dezembro de 2014.
- Barros, R. M. de; Gestal, P. R. E (2014). Proposta de Um Simulador para Auxiliar no Processo de Ensino do Scrum. Simpósio Brasileiro de Sistemas de Informação (SBSI). Universidade estadual de Londrina.
- Borges, K. S.; Schmitt, A. R.; Nakle, M. (2014). eduScrum: Projetos de Aprendizagem Colaborativa Baseados em Scrum. Revista Renote (Novas Tecnologias na Educação), V. 12, N. 1.
- Cohn, M. (2004). User Stories Applied: For Agile Software Development. Editora Addison-Wesley Professional, 304 p. (Coleção Addison-Wesley Signature Series).
- Duarte, R. Ferramentas para trabalhar com Scrum. Disponível em <http://ricardoduarte.net/posts/ferramentas-para-trabalhar-com-scrum.html/>. Acesso em 12 de dezembro de 2014.
- Feitosa, A. C.; Campos, G. M. M. (2010). AprendES: um jogo educacional para auxiliar o processo de ensino-aprendizagem da Engenharia de Software. Anais do Simpósio Brasileiro de Informática na Educação.
- Isotton, E. N. (2008). Scrumming – Ferramenta Educacional para Apoio ao Ensino de Práticas de SCRUM. Monografia de Trabalho de Conclusão de Curso de Graduação, Sistemas de Informação. In: FACIN, PUCRS. Porto Alegre.
- Jaqueira, A.; Coelho, R.; Lucena, M. Conduzindo Projetos Ágeis em um Contexto Multidisciplinar: Um Relato de Experiência. Fórum de Ensino de Engenharia de Software (FEES). Congresso Brasileiro de Software (CBSOFT), 2012.
- Jenny, Juliana (2012). Arquivo de Categoria: Engenharia de Software. Disponível em <<http://julianakolb.com/category/enharia-de-software/page/10/>>. Acessado em 4 de Nov de 2013.
- Falkembach, G.A.M. (2006) – O Lúdico e os Jogos Educacionais. In: CINTED - Centro Interdisciplinar de Novas Tecnologias na Educação, UFRGS. Disponível em

- <http://penta3.ufrgs.br/midiasedu/modulo13/etapa1/leituras/arquivos/Leitura_1.pdf>. Acessado em 05 de Nov de 2013.
- Kieling, E.; Prikladnicki, R. (2007). Ensino de Gerência de Projetos de Software com o Planager. Anais do Simpósio Brasileiro de Informática na Educação.
- Lisboa, M. (2013). A importância do lúdico na aprendizagem, com auxílio de jogos. Disponível em: <<http://brinquedoteca.net/?p=1818>> Acesso em: 12 de dezembro de 2014.
- Molenda, M. (2014). The ADDIE Model. Disponível em: <www.indiana.edu/~idt/shortpapers/documents/IDTf_Bic.pdf>. Acesso em 12 de dezembro de 2014.
- Monsalve, E.; Werneck, V.; Leite, J.C.S.P. (2010). SimulES-W: Um Jogo para o Ensino de Engenharia de Software. In: Fórum de Ensino de Engenharia de Software (FEES), Simpósio Brasileiro de Engenharia de Software (SBES), Bahia.
- Pessoa Filho, H. F. B.; Santana Júnior, C. A. Gusmão, C. M. (2009). Uma abordagem analítica das metodologias ágeis: Crystal, Scrum e XP. Workshop de Desenvolvimento Rápido de Aplicações (WDRA). Simpósio Brasileiro de Qualidade de Software (SBQS).
- Petri, G.; Marcon Júnior, R. P. (2014). Um Jogo Educacional para o Ensino de Metodologias Ágeis. Fórum de Ensino de Engenharia de Software (FEES). Congresso Brasileiro de Software (CBSOFT).
- Pressman, Roger S. (2006). Engenharia de software. São Paulo: McGraw-Hill, 720 p.
- Prikladnicki, R.; von Wangenheim, C. G. (2008). O Uso de Jogos Educacionais para o Ensino de Gerência de Projetos de Software. Fórum de Ensino de Engenharia de Software (FEES), Simpósio Brasileiro de Engenharia de Software (SBES).
- Queiroz, I. C. (2014). Gerenciamento de Projetos: Método lúdico de ensino. Monografia Universidade Estadual de Goiás, curso de Bacharel em Sistemas de Informação.
- Santo, R. E.; Santos, R.; Santos, P. S.; Werner, C. (2009). Uma Análise das Características de Extreme Programming Considerando o Desenvolvimento de Software Orientado a Aspectos. Workshop de Desenvolvimento Rápido de Aplicações (WDRA). Simpósio Brasileiro de Qualidade de Software (SBQS).
- Santos, J. O. (2011). O lúdico na Educação Infantil. Disponível em: <http://br.monografias.com/trabalhos3/ludico-educacao-infantil/ludico-educacaoinfantil.shtml>. Acesso em: 13 de dezembro de 2014.
- Savi, Rafael et al. (2011). Proposta de um Modelo de Avaliação de Jogos Educacionais. 236 f. Tese (Doutorado) - Universidade Federal de Santa Catarina (UFSC), Florianópolis.
- Silva, Monielle V.; Barbosa, Marcelo W. (2011). Desenvolvimento e avaliação de jogo de treinamento para ensino de técnicas de elicitação de requisitos de software. II Encontro Nacional de Informática e Educação (ENINED). Campus Cascavel, Paraná.
- Schwaber, K.; Sutherland, J. (2011). Um guia definitivo para o Scrum: As regras do jogo. Disponível em <<https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum%20Guide%20-%20Portuguese%20BR.pdf>>. Acessado em 30 de Set 2013.
- Sommerville, I. (2011). Engenharia de Software. 9ª edição, Editora Pearson.
- Wangenheim, C. G.; Kochanski, D.; Savi, R. (2009). Revisão Sistemática sobre Avaliação de Jogos Voltados para aprendizagem de Engenharia de Software no Brasil. Fórum de Ensino de Engenharia de Software (FEES), Simpósio Brasileiro de Engenharia de Software (SBES).
- Wangenheim, Christiane G. Carvalho, Osório P., Batistella, Paulo E. (2013). Ensinar Gerência de Equipes em Disciplinas de Gerência de Projetos de Software. Revista Brasileira de Informática na Educação, v. 21, n. 1.