

SPI City: Jogo Educacional para Apoiar o Ensino de Melhoria de Processo de Software

Trabalho Técnico

José Luiz Silveira¹, Marcello Thiry¹, Alessandra Zoucas²

¹Mestrado em Computação Aplicada – Universidade do Vale do Itajaí (UNIVALI) Rua Uruguai – 458 - Centro – CEP 88302-202 – Itajaí – SC – Brasil

²Doutorado em Engenharia e Gestão do Conhecimento – Universidade Federal de Santa Catarina (UFSC) – Campus Universitário Reitor João David Ferreira Lima – Trindade – CEP 88040-970 – Florianópolis – SC – Brasil

{joseluizsilveira2,marcello.thiry,alessandrazoucas}@gmail.com

Abstract. *Researches has shown that the main difficulty in the implementation of Software Process Improvement is related to the lack of knowledge about the basic techniques of Software Engineering and the lack of understanding of the software process reference models. This article presents a digital game developed to support the teaching of Software Process Improvement. The game simulates everyday situations of software development companies in which the expected results of the MPS Reference Model for Software (MR-MPS-SW) are explored. The article still presents the results of experiments performed with the game that demonstrate the usefulness of the game in promoting learning of Software Process Improvement.*

Resumo. *Pesquisas tem demonstrado que uma das principais dificuldades na implementação de Melhoria de Processo de Software estão relacionadas com a falta de conhecimento sobre as técnicas básicas da Engenharia de Software e a falta de entendimento dos modelos de referência de processo de software. Este artigo apresenta um jogo desenvolvido, em meio digital, para apoiar a capacitação de Melhoria de Processo de Software. O jogo simula situações do dia a dia de empresas de desenvolvimento de software em que os resultados esperados do Modelo de Referência MPS para Software (MR-MPS-SW) são explorados. O artigo ainda apresenta os resultados dos experimentos realizados com o jogo que permitiram comprovar que ele auxilia na aprendizagem de Melhoria de Processo de Software.*

1. Introdução

A implementação de Melhoria de Processo de Software é uma atividade fortemente dependente do conhecimento das pessoas envolvidas [Montoni et al 2007]. Os responsáveis pela implementação de melhoria nas organizações devem possuir um profundo conhecimento em Engenharia de Software e nos modelos de referência de processos de software, para poder identificar de que forma uma prática já estabelecida

na organização pode ser aproveitada ou como estabelecer novas práticas inovadoras para produzir melhores resultados nos processos da organização [Montoni et al 2007].

No entanto, pesquisas têm demonstrado que a principal dificuldade na implementação de Melhoria de Processo de Software está justamente relacionada com a competência da equipe da empresa onde está sendo implementada a melhoria [Rocha et al 2005], [Habib 2009] e [Rodrigues e Kirner 2010]. Destacam-se como as principais dificuldades a falta de conhecimento sobre as técnicas básicas da Engenharia de Software [Rocha et al 2005] e a falta de entendimento dos modelos de referência de processo de software [Tsukumo 2006]. Estas dificuldades mostram a importância de se estabelecer estratégias adequadas de capacitação dos envolvidos nas iniciativas de melhoria de processos.

Pesquisas na área de treinamento e educação sugerem que o uso de jogos no ensino pode engajar o estudante, reforçando conceitos através da prática, e aprofundando os conhecimentos [El-shamy 2001]. Neste sentido, espera-se que o uso de jogos educacionais também possa contribuir para a capacitação em iniciativas de melhoria de processos, pois eles permitem a simulação de cenários equivalentes ao do mundo real, com situações realísticas, onde as pessoas possam aprender fazendo, através de um ambiente de riscos relativamente baixos, aumentando a confiança para lidar com situações similares no mundo real [Wangenheim 2009]. Além disso, o fato de jogos serem ferramentas de entretenimento tornam o aprendizado mais divertido, sendo assim, um forte fator motivacional [Gibson; Alderich e Prensky 2007].

Este trabalho apresenta um jogo desenvolvido para apoiar a capacitação em Melhoria de Processo de Software com foco no nível G de maturidade do Modelo de Referência MPS para Software (MR-MPS-SW) [SOFTEX 2011]. O jogo simula situações do dia a dia de empresas de desenvolvimento de software em que os resultados esperados do MR-MPS-SW são explorados, permitindo aos jogadores aprenderem Melhoria de Processo de Software vivenciando na prática conteúdos de um modelo de referência de processo de software. O jogo desenvolvido foi avaliado por meio de experimentos que permitiram comprovar que ele auxilia na aprendizagem de Melhoria de Processo de Software.

Este artigo está organizado da seguinte forma: a seção 2 apresenta a pesquisa e análise dos trabalhos correlatos, a seção 3 apresenta o jogo desenvolvido, na seção 4 são apresentados os resultados das avaliações realizadas e na seção 5 são apresentadas as considerações sobre o trabalho desenvolvido e são apontadas pesquisas futuras.

2. Trabalhos Correlatos

Durante a execução deste trabalho, foi realizada uma pesquisa visando avaliar como o conteúdo dos principais modelos de referência de processo de software tem sido explorado pelos jogos educacionais existentes. A seleção dos jogos foi feita por meio da busca de artigos científicos que abordassem jogos educativos e preferencialmente focados no ensino de Melhoria de Processo de Software e Engenharia de Software.

Para encontrar os trabalhos correlatos foram realizadas pesquisas exploratórias nas seguintes bases de artigos científicos: IEEE Explorer; ACM Digital Library; Citeseer library; SpringerLink; WILEY Interscience; ScienceDirect; e Google Scholar. Foram

utilizadas combinações das seguintes sentenças: CMMI; *education*; *game*; ISO/IEC 15504; *learning*; MPS.BR; *software configuration management*; *software construction*; *software design*; *software engineering*; *software maintenance*; *software measurement*; *software process*; *software process improvement*; *software project*; *software quality*; *software requirements*; *software testing*; *teaching*; e *training*.

Foram examinados todos os artigos sobre jogos para o ensino de Engenharia de Software ou Melhoria de Processo de Software publicados entre 1990 e 2012, e foram excluídos os resultados que: descreviam apenas exercícios que não envolviam simulação; descreviam apenas protótipos conceituais de jogos, que foram apenas modelados e ainda não chegaram a ser implementados; não tenham sido publicados em revistas especializadas ou em anais de congressos ou simpósios ou seminários ou não eram trabalhos de conclusão de curso ou dissertações; não tratavam de jogos com finalidades acadêmicas ou educacionais; não forneciam informação sobre o jogo; e não eram voltados para o ensino de Engenharia de Software ou Melhoria de Processo de Software.

Nesta pesquisa foram encontrados 12 jogos educacionais diferentes, publicados na literatura técnica, além disso, foram adicionados mais 4 jogos desenvolvidos dentro do grupo de pesquisa que esse trabalho está inserido. A seguir uma breve descrição de cada um desses jogos é apresentada.

Groupthink Game [Ye; Liu e Polack-Wahl 2007]: Jogo multi-player, desenvolvido no Second Life¹, onde os alunos formam grupos de 7 a 10 pessoas e recebem uma especificação parcial de um sistema. Cada time deve melhorar e completar a especificação recebida e todos os membros da equipe devem concordar e entender o que foi especificado. Para avaliar a especificação elaborada, cada membro da equipe deve responder uma série de perguntas sobre o comportamento do sistema.

MO-SEProcess [Wang e Zhu 2009]: Jogo multi-player, baseado no jogo SimSE [Navarro e Hoek 2009], desenvolvido no SecondLife. Visa ensinar os princípios do processo de desenvolvimento de software através da simulação do desenvolvimento de um projeto de software de tamanho moderado, onde o jogador irá atuar como um engenheiro de software e colaborar com outros jogadores para desenvolver um sistema.

Planager [Prikladnicki; Rosa e Kieling 2007]: Jogo single-player desenvolvido para o ensino de conceitos de gerência de projetos. É focado em cinco processos de duas áreas de conhecimento do PMBOK [PMI 2013] que são: gerenciamento do escopo e gerenciamento do tempo. Possui cinco fases, a primeira fase aborda a definição do escopo, a segunda a criação da Estrutura Analítica de Projeto (EAP), a terceira a definição das atividades, a quarta o sequenciamento das atividades, e por último a quinta fase aborda o caminho crítico.

PlayScrum [Fernandes e Souza 2010]: Jogo de cartas de competição, onde cada participante assume o papel de um Scrum Master. Durante cada sprint o participante deve desenvolver um número de tarefas definidas no início do jogo. O vencedor é o jogador que primeiro executar todas as tarefas definidas para o projeto sem erros ou o jogador que tiver a maior porcentagem de tarefas sem erros no final da última iteração.

¹ <http://secondlife.com/>

Problems and Programmers [Baker; Navarro e Hoek 2005]: Jogo de cartas, multi-player, que simula o modelo de ciclo de vida Cascata. No início do jogo é retirada uma carta de projeto que possui quatro atributos, são eles: complexidade – influencia no quanto se pode avançar na fase de desenvolvimento, tamanho - define o número de cartas de código que devem estar integradas, qualidade - define o número de cartas que serão inspecionadas, e orçamento – restringe o número de desenvolvedores que podem ser contratados. O jogador que primeiro finalizar o processo de integração e conseguir que todas as cartas de código inspecionadas não possuam erro, vence o jogo.

Re-O-Poly [Renel e Gotel 2008]: Jogo de tabuleiro, baseado no jogo Banco Imobiliário, desenvolvido para ensinar um conjunto de práticas de Engenharia de Requisitos. Vence o jogo quem conseguir o maior número de SSP (Pontos de Satisfação de Stakeholders). Para conseguir mais SSP os jogadores devem percorrer o tabuleiro e dependendo do local do tabuleiro em que pararem eles podem: comprar projetos, desafiar seus oponentes fazendo perguntas de Engenharia de Requisitos sobre seus projetos, pegar cartas de tarefas, onde eles devem responder perguntas genéricas de Engenharia de Requisitos, ou pegar cartas de cenários.

SE-RPG [Benitti e Molléri 2008]: Simula o ambiente de desenvolvimento de software de uma empresa fictícia. Durante o jogo o jogador pode executar as seguintes ações: selecionar o modelo de processo (Cascata, Iterativo ou Prototipação) e linguagem de programação que será utilizada, definir a equipe, atribuir tarefas, consultar o desempenho de cada funcionário, acompanhar o andamento do projeto, contratar ou demitir funcionários, etc. O jogo é finalizado caso o jogador consiga concluir todas as tarefas de cada etapa do modelo de processo escolhido e tenha entregado o software para o cliente, ou caso o tempo ou custo gasto no projeto ultrapasse muito o estipulado.

SimSE [Navarro e Hoek 2009]: Jogo single-player no qual o jogador assume o papel de um gerente de projetos e deve gerenciar uma equipe de desenvolvedores, a fim de completar com sucesso um projeto “virtual” de Engenharia de Software. O jogador conduz o processo contratando e demitindo funcionários, atribuindo tarefas para eles, monitorando seus progressos, comprando ferramentas, etc. No final do jogo o jogador recebe uma pontuação indicando quão bem ele executou o jogo, e informações adicionais são reveladas para mostrar ao jogador por que ele recebeu aquela pontuação.

SimulES [Figueiredo et al 2007]: É uma evolução do jogo Problems and Programmers (PnP) [Baker; Navarro e Hoek 2005] que visa resolver algumas limitações do PnP como: a utilização de um único processo de software, utilização de cartas muito abstratas e sem informações adicionais, ausência de um tabuleiro, ausência de um mapeamento claro entre os artefatos do jogo e os conceitos de Engenharia de Software, etc. Como no PnP, o objetivo do SimulES é que os jogadores disputem para terminar um projeto de software e o vencedor será quem implantar o projeto primeiro

The Incredible Manager [Dantas; Barros e Werner 2004]: Jogo single-player no qual o jogador deve montar um plano de projeto que contenha a definição da equipe de desenvolvedores, cronograma e organização da rede de tarefas do projeto. Após a aprovação do plano de projeto o jogador passa para fase de execução do projeto, nesta fase a rede de tarefas do projeto é executada segundo o plano de projeto aceito. O jogo acaba quando os recursos disponíveis para o projeto acabam sem que todas as tarefas

previstas para o projeto tenham sido concluídas ou quando todas as tarefas sejam concluídas dentro das estimativas de recursos previstas pelo planejamento.

X-MED [Wangenheim et al 2009]: Durante o jogo, o aluno executa sequencialmente cada passo de um programa de medição, por exemplo, passo 1 – caracterização do contexto, passo 2 – identificação dos objetivos de medição, passo 3.1 – definição do *abstraction sheet*, etc. Cada passo segue a mesma sequência, onde primeiro a descrição da tarefa é apresentada, depois informações e materiais sobre a tarefa são apresentados, e por último é requisitado que o jogador selecione a solução mais adequada dentre a lista de alternativas apresentadas. Independente da alternativa selecionada, o jogo avança para o próximo passo, usando sempre a alternativa correta para continuar o jogo. No final, uma pontuação total é calculada baseada na soma das pontuações parciais e um relatório final com todas as pontuações parciais e *feedback* é gerado.

U-TEST [Silva 2010]: Jogo de simulação para apoio ao ensino de teste de software, com foco específico em teste de unidade, abordando questões teóricas e práticas. O jogador assume o papel de um testador responsável por escrever teste de unidade para funções já escritas de um sistema hipotético. Durante o jogo o jogador terá de aplicar técnicas de teste de software, como particionamento em classes de equivalência e análise de valor limite.

A Ilha dos Requisitos [Thiry; Zoucas e Gonçalves 2010]: Jogo single-player para ensino de Engenharia de Requisitos, composto por sete desafios. No primeiro desafio deve-se colocar as fases do processo de Engenharia de Requisitos na sequência correta, no segundo deve-se selecionar as habilidades ligadas diretamente a um analista de requisitos, no terceiro deve-se selecionar as frases que correspondem ao problema, no quarto deve-se separar os requisitos funcionais dos não-funcionais, no quinto deve-se lembrar de validar os requisitos e depois responder por que considera-se importante a validação de requisitos, no sexto deve-se colocar os passos necessários para gerenciar uma mudança de requisitos na ordem correta, e no sétimo deve-se selecionar o que deve ser feito depois que uma mudança de requisitos foi aprovada.

123SPI [Adorno 2012]: Jogo de tabuleiro multi-player, baseado no jogo Belts Challenge² sobre Seis Sigma, que foi adaptado para abordar os 18 processos dos níveis 2 e 3 do modelo CMMI. O jogador deve implantar processos em uma organização, e para conseguir implantar esses processos, devem ser respondidas corretamente as perguntas feitas sobre os processos presentes no tabuleiro. Vence o jogo quem conseguir implantar primeiro todos os processos definidos em sua carta de missão.

Deliver [Wangenheim; Savi e Borgatto 2011]: Jogo de tabuleiro multi-player para ensinar gerenciamento do valor agregado (EVM). No início do jogo são adquiridos membros para equipe do projeto, a duração e o custo do projeto são então estimados. Somando o custo total do projeto, o orçamento para término (BAC - *Budget at Completion*) é calculado. Em cada marco, os jogadores devem realizar uma reunião de status analisando a performance do projeto (SPI - *Schedule Performance Index* e CPI - *Cost Performance Index*) e realizar uma previsão (EAC - *Estimate at Completion*). Baseado na performance monitorada, eles devem demitir ou adquirir recursos humanos.

² <http://www.setecnet.com.br/jogos/beltschallenge.php>

O vencedor é o par de jogadores que primeiro entregar o projeto para o cliente chegando no prazo de entrega sem ultrapassar o orçamento do projeto.

Jogo das 7 Falhas [Diniz 2011]: Jogo de simulação que reproduz a execução de casos de teste em que o jogador deve descobrir as falhas existentes nas funcionalidades de um software a ser testado em menos tempo possível. O jogo possui dois níveis de complexidade: baixa e média. Cada nível é composto por uma funcionalidade onde existem sete falhas a serem descobertas. Caso o jogador descubra as sete falhas do nível 1 e as sete falhas do nível 2 dentro do tempo, o jogador é o vencedor.

2.1. Análise dos Trabalhos Correlatos

Como resultado desta pesquisa, percebe-se que o conteúdo dos principais modelos de referência de processo de software é pouco explorado pelos jogos educacionais existentes. Com exceção ao jogo 123SPI, que visa ajudar a ensinar a implementar os 18 processos do nível 3 do modelo CMMI, não foi encontrado mais nenhum jogo com foco específico no ensino de um modelo de referência de processo de software. Entretanto, foi observado que o conteúdo de todos os demais jogos identificados estão relacionados a pelo menos um dos processos do modelo MR-MPS-SW. Esses jogos podem ajudar as pessoas a entenderem os conceitos envolvidos nos processos do MR-MPS-SW. Estes conceitos são necessários para interpretar e implementar corretamente o modelo MR-MPS-SW. Portanto, apesar desses jogos não serem utilizados para demonstrar como implementar modelos de referência de processos, eles podem apoiar o entendimento de conceitos e técnicas da Engenharia de Software e da Melhoria de Processo de Software e, como consequência, apoiar uma interpretação mais adequada de suas práticas e resultados esperados.

Em relação a como o conteúdo é ensinado durante o jogo, alguns jogos realizam perguntas objetivas que os jogadores devem responder para poderem avançar no jogo (RE-O-POLY; X-MED; U-TEST; A Ilha dos Requisitos; 123SPI), outros simulam a execução de atividades do dia a dia de um profissional da computação (Groupthink Game; Planager; Jogo das 7 Falhas), mas a maioria dos jogos estudados simulam a execução de um projeto de software em que o jogador deve tomar decisões para conseguir concluir o projeto com sucesso (MO-SE-Process; PlayScrum; Problems and Programmers; SE-RPG; SIMSE; SimulES; The Incredible Manager; Deliver).

No jogo “SPI City”, que foi desenvolvido e é apresentado neste artigo, o conteúdo de Engenharia de Software é ensinado por meio da simulação de situações do dia a dia de empresas de desenvolvimento de software onde os resultados esperados do MR-MPS-SW são indiretamente discutidos. Desta forma os jogadores podem aprender através de situações típicas do seu dia a dia. Além disso, o jogo se difere dos demais jogos analisados, por ser o único jogo encontrado com foco específico no ensino do modelo MR-MPS-SW.

3. O Jogo Educativo “SPI City”

SPI City é um jogo educacional para apoio a capacitação em Melhoria de Processo de Software, com foco específico no nível G de maturidade do MR-MPS-SW, que contempla dois processos: (1) Gerência de Projetos - GPR e (2) Gerência de Requisitos - GRE. Cada um desses processos contém um conjunto de resultados esperados que são

práticas que devem estar implementadas nas empresas. No nível G de maturidade a Gerência de Projetos contém 19 resultados esperados enquanto a Gerência de Requisitos conta com 5 resultados esperados [SOFTEX 2011]. O objetivo do jogo é ajudar os jogadores a lembrarem, compreenderem e aplicarem os resultados esperados do modelo MR-MPS-SW.

O jogo se passa em uma cidade chamada SPI City, conhecida pela grande quantidade de empresas de desenvolvimento de software de altíssima qualidade instaladas, o que tem atraído muitos investidores para a cidade nos últimos anos. Neste contexto, o jogador receberá um investimento para montar sua própria empresa, mas como exigência, ele deverá alcançar o nível G de maturidade do MR-MPS-SW no prazo máximo de um ano. A partir daí, a história se passa no interior de uma empresa de desenvolvimento de software fictícia, e o jogador no papel de gerente e dono da empresa, deverá resolver situações relacionadas com os resultados esperados do MR-MPS-SW que serão apresentadas durante o jogo. A Figura 1 apresenta a tela principal do jogo, que simula o ambiente interno de uma empresa de desenvolvimento de software, sendo que esta é dividida em três ambientes: setor administrativo (Figura 1-A), setor de desenvolvimento (Figura 1-B) e sala do consultor (Figura 1-C).

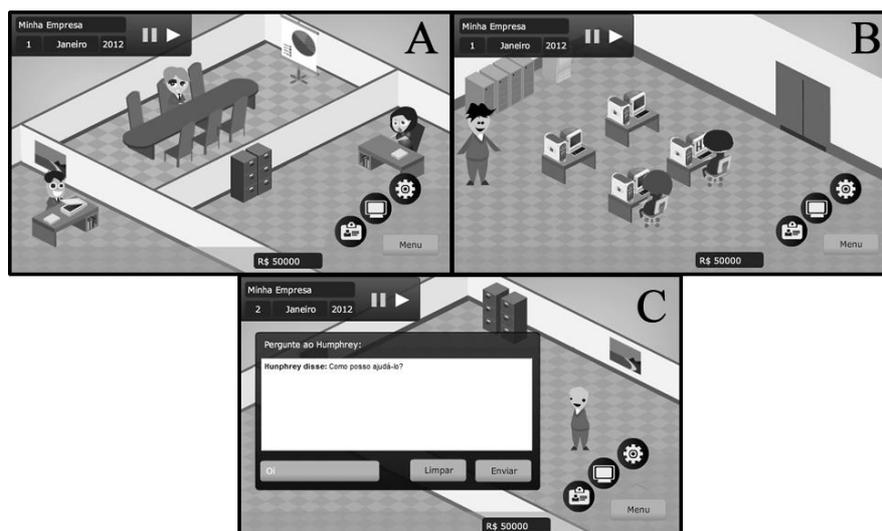


Figura 1. Telas que representam os ambientes da empresa

No setor de desenvolvimento, o jogador pode: contratar, demitir, treinar ou recompensar desenvolvedores; consultar a situação dos projetos de sua empresa; realizar reuniões de acompanhamento de projeto; e conversar com os desenvolvedores ou com o gerente de projetos quando eles precisarem da sua ajuda para resolver alguma situação. No setor administrativo, o jogador pode: consultar suas tarefas; consultar os gastos e receitas de sua empresa; e consultar os resultados esperados do MR-MPS-SW atendidos até o momento. Na sala do consultor, o jogador pode tirar suas dúvidas sobre Melhoria de Processo de Software ou sobre Engenharia de Software.

A principal tarefa do jogador é resolver situações que são apresentadas durante o jogo. Para resolver uma situação o jogador deve conversar com o personagem associado com a situação. A Figura 2 apresenta parte de uma conversa com o gerente de projetos.

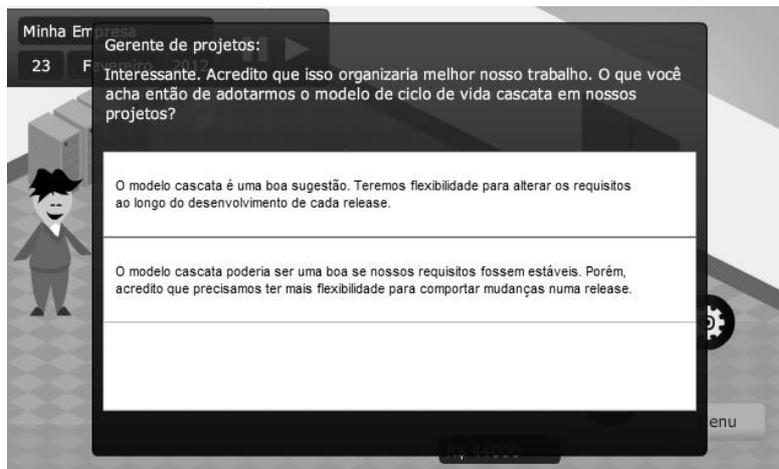


Figura 2. Tela de diálogo (conversa com o gerente de projetos)

Cada situação possui uma lista de diálogos e cada diálogo possui uma ou mais opções de respostas. O jogador pode escolher somente uma opção de resposta para cada diálogo, sendo que só há uma resposta correta. Após selecionar uma opção de resposta, o jogador recebe um *feedback* informando se a opção selecionada está correta ou incorreta, e trazendo explicações adicionais sobre o diálogo. Se a opção incorreta foi selecionada, o diálogo será encerrado e o jogador terá que voltar a conversar com o personagem novamente mais tarde para continuar a resolver a situação. Se a opção correta foi selecionada, será exibido o próximo diálogo, até que se chegue ao último diálogo da situação. Respondendo corretamente todos os diálogos a situação será resolvida. Ao final de uma situação serão apresentados os resultados esperados do MR-MPS-SW atendidos naquela situação, e o jogador poderá consultar os diálogos associados com cada resultado esperado atendido. O jogo será concluído quando o jogador conseguir resolver todas as situações disponíveis.

3.1. Classificação do jogo

O jogo é classificado em relação aos seguintes aspectos:

- **Gênero:** simulador, pois reproduz situações do dia a dia de empresas de desenvolvimento de software.
- **Interação:** por meio de um navegador web o jogo reage aos comandos executados pelo jogador através de mudança do comportamento da tela ou através dos *feedbacks*. O jogador tem também uma interação indireta com outros jogadores, através de um ranking de pontuações que exhibe o resultado de todos os jogadores que já jogaram “SPI City”.
- **Plataforma:** Adobe Flash Player, que além de ser portátil em diversos navegadores web proporciona recursos para desenvolver os mecanismos de interação entre o jogador e o jogo.

3.2. Situações do jogo

O objetivo das situações do jogo é trazer casos comuns que ocorrem em empresas típicas de desenvolvimento de software, em que os resultados esperados do MR-MPS-

SW fossem indiretamente discutidos. Assim, espera-se que os jogadores possam entender com mais facilidade o que é necessário realizar em cada resultado esperado, pois os mesmos estariam associados com situações do seu cotidiano. O procedimento para elaboração dessas situações consistiu na leitura do guia de implementação do nível G de maturidade do MR-MPS-SW [SOFTEX 2011], buscando identificar situações em que os resultados esperados pudessem ser discutidos. Além disso, foram realizadas entrevistas com dois especialistas no modelo MR-MPS-SW, buscando identificar informações de como os resultados esperados são implementados.

Cada situação possui uma lista de diálogos e cada diálogo está associado a um ou mais resultados esperados. Se o jogador responder corretamente ao diálogo, os status dos resultados esperados associados são alterados de acordo com o indicado neste diálogo, sendo: P (parcialmente atendido) - quando o diálogo apenas menciona um ou mais conceitos dos resultados esperados, mas não explica os conceitos; L (largamente atendido) - quando o diálogo explica os conceitos relacionados aos resultados esperados; e T (totalmente atendido) - quando o diálogo dá exemplos de artefatos relacionados aos resultados esperados e de como elaborar esses artefatos. A Tabela 1 apresenta uma situação do jogo e o primeiro diálogo desta situação.

Tabela 1. Situação do jogo (definição da forma de trabalho)

Título	Definição da forma de trabalho.	
Descrição	Seus desenvolvedores estão com uma dúvida sobre como eles devem trabalhar. Minha sugestão é que você vá até a sala de desenvolvimento e converse com eles.	
Personagem	Desenvolvedores.	
Diálogos		Resultados Esperados
<p>Personagem:</p> <ul style="list-style-type: none"> - Estamos em dúvida sobre por onde começar? <p>Jogador:</p> <ul style="list-style-type: none"> - Iremos montar um cronograma. Feedback: Um cronograma é importante, mas para elaborá-lo, você precisa antes saber o que deve ser feito. Pense um pouco mais e converse novamente com sua equipe. - Iremos definir o escopo da próxima versão do produto. Feedback: Boa idéia. Um projeto deve ter um escopo bem definido para que seja possível identificar aquilo que deve ser entregue e estabelecer um prazo final para sua conclusão. - Ainda não sei. Feedback: Pense um pouco mais, mas cuidado para não deixar sua equipe sem saber o que fazer. 		GPR1 = P

Até o momento, o jogo conta com pouco mais de 60 diálogos, distribuídos entre 6 situações diferentes. Todos os resultados esperados dos processos do nível G de maturidade do MR-MPS-SW possuem pelo menos um diálogo associado, porém alguns resultados esperados ainda não estão completamente cobertos pelos diálogos disponíveis no jogo, são eles: GPR11; GPR13; GPR14; GPR16; GRE1; GRE2; e GRE4. Estes resultados esperados ainda não estão completamente cobertos, por que ainda faltam diálogos que expliquem os conceitos relacionados com esses resultados esperados e que apresentem exemplos de artefatos, e de como elaborar esses artefatos, relacionados a esses resultados esperados. Esses diálogos não foram elaborados devido a limitação de prazo dessa pesquisa.

4. Avaliação do Jogo Educativo “SPI City”

Com o objetivo de avaliar a efetividade do jogo desenvolvido foram realizados dois experimentos, um presencial e outro online, e uma avaliação que envolveu somente análise qualitativa. Para apoiar o planejamento dos experimentos foi utilizado o *framework* definido por Kochanski (2009), que objetiva apoiar a construção de experimentos na avaliação empírica de jogos educacionais na área de Engenharia de Software. Para o planejamento da avaliação qualitativa utilizou-se como base o modelo de avaliação de jogos educacionais proposto por Savi (2011).

No experimento presencial inicialmente todos os participantes responderam a um pré-teste, em seguida os participantes do grupo experimental foram submetidos ao jogo desenvolvido e os do grupo de controle jogaram o jogo “A Ilha dos Requisitos” como placebo³. Ao final desta etapa todos os participantes responderam ao pós-teste, e por fim responderam ao questionário qualitativo. O experimento online envolveu a execução dos mesmos passos do experimento presencial, porém não houve divisão de grupos (todos os participantes faziam parte do grupo experimental). Na avaliação que envolveu somente análise qualitativa, os participantes jogaram o jogo desenvolvido por aproximadamente uma hora e em seguida responderam ao questionário qualitativo. Todos os participantes dos experimentos e da avaliação qualitativa possuíam conhecimentos básicos em Engenharia de Software ou Melhoria de Processo de Software, pois eram requisitos para a participação dos mesmos.

Os pré e pós testes foram compostos por 10 questões de múltipla escolha que abordavam os seguintes tópicos: escopo de projeto (GPR1); ciclo de vida de projeto (GPR3); práticas do processo de Gerência de Projetos; práticas do processo de Gerência de Requisitos; identificação de riscos de projeto (GPR6); dados de projeto (GPR9); tamanho de projeto (GPR2); recursos humanos do projeto (GPR7); e mudança de requisitos (GRE5). Para exemplificar, uma das questões dos testes era: “Considerando o nível G de maturidade do modelo MPS.BR, qual das atividades abaixo é a MAIS INDICADA para iniciar um projeto?”. A pontuação de cada questão era de 0 a 1, todas as questões possuíam as mesmas pontuações, e a nota mínima do teste era 0 e a máxima 10. As questões do pós-teste eram as mesmas do pré-teste, porém a sequência das perguntas e das opções de respostas foram alteradas para diminuir a possibilidade de aprendizado de um teste para outro. O questionário qualitativo era composto por 30 questões afirmativas sobre o jogo, e o participante deveria indicar o grau de concordância com cada afirmação, sendo: (1) “Concordo totalmente”, (2) “Concordo parcialmente”, (3) “Indiferente”, (4) “Não concordo parcialmente” e (5) “Não concordo totalmente”. Para exemplificar, uma das questões do questionário qualitativo era: “O design do jogo é atraente”. Ao final dos experimentos os dados coletados foram analisados por meio de testes estatísticos para avaliar a validade das seguintes hipóteses:

- HA₁: As pessoas que utilizam o jogo desenvolvido apresentam melhora na aprendizagem de Melhoria de Processo de Software em relação as pessoas que não utilizam o jogo;

³ Técnica em que apenas os participantes não sabem sobre os detalhes do tratamento que está sendo administrado é denominada teste unicego [Reis, Ciconelli and Faloppa 2002 apud Kochanski 2009].

- H0₁: As pessoas que utilizam o jogo desenvolvido não apresentam melhora na aprendizagem de Melhoria de Processo de Software em relação as pessoas que não utilizam o jogo;
- HA₂: As pessoas que utilizam o jogo desenvolvido apresentam um melhor entendimento dos conceitos de Melhoria de Processo de Software do que as pessoas que não utilizam o jogo;
- H0₂: As pessoas que utilizam o jogo desenvolvido não apresentam um melhor entendimento dos conceitos de Melhoria de Processo de Software do que as pessoas que não utilizam o jogo;
- HA₃: As pessoas que utilizam o jogo desenvolvido aplicam melhor os conceitos de Melhoria de Processo de Software do que as pessoas que não utilizam o jogo;
- H0₃: As pessoas que utilizam o jogo desenvolvido não aplicam melhor os conceitos de Melhoria de Processo de Software do que as pessoas que não utilizam o jogo;
- HA₄: As pessoas que utilizam o jogo desenvolvido consideram que ele é uma ferramenta que motiva o aprendizado de Melhoria de Processo de Software; e
- H0₄: As pessoas que utilizam o jogo desenvolvido não consideram que ele é uma ferramenta que motiva o aprendizado de Melhoria de Processo de Software.

Para analisar o efeito de aprendizagem do jogo desenvolvido por nível de aprendizagem da taxonomia de Bloom (1956), as notas de todos os participantes que jogaram o jogo desenvolvido nos dois experimentos realizados foram tabuladas de acordo com a quantidade de acertos em cada nível de aprendizagem. A Tabela 2 apresenta as notas dos participantes no pré-teste e pós-teste e a quantidade de acertos em cada nível de aprendizagem: Conhecimento (CON), Compreensão (COMP) e Aplicação (AP).

Tabela 2. Notas no pré-teste e pós-teste por nível de aprendizagem

Experimento	Pré-teste				Pós-teste				Diferença
	CON	COMP	AP	Total	CON	COMP	AP	Total	
Online	2,0	1,0	1,0	4,0	3,0	2,5	2,5	8	4,0
Online	2,0	2,0	2,0	6,0	2,0	1,5	1,0	4,5	-1,5
Online	1,0	2,5	1,5	5,0	2,0	2,0	1,0	5,0	0,0
Online	1,5	0,5	0,0	2,0	1,5	1,5	1,0	4,0	2,0
Online	1,0	1,5	1,0	3,5	1,5	3,0	1,0	5,5	2,0
Presencial	1,5	0,5	0,0	2,0	1,0	0,5	1,0	2,5	0,5
Presencial	2,0	2,0	0,0	4,0	0,0	2,0	1,0	3,0	-1,0
Presencial	0,0	0,5	0,5	1,0	1,0	1,5	1,0	3,5	2,5
Presencial	0,5	0,0	0,0	0,5	0,0	3,5	1,0	4,5	4,0
Presencial	1,0	1,0	1,5	3,5	1,0	1,5	1,0	3,5	0,0
Presencial	0,0	1,0	1,0	2,0	0,0	2,5	1,0	3,5	1,5
Presencial	0,0	0,5	0,0	0,5	1,5	0,5	0,0	2,0	1,5

Para análise dos resultados foi aplicado o teste de Wilcoxon [Robson 2002] sobre os dados da Tabela 2. O teste de Wilcoxon é um método não-paramétrico para comparação de duas amostras pareadas, e envolve a execução dos seguintes passos: (1) calcular a diferença entre os pares; (2) ordenar a amostra pela diferença entre os pares,

desconsiderando o sinal e excluindo os valores nulos; (3) realizar a soma das posições na qual a diferença é positiva ou negativa; (4) calcular o p-value de acordo com a soma obtida; e (5) verificar se o valor do p-value é inferior a porcentagem considerada no teste. O resultado da aplicação do teste de Wilcoxon sobre os dados da Tabela 2 é apresentado na Tabela 3.

Tabela 3. Resultados do teste de Wilcoxon

	Conhecimento	Compreensão	Aplicação	Nota final
Z	-,849	-2,209	-1,513	-2,200
p-value	0,396	0,027	0,130	0,028

Analisando os resultados sob a perspectiva do teste de Wilcoxon observa-se que o valor de p-value é menor que 0,05 na nota final e no nível de Compreensão, permitindo que as hipóteses nulas H_{01} e H_{02} sejam rejeitadas, e as hipóteses alternativas HA_1 e HA_2 sejam aceitas. Assim, é possível afirmar com 95% de certeza que as pessoas que utilizam o jogo desenvolvido apresentam melhora no aprendizado de Melhoria de Processo de Software, e apresentam um melhor entendimento dos conceitos de Melhoria de Processo de Software. Como o valor do p-value no nível de Aplicação é superior a 0,05, não é possível rejeitar a hipótese nula H_{03} , desta forma, se aceita a hipótese nula H_{03} e rejeita-se a hipótese alternativa HA_3 . Ou seja, não foi possível afirmar com 95% de certeza que as pessoas que utilizam o jogo desenvolvido aplicam melhor os conceitos de Melhoria de Processo de Software.

Para análise qualitativa foram utilizadas as respostas ao questionário qualitativo de todos os participantes que jogaram o jogo desenvolvido, durante todas as avaliações realizadas, totalizando 32 pessoas. Esta análise considerou os três aspectos (motivação, experiência do usuário e aprendizagem) do modelo de avaliação proposto por Savi (2011). O resultado desta análise permitiu rejeitar a hipótese nula H_{04} e aceitar a hipótese alternativa HA_4 , ou seja, as pessoas que utilizaram o jogo consideram que ele é uma ferramenta que motiva o aprendizado de Melhoria de Processo de Software. Além disso, os participantes tiveram sentimentos de que o jogo auxilia no ensino de Melhoria de Processo de Software. Porém há itens no aspecto experiência de usuário que precisam ser melhorados para deixar o jogo ainda mais atrativo, como por exemplo, a imersão e o desafio.

5. Conclusão e Trabalhos Futuros

O objetivo principal deste trabalho foi desenvolver um jogo para apoiar a capacitação em Melhoria de Processo de Software com foco no nível G de maturidade do MR-MPS-SW, visando oferecer aos alunos de graduação, pós-graduação e profissionais da área de computação/informática um ambiente que permita colocar em prática seus conhecimentos sobre o modelo MR-MPS-SW, através da simulação de situações vividas diariamente por empresas de desenvolvimento de software. Atualmente o jogo pode ser acessado através do seguinte endereço <http://www.incremental.com.br/spicity>.

Visando avaliar a efetividade do jogo desenvolvido foram realizados dois experimentos, um presencial e outro online, além de uma avaliação qualitativa. De acordo com os resultados obtidos, pode-se afirmar que o SPI City auxilia na aprendizagem de Melhoria de Processo de Software de maneira geral, principalmente em nível compreensão (nível entender da taxonomia de Bloom). Já em relação ao nível

de aplicação (nível aplicar da taxonomia de Bloom), não foi possível afirmar que o jogo auxilia na aprendizagem deste nível de conhecimento. Neste ponto, considera-se que aumentar a quantidade de questões voltadas ao nível de aplicação e a condução de mais experimentos fornecerão mais evidências estatísticas.

Por fim, em relação à percepção das pessoas sobre o jogo, os participantes consideraram que o jogo é uma ferramenta que motiva o aprendizado de Melhoria de Processo de Software, e tiveram sentimentos de que o jogo ensina sobre este conteúdo. Embora existam pontos no aspecto “experiência de usuário” que precisam ser melhorados para deixar o jogo ainda mais atrativo, como por exemplo, a imersão e o desafio, a versão atual do jogo já permitiu confirmar que jogos podem ser um instrumento significativo para o ensino de Melhoria de Processo de Software. Porém, considera-se relevante realizar novos experimentos para avaliar a efetividade de aprendizagem de forma a aumentar a força estatística dos resultados.

Agradecimentos

Este trabalho e sua pesquisa contaram com o apoio do Fundo de Apoio à Manutenção e ao Desenvolvimento da Educação Superior (FUMDES), programa de concessão de bolsas de estudo da Secretaria de Educação do Estado de Santa Catarina, e foi desenvolvido dentro do Projeto 16147/2011-2, Jogo Educacional para Apoiar o Ensino de Melhoria de Processos de Software, do Programa Institucional de Bolsas de Iniciação Científica (PIBIC) do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). Os autores agradecem também aos participantes dos experimentos realizados e a todos que contribuíram diretamente ou indiretamente com o desenvolvimento deste trabalho.

Referências

- Adorno, K. S. V. P. **Jogo educacional para apoiar aplicação das áreas de processo do CMMI-DEV: níveis de maturidade 2 e 3**. 2012. 195 f. Dissertação (Mestrado) - Curso de Computação Aplicada, Programa de Mestrado Acadêmico em Computação Aplicada, Universidade do Vale do Itajaí, São José, 2012.
- Baker A.; Navarro, E. O.; Hoek, A. **An experimental card game for teaching software engineering processes**. Journal of Systems and Software 75(1-2): 3-16 (2005).
- Benitti, F. B. V.; Molléri, J. S. **Utilização de um RPG no ensino de gerenciamento e processo de desenvolvimento de software**. In: WEI - Workshop sobre Educação em Computação, 2008, Belém do Pará. Anais do XXVIII Congresso da SBC, 2008. p. 258-267.
- Bloom, B. **Taxonomy of Educational Objectives**. Boston: Allyn and Bacon, 1956.
- Dantas, A. R.; Barros, M. O.; Werner, C. M. L. **Treinamento Experimental com Jogos de Simulação para Gerentes de Projeto de Software**. In: XVIII Simpósio Brasileiro de Engenharia de Software, 2004, Brasília, DF. Anais do XVIII Simpósio Brasileiro de Engenharia de Software, 2004. p. 23-38.

- Diniz, L. L.. **Jogo das 7 Falhas - Um Jogo Educacional para o Apoio ao Ensino do Teste de Caixa-Preta.** 2011. 230 f. Dissertação (Mestrado) - Curso de Computação Aplicada, Universidade do Vale do Itajaí, São José, 2011.
- El-shamy, S. **Training Games: Everything You need to Know About Using Games to Reinforce Learning.** Virginia: Stylus Publishing, 2001.
- Fernandes, J. M.; Souza, S. M. **PlayScrum - A Card Game to Learn The Scrum Agile Method.** In: Second International Conference on Games and Virtual Worlds for Serious Applications, 2010.
- Fernandes, P. G.; Oliveira, J. L.; Mendes, F. F.; Souza, A. **Resultados de Implementação Cooperada do MPS.BR.** ProQualiti (UFLA), v. 3, n.3, p. 11-18, 2007.
- Figueiredo, E.; Lobato, C.; Dias, K.; Leite, J.; Lucena, C. **Um Jogo para o Ensino de Engenharia de Software Centrado na Perspectiva de Evolução.** In: XV Workshop sobre Educação em Computação (WEI), 2007, Rio de Janeiro. co-afiliado ao XXVII Congresso da SBC, 2007. p. 37-46.
- Gibson, D.; Alderich, C.; Prensky, M. **Games and Simulation in Online Learning.** London: INFOSCI, 2007.
- Kochanski, D. **Um framework para apoiar a construção de experimentos na avaliação empírica de jogos educacionais.** 2009. 223f. Dissertação (Mestrado) - Curso de Computação Aplicada, Programa de Mestrado Acadêmico em Computação Aplicada, Universidade do Vale do Itajaí, São José, 2009.
- Montoni, M.; Cerdeiral, C.; Zanetti, D.; Rocha, A. R. **Uma Abordagem para Condução de Iniciativas de Melhoria de Processos de Software.** ProQualiti (UFLA), v. 3, n.3, p. 19-24, 2007.
- Navarro, E.; Hoek, A. **Multi-Site Evaluation of SimSE,** In Proceedings of the The 40th ACM Technical Symposium on Computer Science Education, Chattanooga, TN, March 2009.
- PMI - PROJECT MANAGEMENT INSTITUTE. Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos: Guia PMBOK. Pensilvânia: PMI, Quinta ed. 2013.
- Prikladnicki, R.; Rosa, R.; Kieling, E. **Ensino de Gerência de Projetos de Software com o Planager.** In: XVIII SBIE - Simpósio Brasileiro de Informática na Educação, 2007, São Paulo. XVIII SBIE - Simpósio Brasileiro de Informática na Educação, 2007.
- Renel, S.; Gotel, O. **Gameplay to Introduce and Reinforce Requirements Engineering Practices,** 16th IEEE International Requirements Engineering Conference, Barcelona, Spain, IEEE Computer Society Press. (2008).
- Robson, C. **Real World Research: A Resource for Social Scientists and Practitioner-Researchers, 2nd Edition.** Wiley-Blackwell, 2002.
- Rocha, A.; Montoni, M.; Santos, G.; Oliveira, K.; Natali, A.; Mian, P.; Conte, T.; Mafra, S.; Barreto, A.; Albuquerque, A.; Figueiredo, S.; Soares, A.; Bianchi, F.;

- Cabral, R.; DIAS, A. **Fatores de Sucesso e Dificuldades na Implementação de Processos de Software Utilizando o MR-MPS e o CMMI**. ProQualiti (UFLA), v. 1, p. 13-18, 2005.
- Rodrigues, J. F.; Kirner, T. G. **Benefícios, Fatores de Sucesso e Dificuldades da Implantação do Modelo MPS.BR**. SBQS 2010. Anais, Belém, 2010.
- Savi, R. **Avaliação de jogos voltados para a disseminação do conhecimento**. 2011. 236 f. Tese (Doutor) - Curso de Engenharia e Gestão do Conhecimento, Programa de Pós-graduação em Engenharia e Gestão do Conhecimento, UFSC, Florianópolis, 2011.
- SEI – Software Engineering Institute. **CMMI for Development: Version 1.3: CMMI-DEV**. USA: SEI, 2010.
- Silva, A. C. **Jogo educacional para apoiar o ensino de técnicas para elaboração de testes de unidade**. 2010. 195 f. Dissertação (Mestrado) - Curso de Computação Aplicada, Universidade do Vale do Itajaí, São José, 2010.
- SOFTEX – Associação para Promoção da Excelência do Software Brasileiro. **MPS.BR - Guia Geral 2011**. Junho de 2011. Disponível em: <<http://www.softex.br>>, 2011.
- SOFTEX – Associação para Promoção da Excelência do Software Brasileiro. **Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS, versão 2011**. Julho de 2011. Disponível em: <<http://www.softex.br>>.
- Thiry, M.; Zoucas, A.; Gonçalves, R. Q. **Promovendo a Aprendizagem de Engenharia de Requisitos de Software Através de um Jogo Educativo**. In: XXI SBIE - Simpósio Brasileiro de Informática na Educação, 2010, João Pessoa. XXI SBIE - Simpósio Brasileiro de Informática na Educação, 2010.
- Tsukumo, A. N.; Martino, W. R.; Sérgio, M. P.; Salviano, C. F. **Lições aprendidas na aplicação do Método Coop-MPS para Projetos Cooperativos de Melhoria de Processo de Software com MPS.BR**. ProQualiti (UFLA), v. 2, n. 2, p. 45-56, 2006.
- Wang, T.; Zhu, Q. **A Software Engineering Education Game in a 3-D Online Virtual Environment**, vol. 2, pp.708-710, 2009 First International Workshop on Education Technology and Computer Science, 2009.
- Wangenheim, C. G. V.; Thiry, M., Kochanski, D.; Steil, L., Silva, D.; Lino J. **Desenvolvimento de um jogo para ensino de medição de software**. SBQS, Ouro Preto, 2009. Anais, Ouro Preto, 2009.
- Ye, E.; Liu, C.; Polack-Wahl, J. A. **Enhancing Software Engineering Education Using Teaching Aids in 3-D Online Virtual Worlds**, 37th ASEE/IEEE Frontiers in Education Conference T1E-8, Milwaukee, WI, October 2007.
- Wangenheim, C. G.; Savi, R.; Borgatto, A.; **DELIVER - An Educational Game for Teaching Earned Value Management in Computing Courses**. Information and Software Technology, Elsevier, pre-print, 2011.