SimulES-W: Um Jogo para o Ensino de Engenharia de Software

Elizabeth Suescún Monsalve[1], Vera Maria B. Werneck[2], Julio Cesar Sampaio do Prado Leite[1]

[1] Pontifícia Universidade Católica de Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brasil

[2]Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, Brasil

emonsalveinf.puc-rio.br, vera@ime.uerj.br, www.inf.puc-rio.br/~julio

Abstract. *This paper describes SimulES-W, a software engineering educational card board game. This game allows the student to assume different roles in tasks and decisions of a software construction project. In addition the game encourages competitiveness, testing the student's ability in dealing with problems and applying software engineering concepts to solve these problems or improve their performance in the game, simulating situations difficult to exercise in traditional classrooms. SimulES-W is implemented by software which provides a collaborative infrastructure for Web game playing.*

Resumo. *Este artigo apresenta SimulES-W, um jogo educacional de cartas que ensina engenharia de software. Este jogo permite que um aluno assuma diferentes papéis num projeto de construção de software, permitindo que ele vivencie tarefas e decisões usuais num contexto de produção de software. O jogo estimula a competividade testando a capacidade do aluno em lidar com problemas e aplicar conceitos de engenharia de software para solucionar problemas ou melhorar seu desempenho no jogo, simulando situações que dificilmente ocorrem em aulas tradicionais. SimulES-W é implementado por um software que provê um ambiente colaborativo na Internet.*

1. Introdução

Jogos estão cada vez mais presentes como uma prática habitual no ensino e treinamento, sendo concebidos como uma atividade lúdica que é bastante motivadora no processo de ensino-aprendizado. Desta forma, o uso de jogos para treinar, aprender e executar atividades reais em ambientes realísticos pode melhorar o desempenho dos alunos, pois possibilita a vivência de experiências de aprendizagem que não são fornecidas de forma teórica.

A Engenharia de Software é uma área onde aspectos teóricos e aspectos práticos interagem, tornando fundamentais as decisões e experiências da prática no desenvolvimento de softwares de qualidade, econômicos, úteis e no prazo esperado.

Para contemplar esses aspectos práticos que dificilmente são contemplados pelo ensino tradicional de aulas expositivas, com pouca iteração dos alunos, surgiram propostas do ensino de Engenharia de Software através de pequenos projetos de

desenvolvimento (Pressman, 2006), (Claypool e Claypool, 2005) e (Sweedyk e Keller, 2005). Entretanto, situações vivenciadas na prática, como em projetos de médio a longo porte, dificilmente podem ser exercitadas em sala de aula. O uso de jogos ajuda o enfrentamento desse problema simulando situações e decisões reais vividas pelos engenheiros de software no dia a dia de projetos.

Neste contexto surgiram alguns jogos de tabuleiros, de cartas e simuladores para apoiar o ensino de engenharia de software (Baker, 2003), (SESAM, 2009), (Jain e Boehm, 2006), (Birkhoelzer et al, 2005). SimulES (Figueiredo, 2007), um jogo de tabuleiro para ensino de Engenharia de Software foi proposto com esse objetivo de exercitar, através de simulação, algumas situações e decisões do processo de desenvolvimento de software.

SimulES-W, implementado em um software que utiliza a Internet, daí o W, é a versão digital do SimulES que evoluiu até a versão SimulES 2.0. Essas evoluções foram baseadas na utilização do jogo em diversos cursos de Engenharia de Software na graduação e pós-graduação (Serrano 2007), (Napolitano 2007), (Monsalve, 2010), e (Monsalve, Werneck e Leite, 2010). SimulES-W explora conceitos de complexidade, tamanho e qualidade do produto, de produtividade e maturidade da equipe de desenvolvimento, de gestão de orçamento, de riscos do desenvolvimento versus qualidade e aceitação do produto final. Além desses conceitos básicos os jogadores são expostos a conceitos e problemas gerais de engenharia de software que têm que ser discutidos e argumentados pelos participantes do jogo centrado num processo de colaboração.

A coloboração é outro aspecto importante no processo de produção de software. Segundo Fuks et al (2003) indivíduos que trabalham em grupo podem produzir melhores resultados decorrentes da complementação de capacidades, conhecimentos e de esforços individuais, além da interação entre as pessoas com seus entendimentos, pontos de vista e habilidades.

Neste contexto colaborativo, SimulES-W tem como objetivo ajudar no ensino da engenharia de software de uma forma lúdica, agradável e divertida gerando situações que espelham situações reais num projeto de construção de um software, O jogo é jogado através de tarefas individuais e coletivas segundo regras pré-estabelecidas. Os participantes trabalham com um objetivo individual de ganhar o jogo, além de interagir entre si compartilhando e argumentando informações e conceitos relacionados com a engenharia de software. Ganhar o jogo significa ser o primeiro a entregar o produto a ser produzido durante o jogo.

Este artigo está dividido em cinco Seções. A Seção 2 apresenta alguns jogos usados como ferramentas para o Ensino na Engenharia de Software. Na Seção 3 é fornecida uma visão detalhada do jogo de tabuleiro SimulES na sua versão 2.0, descrevendo suas regras gerais, seus elementos e o objetivo de cada partida. Na Seção 4 descreve-se o SimulES-W, implementação do jogo na Web, que foi baseada na versão de tabuleiro descrita na Seção 3. Finalmente, na Seção 5 concluímos e relatamos como os trabalhos futuros serão abordados.

2. Jogos como Ferramentas de Ensino na Engenharia de Software

Jogos estão cada vez mais presentes como uma prática habitual no ensino e treinamento, sendo concebidos como uma atividade motivadora no processo de ensino-aprendizado. No domínio da Engenharia de Software podemos citar alguns jogos, tais como: PnP (Baker, 2003), SESAM (2009), SimVBSE (Jain e Boehm, 2006), SimSE (Birkhoelzer et al, 2005).

O jogo Problems and Programmers (PnP) é um jogo de cartas educacional direcionado para o ensino geral de engenharia de software (Baker, 2003) cujo objetivo é simular o processo de desenvolvimento de sistemas de uma maneira competitiva entre jogadores. SESAM (Software Engineering Simulation by Animated Models) (2009) é focado no ensino de gestão de projetos através de uma simulação com a criação e execução de um modelo do processo de desenvolvimento de software e executá-lo usando um sistema de simulação. SimVBSE baseia-se no ensino do valor da engenharia de software (Jain e Boehm, 2006) através conceito de “fator crítico de sucesso”. O objetivo deste jogo é integrar esse valor considerando-o em toda a gama existente e emergente dos princípios e práticas de engenharia de software. SimSE permite aos alunos praticar de forma “virtual” o processo de engenharia de software (ou seus sub-processos) de uma forma totalmente gráfica (Birkhoelzer et al, 2005), além disso, permite saber a causa e efeito das complexas relações que permeiam as atividades de engenharia de software.

A tabela 1 apresenta uma síntese dos elementos mais importantes dos jogos apresentados nesse trabalho. Além disso, foram identificadas outras iniciativas que estão em processo de desenvolvimento tais como Scrumming (Isotton, 2008), Planager (Kieling e Rosa, 2006), X-MED (Lino, 2007), GamePlay (Smith e Gotel, 2008) as quais têm como foco realizar simulações para ensinar conceitos de gerência de projetos, práticas ágeis de gerenciamento de projetos, principalmente o SCRUM, medição de software e gerência de requisitos respectivamente.

Tabela 1. Resumo das características de alguns Jogos Educacionais para Ensino na Engenharia de Software

Nome do Jogo	Objetivo do jogo	Objetivo no Jogo
1. Problems and Programmers (PnP)	Ensinar engenharia de software.	Simular o processo de desenvolvimento de software em cascata.
2. SESAM	Ensinar gestão de projetos.	Criar um modelo do processo de desenvolvimento de software e executá-lo usando um sistema de simulação.
3. SimVBSE	Ensinar o conceito de "valor" na engenharia de software.	Identificar os interessados no sistema com aquilo que eles entendem como *fatores críticos de sucesso e valores de preferência* dentro de uma configuração simulada.
4. SimSE	Ensinar processo de engenharia de software.	Completar um projeto de engenharia de software.

3. SimulES

3.1 Visão Geral

SimulES (Figueiredo, 2007) é um jogo de tabuleiro educacional direcionado para o ensino de engenharia de software, concebido a partir da evolução do jogo "Problems and Programmers" (PnP) (Baker, 2003), que é também a base de outros jogos (SESAM, 2009), (Jain e Boehm, 2006), (Birkhoelzer et al, 2005) descritos anteriormente.

Na utilização do SimulES em diversos cursos de Engenharia de Software na graduação e pós-graduação foram identificados requisitos que permitiram uma evolução do SimulES original documentados em (Serrano 2007) e (Napolitano 2007). O resultado dessas experiências foi a definição do SimulES 2.0 (Serrano et al 2007). As descrições e regras apresentadas neste artigo referem-se ao SimulES 2.0 que será a partir de agora denominado simplesmente SimulES.

O objetivo do SimulES é que cada jogador construa um mesmo produto de software. No jogo o aluno/jogador assume vários papeis, tais como: engenheiro de software, coordenador técnico, controlador de qualidade e gerente de projeto. Dentre as tarefas do jogo, jogador deve lidar com: (i) complexidade e tamanho do produto, (ii) noção de tratamento de qualidade do produto com base na verificação por meio de inspeção (iii) risco de se ter produtos de má qualidade, (iv) orçamento em termos de valor do projeto, (v) contratação e demissão de engenheiros de software, (vi) visão de recursos em função do valor, produtividade e maturidade das pessoas envolvidas e (vii) construção de diversos artefatos necessários para término do projeto.

Para ganhar o jogo, entregar o produto, o jogador deve criar uma estratégia que irá melhorar o seu jogo em relação à dos seus adversários e ao mesmo tempo deverá fazer jogadas que desestabilizem os jogos de seus adversários, através das Cartas Problemas. Os adversários podem responder a esses problemas com conceitos de engenharia de software e bloquear o ataque de um jogador. Esse processo é colaborativo no sentido de que todos os jogadores devem estar cientes tanto do problema como do remédio (conceito) e concordar com sua aplicação.

A partida termina quando um jogador constrói todos os módulos necessários para completar o projeto com a qualidade estipulada, onde os módulos do projeto podem ter sidos todos inspecionados ou não. A tarefa de inspeção tem custo associado. Se o jogador escolher não inspecionar todos os seus módulos, a inspeção final obrigatória, segundo o critério de qualidade do projeto, poderá ou não encontrar defeitos. Se o número de defeitos encontrado for maior que o permitido, o jogo continua, porque aquele jogador não atingiu a meta. Essa dinâmica onde qualidade, custo e risco são exercitados é um dos pontos principais sob a ótica do aprendizado. Ou seja, usa-se o término do jogo, primeiro jogador a completar o projeto com seus módulos, como a didática para ensino dos conceitos de qualidade, custo e risco.

3.2 Recursos do Jogo

Os recursos do jogo podem ser visualizados na Figura 1 (Serrano et al, 2009) sendo os seguintes: Tabuleiro Principal, Tabuleiro Individual, Cartas de Artefato (Brancas e Cinzas), Cartão do Projeto, Engenheiros de Software e Cartas Conceitos e Cartas Problemas. Esses recursos que estão, também, presentes no SimulES-W, são apresentados nesta sessão para explicar as características gerais do jogo.

O Tabuleiro Principal (Figura 1 parte a) é colocado no centro da mesa do Jogo, e nele estão dispostos as Cartas de Projeto, as Cartas de Artefato Brancas e Cinzas, as Cartas Conceito e as Cartas Problemas, assim como também uma pilha de Engenheiros de Software. O projeto escolhido deve ficar visível no tabuleiro principal e os lançamentos do dado, também, são efetuados nesta área.

O Tabuleiro Individual (Figura 1 parte b) é o local onde cada jogador coloca seus Engenheiros de Software em colunas e os artefatos que vai construindo em linhas. Os artefatos podem ser dos seguintes tipos: requisito, desenho, código, rastro e ajuda. A quantidade e tipos de artefatos a construir dependerão do projeto escolhido no inicio do jogo. As Cartas de Artefato Brancas e Cinzas são colocadas nas células do tabuleiro, abaixo do engenheiro que as construíram e nas linhas referentes aos seus tipos. Essas cartas simbolizam os produtos construídos pelos engenheiros de software que podem ou não conter defeitos ("bugs"). Cartas de Artefato Brancas custam dois "pontos de tempo" para serem compradas e as Cartas de Artefato Cinzas a metade, mas as de cor branca contêm menos defeitos (proporção de 5 cartas para 1 defeito) que as cinzas (proporção é de 3 para 2). Por isso, custa mais construir o software com cartas brancas mas estas podem trazer maior beneficio na inspeção nos artefatos. Antes de uma inspeção os artefatos estão todos virados para baixo, o que esconde o conhecimento se estão marcadas com um inseto ("bug") no verso ou não. Portanto esse truque do jogo simula um artefato real, no qual a qualidade deixou de ser aferida. O cartão do projeto (Figura 1 parte c) é escolhido no inicio do projeto e é o mesmo durante todo o jogo, contendo os seguintes elementos: (i) Complexidade que indica quantos pontos de tempo um engenheiro de software precisa gastar para construir um bom artefato; (ii) Tamanho que indica o número e os tipos de módulos devem ser construídos e integrados para que o produto software esteja completo; (iii) Qualidade que representa o quão livre de defeitos (bugs) deve estar o produto final, indicando o número mínimo de módulos sem defeitos necessários para submeter o produto e vencer o jogo; (iv) Orçamento que determina a quantidade de dinheiro disponível para gastar com o projeto e representa também uma restrição para contratação de engenheiros de software e para o uso de Cartas de Conceitos.

As Cartas de Conceitos e Problemas (Figura 1 parte e) descrevem boas práticas e problemas clássicos de Engenharia de Software respectivamente. As Cartas de Problema são utilizadas para que os jogadores criem obstáculos ao progresso dos jogos de outros jogadores enquanto que as Cartas Conceito servem para melhorar o progresso no jogo ou para neutralizar uma Carta Problema.

O jogador é um participante do jogo que tem por objetivo vencer o jogo. Em uma partida ele desempenha os papéis de jogador da vez e adversário. O jogador da vez, através de seus engenheiros de software, pode construir um artefato ou corrigir ou inspecionar artefatos ou integrar um artefato ou submeter produto. Os adversários são os oponentes do jogador da vez, principalmente, na rodada de conceitos, no tratamento de problemas e na submissão de produtos.

3.3 Regras do Jogo

As regras do jogo apresentadas em (Serrano et al 2007) foram utilizadas e aprimoradas na versão do SimulES-W como é discutido em (Monsalve, Werneck e Leite, 2010). Estas servem para contextualizar e melhorar o entendimento da dinâmica do jogo e por isso são apresentadas a seguir.

Os jogadores executam uma serie de ações chamadas "Rodadas" dispostas seqüencialmente onde todos os jogadores devem participar de forma ordenada. O jogo possui as seguintes rodadas: Inicio, Ações, Conceitos, Tratamento de problemas e Submissão do produto.

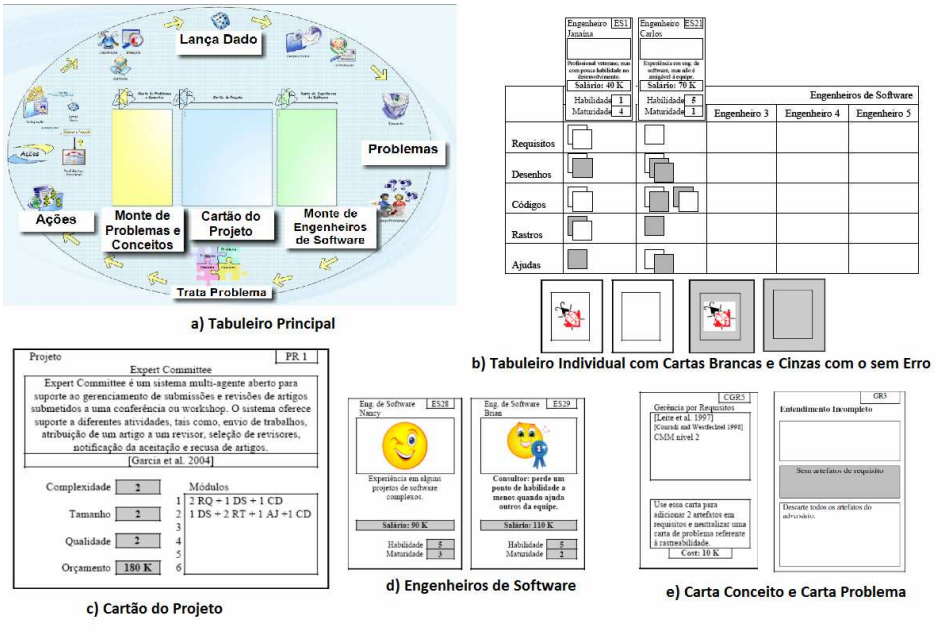

Figura 1. Principais Elementos do SimulES Jogo de Tabuleiro [9]

Na jogada de inicio é escolhido aleatoriamente o projeto que deve ser tratado durante todo o jogo, como também o primeiro Engenheiro de Software para cada um dos jogadores. A jogada de ações começa com o lançamento do dado e de acordo com o número tirado, retira 1 a 3 Cartas de Conceito e Problema e se o valor do dado for mais de 4 pode pegar também (dado–3) Cartas de Engenheiro de Software. A jogada de ações continua e o jogador da vez deve tratar (construir, inspecionar ou empacotar) os artefatos dispostos no tabuleiro individual. Na construção pode retirar novas Cartas de Artefato dependendo da complexidade do projeto e da habilidade de sua equipe de Engenheiros de Software. A habilidade determina quantos “pontos de tempo” ele tem e, portanto, quantas ações ele pode desempenhar por vez. Analogamente ele pode inspecionar dependendo também da habilidade da sua equipe. Na jogada de conceitos e tratamento de problemas os jogadores podem ser atacados e atacam seus adversários usando as Cartas Conceitos e Cartas Problemas com conceitos típicos da engenharia de software. Esta instância do jogo é importante porque permite que os jogadores analisem tanto as Cartas Conceitos e Cartas Problemas que possuem quanto as que possuem seus adversários para realizar a melhor jogada possível. Essas cartas possuem conhecimento de engenharia de software e devem ser lidas, entendidas e discutidas entre os jogadores. Além disso, o professor (jogando ou “só olhando”) pode induzir a uma análise mais profunda que permita aos jogadores melhor aprender os conceitos sendo discutidos. Por último, o produto de software é submetido, sendo que o primeiro jogador a chegar nesta instância ganha o jogo, se o produto passar pela inspeção final e se que não houver penalidade pendente.

Para vencer o jogo, é preciso completar os módulos do projeto definidos no cartão de projeto selecionado no início do jogo, na qualidade exigida. Após serem construídos os artefatos estes devem ser empacotados na Rodada de Ações e depois submetidos como produtos finais.

No decorrer do jogo, o jogador poderá refletir sobre os problemas e conceitos de engenharia de software que tanto ele quanto os demais jogadores apresentem dentro do jogo os quais referenciam a problemas típicos na engenharia de software. O jogador poderá, também, escolher para a construção de seu produto de software a abordagem de desenvolvimento de seus produtos (Figueiredo, 2007), pois, SimulES não obriga a utilização de uma única estratégia ou processo seqüencial, ou seja, o jogador pode estar na construção de código e pode voltar a trabalhar em requisitos ou desenho. No final do jogo ele identificará que uma escolha flexível para a construção de seu produto de software pode ser a mais adequada, além disso, ele notará a evolução dos diferentes artefatos criados no processo de desenvolvimento os quais através da inspeção podem precisar ou não de melhorias que podem ser aplicadas através da correção.

O jogo SimulES em suas várias versões já foi utilizado uma dezena de vezes, sempre com uma retroalimentação positiva sobre seu potencial de ensino. Relatos de validações e de experiências com o jogo podem ser encontrados em (Serrano et al 2007), (Serrano 2007) e (Napolitano 2007). Além disso, a avaliação de sobre o uso de SimulES 2.0 que foi publicada em (Monsalve, Werneck e Leite, 2010) foi utilizada na versão de SimulES-W apresentada na próxima seção.

4. SimulES-W

A evolução do SimulES para o SimulES-W foi realizada com base num estudo de literatura sobre jogos educacionais na engenharia de software, alguns deles apresentados na Sessão 2 deste artigo e em experiências de uso do jogo. Nessa pesquisa identificou-se que as descrições sobre as modelagens do sistema eram vagas ou inexistentes e que a interação entre usuários não era considerada na implementação desses jogos. Assim foi escolhida a modelagem intencional para representação dos modelos de requisitos, pois esta considera a modelagem da interação entre os atores (Yu, 1995). Adotou-se também o método ERi*c (Oliveira, 2008) para facilitar a criação dos modelos iniciais sendo estes baseados no conceito de intencionalidade do “framework” i* (Yu, 1995). A descrição dessa modelagem pode ser encontrada em Monsalve (2010).

SimulES-W foi desenvolvido mapeando as informações dos modelos intencionais até o código e para isso foi escolhida uma arquitetura do tipo MVC (NETBEANS 6.5, 2008) e como ferramenta de desenvolvimento a linguagem Java. Além disso, foram utilizados os seguintes softwares de apoio: MySQL, Visual Web JavaServer Faces, Hibernate e jMaki Ajax.

O SimulES-W é jogado na internet com navegadores padrão, permitindo que jogadores geograficamente separados trabalhem tanto em tarefas individuais como em tarefas em equipe. Esta é uma das vantagens dos sistemas colaborativos: para cooperar os indivíduos trocam informações, se organizam e operam em conjunto num espaço compartilhado (Fuks et al 2003). No caso do SimulES-W esta cooperação é calcada nas interfaces apresentadas no navegador e onde os jogadores negociam, tomam decisões e observam o que está acontecendo, sendo estas ações necessárias para sua atuação dentro da interação.

Na Figura 2 é apresentada a interface principal do SimulES-W onde são dispostas as mensagens trocadas entre os jogadores e as informações sobre jogadas e estado do jogo. Nesta página têm-se, também, as informações de projeto escolhido para

a jogada e os jogadores on-line, e finalmente a lista das jogadas aceitadas ou não entre os jogadores.

Na Figura 3 mostra-se o Tabuleiro Individual em duas situações: a) Construção de Artefato onde o jogador tem os Engenheiros de Software contratados para a construção dos diferentes artefatos, cada um deles com um link que apresenta as informações dos Engenheiros de Software. Na parte a dessa Figura as cartas de artefato brancas e as cartas de artefato cinza aparecem sem terem sido desviradas ainda. Na parte inferior desta página aparecem, também, as operações possíveis a serem realizadas dentro do tabuleiro. A Figura 3 parte b apresenta uma situação de Inspeção de Artefatos, pois o jogador já construiu o artefato (parte a) e ele pode inspecioná-lo. Nessa segunda parte da Figura 3 é ilustrado o resultado de uma inspeção quando o artefato tem erro e quando não tem. SimulES-W aleatoriamente escolhe a qualidade da carta de artefato branca ou carta cinza, sendo que as cartas brancas têm menos possibilidades de terem erro que as cartas cinzas.

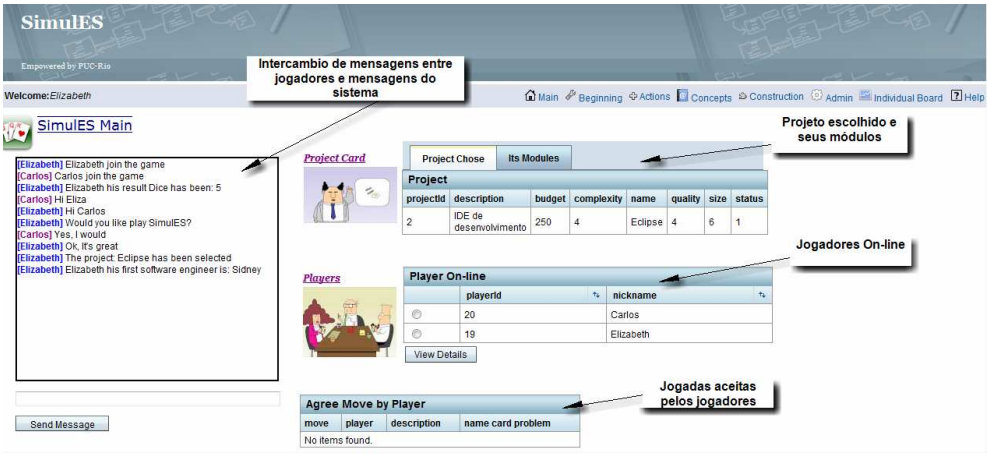

Figura 2. Tela principal de SimulES-W

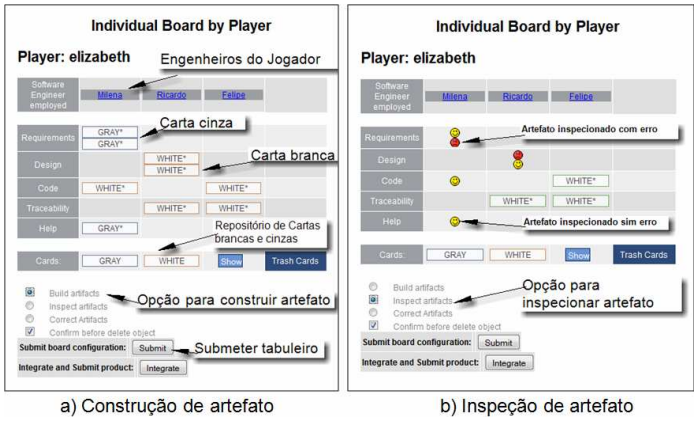

Figura 3. Tabuleiro Individual nas situações de Construção de Artefato e Inspeção de Artefato

SimulES-W (Monsalve 2010) encontra-se em fase de empacotamento para disponibilização como software livre. A primeira validação do SimulES-W (Monsalve 2010) permitiu observar os usuários no ambiente e com isso identificar suas expectativas e implementar as melhorias sugeridas na versão a ser disponibilizada. Essa

versão, além de implementar as características do jogo do tabuleiro possui a vantagens adicionais, tais como: software livre e aberto para continuar sua evolução; funcionamento como um serviço Web rodando num servidor e com interfaces feita no navegador, sendo disponível sem necessidade de realizar "downloads", instalações e configurações; sendo , portanto, menos intrusivas que outras aplicações disponíveis para "download".

Como limitações deste trabalho, pode-se destacar que o Jogo SimulES-W ainda não foi suficientemente utilizado, embora ele já tenha sido submetido a validação nos elementos do jogo de tabuleiro o qual já foi extensamente exposto como foi apresentado anteriormente. Experimentos de validação que permitam melhor avaliar a eficiência/eficácia do SimulES-W no ensino da engenharia de software são trabalhos futuros. Além disso, é preciso comparar tanto o jogo de tabuleiro quanto SimulES-W para identificar restrições e vantagens de cada um dos ambientes do jogo. No entanto, uma das grandes vantagens do SimulES-W sobre as versões anteriores é a possibilidade de edição das Cartas de Problemas e das Cartas de Conceito, fazendo portanto que o material a ser enfatizado pelo curso seja customizado pelo professor.

5. Conclusões e Trabalhos Futuros

Jogos educacionais são promissores porque ajudam na motivação, simulam ambientes reais, podem ser colaborativos e ajudam no entendimento do perfil profissional (Monsalve, Werneck e Leite, 2010), (Gramigma, 1994). Considerando que outras áreas do conhecimento entendem jogos como um aliado da aprendizagem e o fato de que esta tendência é abastecida por recursos tecnológicos cada vez mais sofisticados, entendemos que o SimulES-W é uma plataforma promissora para alcançar resultados eficazes de disseminação de conhecimento de Engenharia de Software.

Lidando com: risco, custo, qualidade com base na verificação por inspeção, conhecimentos pontuais das cartas, a relação da complexidade e do tamanho do projeto, habilidade, maturidade e salário, o aluno tem oportunidade a aprender conceitos tanto sob a ótica da gerência técnica como da gerência de projeto.

Planejamos, para o futuro, a avaliação das diferentes interfaces do SimulES-W, tanto com o foco da comunicabilidade, como com o foco da observação dos jogadores, para que possamos conhecer melhor sobre as interações do jogo. Na distribuição do SimulES-W pretende-se estimular que outros grupos possam também avaliar a eficácia/eficiência do uso do SimulES-W como prática educacional. Essas avaliações devem procurar medir e comparar os resultados do uso do jogo e para isso, métricas precisam ser desenvolvidas.

Referências

Baker, A. (2003). "Problems and Programmers". Honors Thesis, Department of Informatics, School of Information and Computer Science, University of California, Irvine, CA.

Birkhoelzer, T., Navarro, E. e Van Der Hoek, A. (2005). "Teaching by Modeling instead of by Models", Proceedings of the 6th International Workshop on Software Process Simulation and Modeling, MO.

Claypool, K. e Claypool, M. (2005). "Teaching software engineering through game design". ACM SIGCSE Bulletin, v.37 n.3.

Figueiredo, Eduardo; Lobato, Cidiane; Dias, Klessis; Leite, Julio e Lucena, Carlos. (2007). “Um Jogo para o Ensino de Engenharia de Software Centrado na Perspectiva de Evolução” em Workshop sobre Educação em Computação (WEI – 2007), 15, Rio de Janeiro, , 37-46.

Fuks, H., Raposo, A.B. & Gerosa, M.A. (2003) “Do Modelo de Colaboração 3C à Engenharia de Groupware”, Simpósio Brasileiro de Sistemas Multimídia e Web – Webmidia 2003, Trilha especial de Trabalho Cooperativo Assistido por Computador, 03 a 06 de Novembro de 2003, Salvador-BA.

Gramigma, M. R. M. (1994). “Jogos de empresa e técnicas vivenciais”. 1ªEdição, SPaulo, Makron Books.

Isotton, E. (2008). “Ferramenta Educacional para Apoio ao Ensino de Práticas de SCRUM”, Monografia de Trabalho de Conclusão de Curso Graduação, Sistemas Informação, FACIN-PUCRS, Porto Alegre.

Jain, A. e Boehm, B. (2006). “SimVBSE: Developing a Game for Value-Based Software Engineering. Proceedings 19th Conference on Software Engineering Education and Training”. Paginas 103 -114.

Kieling, E.; Rosa, R. Planager (2006), “Um Jogo para Apoio ao Ensino de Conceitos de Gerência de Projetos de Software”, Monografia de Trabalho de Conclusão de Curso de Graduação, Ciência da Computação, FACIN, PUCRS, Porto Alegre.

Lino, J. I. (2007), “Proposta de um Jogo Educacional para Medição e Análise de Software”, Trabalho de Conclusão de Curso, Sistemas de Informação, Universidade Federal de Santa Catarina, Florianópolis.

Monsalve, E. S. (2010), “Construindo um Jogo Educacional com Modelagem Intencional Apoiado em Princípios de Transparência”, Dissertação de Mestrado, PUC–Rio, Março de 2010.

Monsalve, E.; Werneck, V.; Leite, J.C.S.P. (2010), “Evolución de un Juego Educacional de Ingeniería de Software a través de Técnicas de Elicitación de Requisitos”, Proceedings of XIII Workshop on Requirements Engineering (WER’2010), Cuenca, Ecuador, Abril 2010, 12–23.

NETBEANS 6.5 (2008), NetBeans IDE 6.5 Release Candidate Information. Disponível em <http://www.netbeans.org/community/releases/65/>, acesso em: fevereiro de 2010.

Oliveira, A. P. A. (2008). “Engenharia de Requisitos Intencional: Um Método de Elicitação, Modelagem e Análise de Requisitos”, Tese de Doutorado. PUC–Rio, Março de 2008.

Pressman, Roger (2006). “Software Engineering: A Practitioner's Approach”, 7ª Edição, Mc Graw Hill.

Serrano, M.; Serrano, M.; Napolitano, F.; Soares, B. (2007). “Evolução do SimulES Versão 2.0”, Monografia em Ciências da Computação, Departamento de Informática. PUC–Rio.

Serrano, M. (2007) http://mileneserrano.wordpress.com/2007/06/14/um-dia-de-jogo-simules/ acesso em 7 de abril.

Napolitano, F. (2007) http://fnapolitano.blogspot.com/2007/06/simules-simulador-de-engenharia-de.html acesso em 7 de abril.

Smith, R, Gotel, O. (2008). “Gameplay to Introduce and Reinforce Requirements Engineering Practices”, Proceedings of 16th IEEE International Requirements Engineering Conference, ISBN ~ ISSN:1090-705X , 95-104.

Software Engineering Simulation by Animated Models (SESAM) (2009), Stuttgart, Alemanha, Disponível em: <http://www.iste.uni-stuttgart.de/se/research/sesam/overview/index_e.html> acesso em 7 de abril.

Sweedyk, Elizabeth e Keller, Robert M., (2005), “Fun and games: a new software engineering course”, Proceedings of the 10th annual SIGCSE.

Yu, E. (1995). “Modeling Strategic Relationships for Process Reengineering”, Ph.D. Thesis, Graduate Dept. of Comp. Science, University of Toronto.