Utilização de um RPG no Ensino de Gerenciamento e Processo de Desenvolvimento de Software

Fabiane Barreto Vavassori Benitti[1, 2], Jefferson Seide Molléri[1]

[1]Centro Tecnológico da Terra e Mar
Universidade do Vale do Itajaí, (UNIVALI) – Itajaí, SC – Brazil

[2]Departamento de Sistemas e Computação
Universidade Regional de Blumenal (FURB) – Blumenau, SC – Brazil

fabiane.benitti@univali.br, sirvexo@gmail.com

Abstract. *The education on the software engineering discipline is based on the use of models that provides assistance to the management and development of software project. SE•RPG (Software Engineering • Role-playing Game) appears as a pedagogical proposal that intents to support the learning process through the simulation of software company environment, challenging the students with the activities and problems from the development process management. A classroom validation evidenced SE•RPG as a tool capable to minimize the gap between theory and practice in the learning process as well as being a stimulating resource.*

Resumo. *O ensino da disciplina de engenharia de software é fundamentado na utilização de modelos que fornecem subsídios para o gerenciamento e desenvolvimento de um projeto de software. O SE•RPG (Software Engineering • Role-playing Game) surge como uma proposta pedagógica de apoio ao aprendizado através da simulação do ambiente de uma empresa de software, confrontando o acadêmico com as atividades e desafios do gerenciamento do processo de desenvolvimento. Uma validação em sala de aula evidenciou o SE•RPG como ferramenta capaz de minimizar a lacuna existente entre a teoria e a prática no aprendizado constituindo-se, também, em um agente motivador.*

1. Motivação

Poucas áreas da educação, conforme citado por Ohlsson e Johansson (1995), enfrentam desafios semelhantes à engenharia de software quando se trata de selecionar o conteúdo curricular que terá valor na carreira profissional dos estudantes. Não somente porque a engenharia de software é uma área jovem, mas também porque contínuas mudanças ocorrem no campo de seus fundamentos tecnológicos, mudanças estas que se concretizam nos métodos e ferramentas de suporte da engenharia de software.

Quando se trata de abordar, em sala de aula, os conceitos relativos a processo de desenvolvimento de software e gestão de projetos, estes são apresentados através de aulas teóricas e, posteriormente, desenvolvendo pequenos projetos de caráter didático, conforme relatado em Navarro e Hoek (2002), Ohlsson e Johansson (1995), Silva, Leite e Breitman (2004), dentre outros. No entanto, normalmente os projetos desenvolvidos

em sala de aula pelos estudantes, em função do tempo e da natureza didática, não permitem evidenciar diversos aspectos pertinentes ao desenvolvimento de software, como adequação do processo em função de características específicas do projeto (por exemplo: tamanho, complexidade e conhecimento do domínio) e considerar as habilidades e produtividade dos membros de uma equipe para alocação de atividade, para citar alguns exemplos.

Baker, Navarro e Hoek (2003) reforçam que este problema está relacionado na maneira com que a engenharia de software é comumente ensinada, que não abrange adequadamente as ações críticas envolvidas no processo de desenvolvimento e na gestão de projetos. Mesmo que o professor possa explicar a maioria destas ações em aulas expositivas, os estudantes não terão a oportunidade de participar de um processo de desenvolvimento de software completo.

Sendo assim, Baker, Navarro e Hoek (2003) sugerem a utilização de uma abordagem diferenciada no ensino de engenharia de software, utilizando um jogo que simule o processo de desenvolvimento desde a especificação de requisitos até a entrega do sistema, possibilitando aos estudantes uma experiência prática do processo de desenvolvimento mais próxima ao real, que possa ser assimilada com rapidez e utilizada repetidamente durante um período curto de tempo. Esta característica é suportada pelo RPG que, de acordo com o Departamento de Educação da Bahia (2006), possibilita a criação de simulações que seriam custosas, ou mesmo impossíveis de se vivenciar na realidade, provendo ao professor a possibilidade de demonstrar a importância de um determinado conteúdo didático em ambiente real.

A partir das dificuldades vivenciadas em sala de aula e baseando-se nos estudos efetuados ([Backer, Navarro e Hoek 2003], [Navarro e Hoek 2002] e [Ohlsson e Johansson 1995]), desenvolveu-se o SE•RPG (Software Engineering • Role Playing Game), uma ferramenta que auxilia o aprendizado da Engenharia de Software por meio da simulação de um ambiente real, com regras que permitem ao acadêmico se defrontar com as atividades e desafios típicos da gestão de projetos de software.

Este artigo demonstra, segundo o estudo realizado com a ferramenta SE•RPG, os aspectos positivos da utilização do RPG como apoio ao aprendizado no processo de desenvolvimento de software e gestão de projetos, através do resumo dos principais conteúdos referentes à Engenharia de Software que integram o SE•RPG (seção 2); apresenta brevemente o roteiro de jogo descrevendo suas principais funcionalidades (seção 3), bem como aborda as validações (seção 4) e os resultados obtidos (seção 5).

2. Conceitos Abordados pelo SE•RPG

De acordo com Rocha, Maldonado e Weber (2001), os processos que envolvem o ciclo de vida de software são agrupados em três classes que representam a sua natureza: (i) processos fundamentais; (ii) processos organizacionais; e (iii) processos de apoio; definição esta fundamentada na norma ISO/IEC 12207. Os processos apresentados durante o jogo compreendem aspectos e tarefas do ciclo de vida de software nas classes fundamental (envolvendo o processo de desenvolvimento) e organizacional (abordando o processo de gerência).

O processo de desenvolvimento de software, segundo Jalote (1997), é composto por uma seqüência de passos, ou etapas, cada uma das quais executando um conjunto de atividades bem definidas que conduzem à conclusão do projeto. O final de uma etapa conduz ao início de outra, e desta forma são necessárias a verificação e validação de cada etapa, a fim de que sejam geradas as informações corretas para a seguinte.

Embora existam muitos processos de software diferentes, algumas etapas são fundamentais a todos eles, numeradas a maneira de cada autor, como os exemplos a seguir: (i) Pressman (2006) descreve o processo utilizando três fases genéricas: definição, desenvolvimento e manutenção; e (ii) Sommerville (2003) cita as seguintes etapas: especificação, projeto e implementação, validação e evolução do software.

Os modelos de processo de desenvolvimento de software são representações abstratas do processo e suas etapas. Estas abstrações são bastante úteis, visto que podem ser utilizadas para explicar diferentes abordagens do desenvolvimento de software, e entre os principais pode-se citar: (i) modelo em cascata; (ii) modelo iterativo; (iii) modelo de prototipação; (iv) modelo em espiral; (v) técnicas de quarta geração; (vi) modelo formal; e (vii) desenvolvimento orientado a reuso.

Segundo Sommerville (2003), o modelo em Cascata é assim conhecido em virtude da seqüência sistemática de uma fase para outra, onde o resultado de cada etapa dá início à fase seguinte. O SE•RPG permite ao estudante "experimentar" o desenvolvimento baseado no modelo em Cascata abordando as fases de especificação de requisitos, análise e projeto de sistema, implementação e testes.

A abordagem Iterativa, ainda segundo Sommerville (2003), consiste em desenvolver um software de maneira incremental, utilizando o conhecimento adquirido anteriormente, em cada versão disponibilizada, ou seja, cada iteração do desenvolvimento corresponde a uma mini-cascata. O SE•RPG utiliza o modelo Iterativo conforme descrito por Jalote (1997), apresentando em cada iteração as fases de especificação de requisitos, análise e projeto de sistema, implementação e testes.

Referente a gestão de projetos, para Pressman (2006), constitui tarefa fundamental no processo de desenvolvimento de software e envolve uma série de preocupações, como a falta de tempo para executar a tarefa, a complexidade do trabalho ou o orçamento inadequado. Strauss (1997) recomenda a compreensão de três dimensões fundamentais do gerenciamento e suas inter-relações para orientar o procedimento nestas situações: (i) tempo; (ii) tarefa; e (iii) recursos. Estes fatores, mostrados na Figura 1, são fortemente abordados no SE•RPG, constituindo a base para os estudantes tomarem decisões sobre o processo de desenvolvimento e também para nortearem o resultado do jogo.

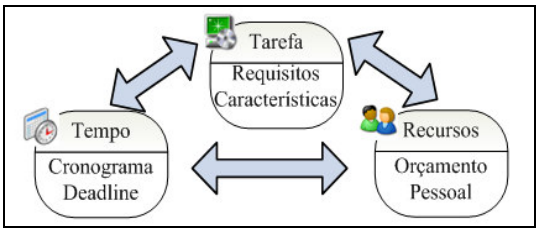

Figura 1. Dimensões do Gerenciamento de Projeto
Fonte: Adaptado de Strauss (1997).

3. SE•RPG

O SE•RPG é uma ferramenta que simula o ambiente de desenvolvimento de software através de um jogo que tem por cenário uma empresa de desenvolvimento fictícia com diversas personagens com as quais o estudante deve interagir durante o progresso do jogo, sendo suportado por um sistema de regras através das quais as ações são validadas.

Ao início do jogo é apresentada uma breve descrição do projeto selecionado contendo o orçamento e prazo para o seu desenvolvimento. Com base nestas informações, o jogador deverá definir em que modelo de processo será o desenvolvimento (sendo que atualmente é abordado apenas o modelo Cascata e Iterativo) e qual linguagem de programação será utilizada durante a etapa de implementação. Após estas definições, cabe ao jogador definir uma equipe para o desenvolvimento, baseando-se num breve resumo das habilidades de cada personagem, conforme ilustra a Figura 2.

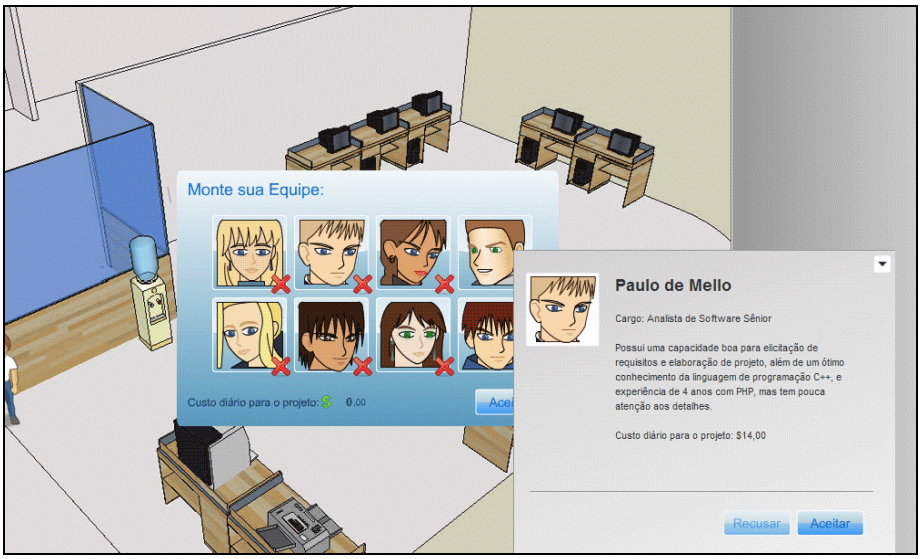

Figura 2. Interface do jogo para a escolha da equipe de desenvolvimento

O jogador pode então iniciar o desenvolvimento do software, atribuindo tarefas para a sua equipe, conforme ilustrado pela Figura 3(A) e (B). As tarefas devem ser atribuídas seqüencialmente a partir da elicitação de requisitos, conforme demonstrado nos diagramas de atividades das Figuras 4a e 4b. No exemplo da Figura 3(D), tem-se representado a escolha pelo modelo Iterativo, podendo-se observar os módulos (no caso, referente a um sistema para clínica médica, sendo Histórico de Pacientes, Controle de materiais, Agendamento de consulta e Financeiro) correspondentes a cada iteração e suas etapas constituindo uma "mini cascata".

O jogador pode controlar o desenvolvimento do projeto, Figura 3(C), verificando o tempo gasto no projeto e o orçamento restante. Além disso, o jogador acompanha o andamento do processo de desenvolvimento, através da ilustração da Figura 3(D). Durante todo o processo de desenvolvimento o jogador pode contratar ou demitir funcionários, conforme consta na Figura 3(E).

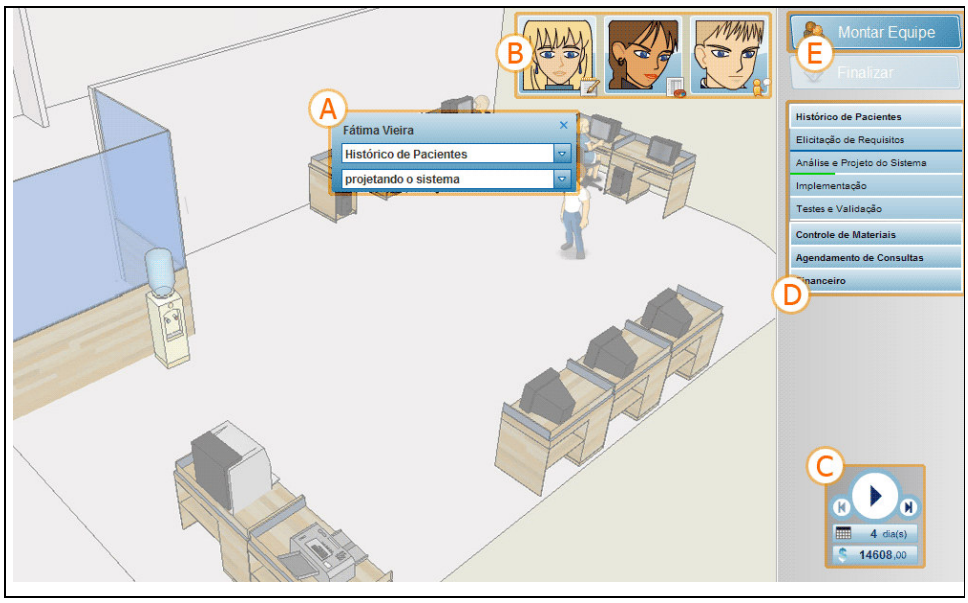

Figura 3. Principais controles do SE•RPG

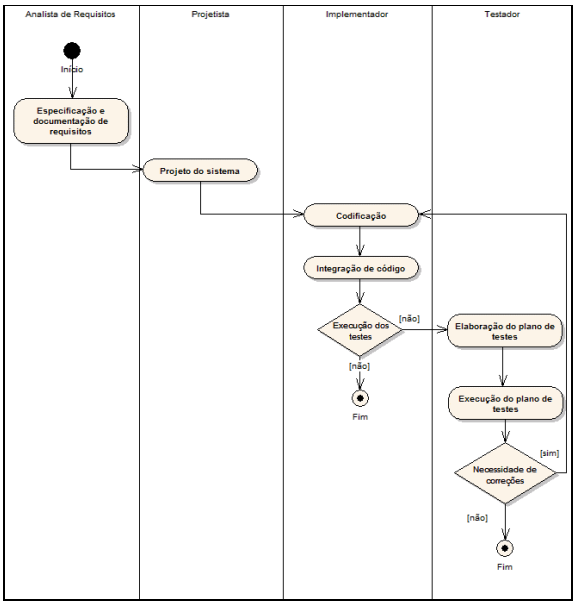

Figura 4a. Etapas para Modelo Cascata

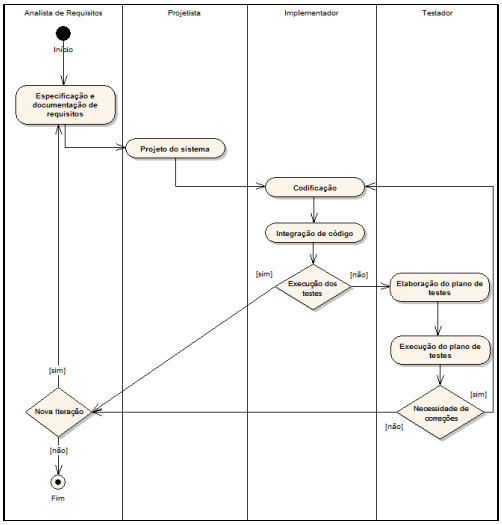

Figura 4b. Etapas SE•RPG para Modelo Iterativo

Durante o andamento do jogo é possível verificar o progresso de cada etapa, bem como a produtividade de cada membro da equipe em separado, da forma ilustrada na Figura 5.

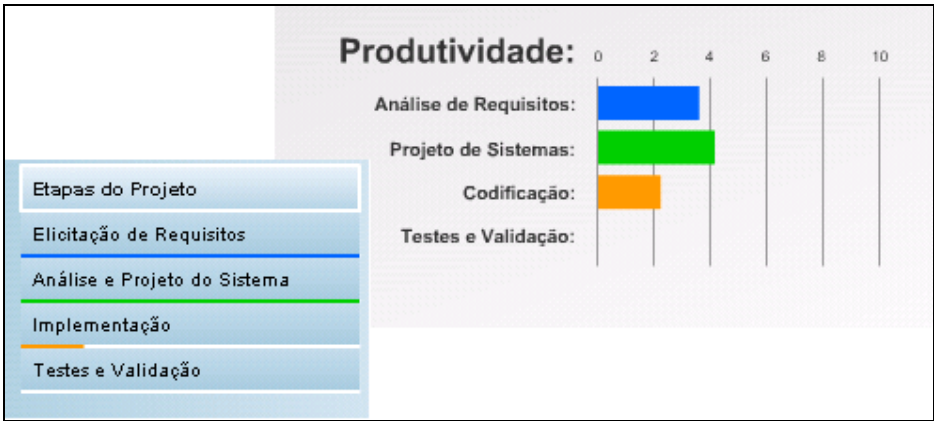

Figura 5. Recursos de acompanhamento do projeto

Ao final da etapa de implementação é possível finalizar o projeto, entregando o software ao cliente e concluindo-se o jogo. É realizado então um breve comentário a respeito das metas do jogador e resultados obtidos, com uma pontuação determinada pelas metas de prazo, custo e escopo, segundo as três dimensões fundamentais apresentadas por Strauss (1997). Também é avaliada a escolha do modelo de processo, considerando as características do projeto (esta informação é definida pelo professor e pré-configurada no sistema quando da definição dos projetos disponíveis). A Figura 6 exemplifica a apresentação do resultado ao jogador.

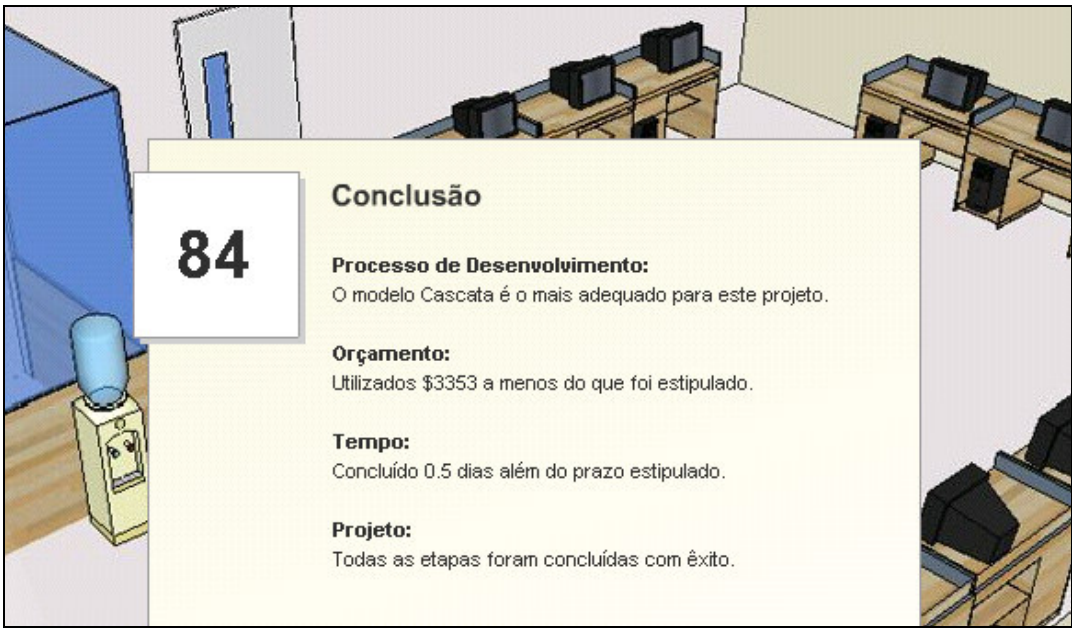

Figura 6. Apresentação do resultado

Para implementação do SE•RPG foi utilizada a linguagem ActionScript utilizando Flash, sendo que as informações referentes aos projetos propostos para desenvolvimento no SE•RPG, bem como os atributos das personagens (que impactam no andamento do jogo) estão definidos em arquivos XML, flexibilizando alterações nos projetos e características das personagens.

4. Validação

Como forma de verificar a adequação do SE•RPG como ferramenta de auxílio ao aprendizado do processo de desenvolvimento de software e gestão de projetos realizou-se uma avaliação na forma de dois questionários aplicados (durante uma aula da disciplina de Engenharia de Software do curso de Ciência da Computação) em dois momentos distintos: antes e depois da utilização do jogo. Cada questionário forneceu uma amostra pareada de 24 acadêmicos, sendo que o questionário inicial forneceu um parâmetro de comparação para os dados coletados na segunda aplicação do questionário.

A fim de se obter a tendência da amostra, aplicou-se o teste t-Student que, de acordo com Barbetta (1994), é um dos métodos mais comuns para comparar os dados coletados, em termos de seus valores médios em uma pequena quantidade de dados amostrais e permite analisar as diferenças entre duas amostras pareadas de dados que possam ser justificadas como fatores casuais ou diferenças reais ou significativas.

Na amostragem coletada, as respostas analisadas são relativas às questões: (i) Quais características estão presentes no modelo em Cascata; e (ii) Quais características estão presentes no modelo Iterativo; ambas de múltipla escolha, contendo as alternativas indicadas na Tabela 1. Estas questões obtiveram um resultado positivo quando observada a quantidade de acadêmicos que acertaram as três alternativas corretas após a utilização do software enquanto houve queda nos que acertaram somente duas ou mesmo uma única alternativa, fatores ilustrados Figuras 7 e 8.

Tabela 1. Análise das Questões por t-Student

Questão	Modelo Cascata	Modelo Iterativo
Alternativas	1. possui uma etapa explícita de análise de riscos 2. os testes são executados uma única vez ao final da implementação 3. há possibilidade da equipe atuar em diferentes etapas simultaneamente 4. as etapas são executadas linearmente (seqüencialmente) 5. compreende, necessariamente, o desenvolvimento de um protótipo 6. todos os requisitos devem ser especificados no início do projeto	1. deve ser definido, a priori, os módulos a serem desenvolvidos e o escopo de cada um 2. possui uma etapa explícita de análise de riscos 3. há possibilidade da equipe atuar em diferentes etapas simultaneamente 4. compreende, necessariamente, o desenvolvimento de um protótipo 5. todos os requisitos devem ser especificados no início do projeto 6. os testes devem ser executados a cada módulo implementado 7. deve haver uma atividade voltada a integração do código
Alternativas corretas	2, 4 e 6	1, 3, 6 e 7
Médias de alternativas corretas	Antes do uso da ferramenta = 1,9 Após o uso da ferramenta = 2,4	Antes do uso da ferramenta = 1,8 Após o uso da ferramenta = 2,1
Tendência da amostra (teste t-Student): $\mathbf{t = (\Delta D * \sqrt{n}) / S_D}$	ti = 2,632	tii = 2,695

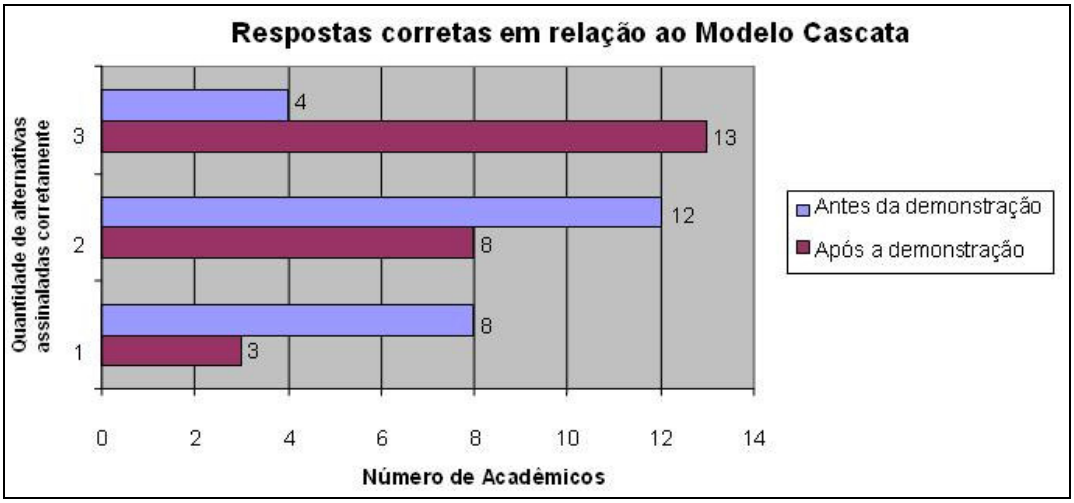

Figura 7. Relação de acertos referente ao Modelo Cascata

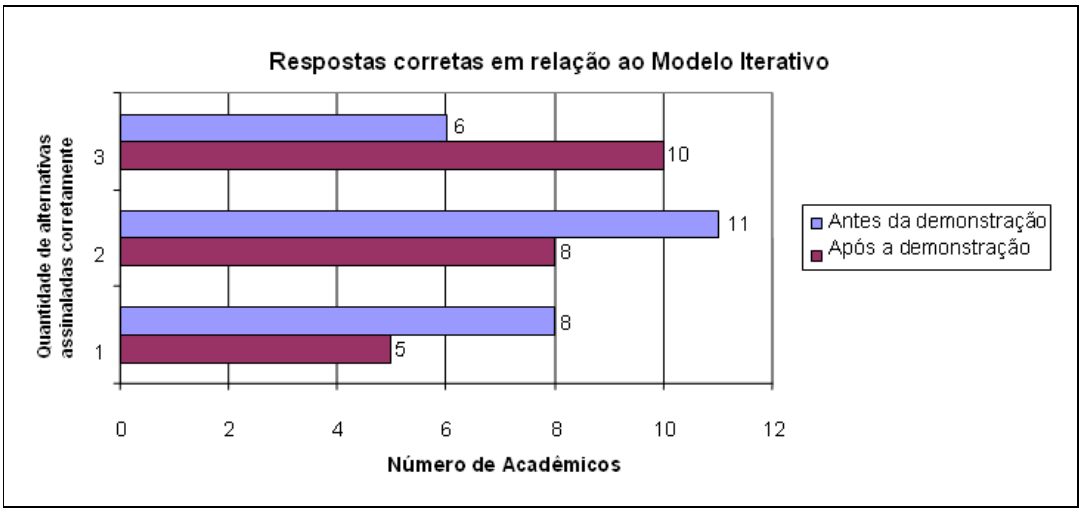

Figura 8. Relação de acertos referente ao Modelo Iterativo

Através da média de alternativas corretas dos acadêmicos para cada questão, ilustrada na Tabela 1, assumiu-se como hipóteses para contribuição fornecida pela ferramenta: (i) H0: a média de acertos nas questões sobre os processos de desenvolvimento não apresentou diferença significativa após o uso do o SE•RPG; e (ii) H1: a média de acertos nas questões sobre os processos de desenvolvimento apresentou diferença significativa após o uso do SE•RPG.

Considerando-se que o valor 2,807[1] corresponda a probabilidade de significância de 95% para a amostra igual a 24, realizou-se o cálculo de t-Student, demonstrado na Tabela 1, obtendo-se os valores de tendência ti = 2,632 para a média de acertos na

[1] Conforme tabela de distribuição t-Student. Mais informações sobre o método estatístico empregado pode ser encontrada em Barbetta (1994).

questão relativa ao modelo de processo em Cascata; e tii = 2,695 para a média de acertos na questão relativa ao modelo Iterativo.

Portanto, conclui-se que ambos os valores ti e tii estão contemplados na probabilidade de significância na distribuição t-Student, o que resulta na afirmação da hipótese H1 e evidencia a contribuição significativa da ferramenta na fixação dos conteúdos relativos aos processos de desenvolvimento em Cascata e Iterativo.

5. Considerações Finais

SE•RPG representa uma alternativa de utilizar um jogo no estilo RPG para auxiliar no ensino de conceitos relativos a gerenciamento de projetos de software e processo de desenvolvimento. O objetivo da ferramenta é permitir aos estudantes de disciplinas da área de Engenharia de Software "experimentar" o papel de gerente de um projeto de software, as decisões e conseqüências, considerando alguns aspectos típicos do cotidiano de um líder de projeto. Também permite ao professor explorar diferentes modelos de processos, suas características, vantagens e desvantagens, favorecendo a criação de uma visão crítica em seus alunos.

A validação em sala de aula evidenciou que o SE•RPG, como ferramenta de apoio, foi capaz de minimizar a lacuna existente entre o aprendizado teórico e a prática, através da simulação de um ambiente de desenvolvimento e, sobretudo, de uma forma motivadora de favorecer o aprendizado. A "visão prática" do processo de desenvolvimento apresentada pela ferramenta foi amplamente mencionada no questionário de avaliação da ferramenta por cerca de 70% (setenta por cento) dos estudantes, enquanto o aspecto motivador e desafiante foi identificado por 50% (cinqüenta por cento) destes, o que evidencia a aceitação do uso da ferramenta no contexto de sala de aula.

Através do teste estatístico apresentado na seção 4 foi possível analisar as diferenças entre as amostras pareadas de dados obtidos com a validação e concluiu-se, através da hipótese confirmada por este teste, que a utilização do jogo em sala de aula obteve uma contribuição significativa no aprendizado dos processos de desenvolvimento de software. Tem-se ciência de que a amostragem é pequena para embasar conclusões definitivas, no entanto, a experiência aponta para aspectos que motivam a sua utilização. Além disso, a ferramenta já foi utilizada por 4 turmas de engenharia de software (sem uma avaliação formal) sendo visível a motivação dos alunos, gerando, inclusive, sugestões de melhorias para o ambiente, dúvidas relacionadas ao conteúdo e maior desenvoltura na utilização de termos da área.

Vale ressaltar que a ferramenta apresentada foi desenvolvida como um instrumento complementar ao ensino dos conceitos e a prática de um processo. Ou seja, tem-se utilizado a ferramenta em sala de aula para reforçar os conceitos teóricos apresentados e fornecer uma visão mais ampla aos alunos, sendo que, atualmente no contexto das disciplinas de Engenharia de Software[2], complementa-se o aprendizado

[2] Em Alves e Benitti (2006) é apresentado o currículo atual da área de Engenharia de Software no curso de Ciência da Computação, sendo composto de 3 disciplinas somando 210hs/a, todas integradas através de um processo de desenvolvimento didático. Neste contexto a ferramenta é usada na primeira disciplina para fornecer uma visão mais ampla sobre processos e gerenciamento.

com a prática em um projeto desenvolvido em grupo, neste momento, explorando um único processo e praticando alguns aspectos específicos.

A avaliação realizada e o uso atual da ferramenta por mais 4 (quatro) turmas de Engenharia de Software (aproximadamente 70 alunos) apóiam o contínuo desenvolvimento da ferramenta de modo a abranger de maneira ainda mais ampla os conteúdos inerentes ao processo de desenvolvimento de software e gestão de projetos. Sendo assim, trabalhos futuros incluem uma nova versão da ferramenta de modo a contemplar: (i) a inclusão do modelo de desenvolvimento de prototipação (não abordado na versão atual); (ii) possibilidade de aquisição de ferramentas CASE, recurso que impactará em maior produtividade da equipe de desenvolvimento e, (iii) ocorrência de "elementos surpresa" ao desenvolvimento como falta de um funcionário (fato que deverá ser gerenciado para não impactar no cumprimento do prazo). A versão atual do SE•RPG pode ser acessada em http://www.inf.furb.br/~fabiane/serpg.

6. Referências

Alves, A. G. e Benitti, F. B. V. (2006) "Processo de Desenvolvimento Integrando Disciplinas de Engenharia de Software", In Congresso da Sociedade Brasileira de Computação (SBC) - Workshop de Ensino de Informática (WEI), 14., Campo Grande, p. 206-215.

Baker, A., Navarro, E. O. e Hoek, A. (2003) "An experimental card game for teaching software engineering", In *Conference on Software Engineering Education and Training*, vol. 16, Spain, http://www.ics.uci.edu/~emilyo/papers/CSEET2003.pdf.

Barbetta, P. A. (1994), Estatística Aplicada às Ciências Sociais, Editora da UFSC, 1ª. edição.

Departamento de Educação da Universidade do Estado da Bahia. (2006) "Ensino on-line: trilhando novas possibilidades pedagógicas mediadas pelos jogos eletrônicos", http://www.comunidadesvirtuais.pro.br/ead/.

Jalote, P. (1997), An integrated approach to software engineering, Springer, 2ª.edição.

Navarro, E. O. e Hoek, A. "Towards Game-Based Simulation as a Method of Teaching Software Engineering," In ASEE/*IEEE Frontiers in Education Conference*. vol 32, 2002, Boston, http://www.ics.uci.edu/~emilyo/papers/FIE2002.pdf.

Ohlsson, L. e Johansson, C. (1995) "A Practice Driven Approach to Software Engineering Education." In *IEEE Transactions on Education*, vol. 38, no. 3.

Pressman, R. S. (2006), Engenharia de software, McGraw-Hill, 6ª. edição.

Rocha, A. R. C. Maldonado, J. C. e Weber, K. C. (2001) Qualidade de Software, Prentice Hall.

Silva, L. F., Leite, J. C. S. P. e Breitman, K. K. (2004). "Ensino de Engenharia de Software: Relato de Experiências." In Congresso da Sociedade Brasileira de Computação (SBC) - Workshop de Ensino de Informática (WEI), 12., Salvador, BA, http://www.inf.puc-rio.br/~lyrene/silva_WEI2004.pdf.

Sommerville, I. (2003), Engenharia de software, Addison Wesley, 6ª. edição.

Strauss, R. (1997), Managing Multimedia Projects, Butterworth-Heinemann Newton.