

SPARSE: Um Ambiente de Ensino e Aprendizado de Engenharia de Software Baseado em Jogos e Simulação

Mariane M. Souza^{1,2}, Rodolfo F. Resende², Lucas S. Prado¹, Edgar F. Fonseca¹,
Flavio A. Carvalho¹, Alexsander D. Rodrigues¹

¹Instituto de Ciências Exatas
Universidade Federal de Alfenas (Unifal-MG) – Alfenas, MG – Brazil

²Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brazil

{mariane,rodolfo}@dcc.ufmg.br,
{a06018,a06005,a06008,a06019}@bcc.unifal-mg.edu.br

Abstract. *Nowadays, a lot have been discussed about Software Engineering teaching and learning. The key issue addressed in many works is the contrast between the essentially practical nature of this area and the teaching methods, which are highly theoretical. This paper presents the game SPARSE, built as a practical approach for teaching and learning Software Engineering, based on games and simulation. This paper also presents the results related to an evaluation made by the use of SPARSE by undergraduate students of different skill levels.*

Resumo. *Atualmente, muito tem se discutido sobre o ensino e aprendizado de Engenharia de Software. O problema chave abordado em vários trabalhos é o contraste entre a natureza essencialmente prática da área e o método de ensino, de característica extremamente teórica. Este artigo apresenta o jogo SPARSE, desenvolvido no contexto de um trabalho mais amplo, como uma abordagem prática para o ensino e aprendizado de Engenharia de Software baseada em jogos e simulação. O artigo apresenta ainda os resultados de uma avaliação do software através da utilização do SPARSE por alunos de graduação de diferentes perfis.*

1. Introdução

O ensino de Engenharia de Software (ES) tem sido amplamente discutido e pesquisado nos últimos anos ([Drappa e Ludewig 2000], [Pfahl et al. 2004], [Baker et al. 2005], [Navarro 2006], [Damian et al. 2006], [Chen et al. 2008]). Neste contexto, o grande problema existente e contemplado em diversos trabalhos é a natureza essencialmente prática da área de ES, que entra em confronto com a maneira predominantemente teórica em que este conteúdo é passado ao aluno durante um curso de graduação.

O processo tradicional de ensino de ES consiste na apresentação de conceitos teóricos relevantes e indispensáveis para o aprendizado do aluno. Embora essencial, pesquisas mostram que apenas esse conhecimento não é suficiente para que o aluno esteja preparado para a realidade de mercado, principalmente com relação à tomada de decisões em ambientes dinâmicos [Damian et al. 2006].

Segundo Navarro [Navarro 2006], alguns cursos procuram amenizar este problema introduzindo noções práticas ao aluno através do desenvolvimento de

pequenos projetos fictícios. Por outro lado, as restrições de prazo e escopo impostas aos mesmos não permitem a vivência de diversas situações típicas de ambientes reais de desenvolvimento de software [Damian et al. 2006].

Como se não bastasse, o conteúdo essencialmente teórico faz diminuir a motivação no aprendizado desta área por parte dos alunos, que não conseguem entender como os problemas surgem em um ambiente real, e quais as suas principais causas e consequências práticas.

A utilização de simulação e jogos educativos tem sido abordada em alguns trabalhos na literatura, com foco no aprendizado através da simulação de gerência de projeto (o ambiente “SESAM” [Drappa e Ludewig 2000], o jogo “The Incredible Manager” [Dantas et al. 2004], o ambiente “VirtualTeam” [Guedes 2006], o jogo “SimSE” [Navarro 2006], dentre outros). A principal vantagem da utilização de jogos como auxílio no aprendizado é a de que os mesmos estimulam e motivam o aprendiz, através da simulação de diversas situações reais do desenvolvimento, além de promover um aprendizado de baixo custo e em menor tempo [Law e Kelton 2000].

Este artigo apresenta o desenvolvimento do jogo SPARSE (*Software Project semi-Automated Reasoning tool for Software Engineering*), bem como os resultados de uma avaliação inicial com o uso do jogo por alunos de diferentes níveis de um curso de graduação. O SPARSE foi desenvolvido no contexto de um trabalho mais amplo, cujo objetivo é definir um método de ensino e aprendizado de Engenharia de Software que combine a teoria passada em sala de aula com uma abordagem prática, capacitando o aprendiz na tomada de decisões futuras em cenários reais.

O artigo está estruturado da seguinte forma: a próxima seção apresenta os principais conceitos relacionados ao uso de simulação e jogos como auxílio no aprendizado, bem como alguns trabalhos relacionados nesta área. A seção 3 descreve o desenvolvimento do jogo SPARSE, mostrando o modelo de simulação escolhido, os detalhes dos elementos que compõem o simulador, e finalmente as interfaces de jogos produzidas. A seção 4 descreve a avaliação do jogo realizada por alunos de graduação, e seus resultados são apresentados na seção 5. A seção 6 apresenta as considerações finais e trabalhos futuros relacionados a essa pesquisa.

2. Ensino e Aprendizado baseado em Jogos e Simulação

O ensino e aprendizado baseados em simulação estão diretamente relacionados à construção de modelos. De forma geral, modelos são considerados abstrações (simplificações) da realidade, utilizados para solucionar problemas que seriam muito caros em uma solução experimental real, ou muito complicados para tratamento analítico [Barros 2001].

A simulação de modelos em computador pode ser utilizada com diferentes objetivos [Hoover e Perry 1989]. Um deles é o de produzir dados e situações que conduzam a tomada de decisões reais imediatas, visando solucionar diferentes problemas. Outro objetivo, relacionado com este trabalho, é o de capacitar e treinar pessoas de maneira mais rápida e barata, de forma que estas se tornem mais aptas a realizarem determinadas funções e resolverem possíveis problemas que encontrarem em um cenário real.

Aliada à simulação, o uso de jogos educativos em diferentes áreas de conhecimento tem surtido efeito, principalmente no que se diz respeito à motivação do aprendiz [Catapan et al. 1999]. No contexto da Engenharia de Software isso é especialmente importante, uma vez que a motivação dos alunos tende a diminuir devido ao excesso de teoria a ser passada aos mesmos.

Segundo Borges [Borges 2005], a pedagogia que utiliza o jogo como uma ferramenta de apoio ao processo de aprendizagem oferece algumas vantagens como ludicidade, cooperação, participação, prazer e motivação. Logo, devido à grande aceitação deste tipo de ferramenta, principalmente pelo público jovem, a adoção dos jogos na área educacional representa um processo natural [Timm et al. 2008].

2.1. Trabalhos Relacionados

Na literatura, diversos trabalhos utilizam a tecnologia de jogos e simulação como auxílio em diversos contextos. Alguns trabalhos que utilizam simulação/jogos como auxílio na capacitação em gestão de projetos e aprendizado são listados a seguir.

SESAM (*Shell for Simulated Agent Systems*) [Drappa e Ludewig 2000] é um ambiente pioneiro na área de simulação para auxílio na gerência de projetos, focado em fornecer uma ferramenta fácil para a construção de modelos complexos. O ambiente não é baseado em jogos, destinando-se mais a profissionais do que estudantes.

Incredible Manager [Dantas et al. 2004] e Virtual Team [Guedes 2006] são jogos baseados em dinâmica de sistemas para o aprendizado do processo de gestão de projetos. Como o objetivo é a capacitação de gerentes de projeto, não é abordado o ensino específico de técnicas de Engenharia de Software. No caso do Incredible Manager, um ponto negativo é a interface utilizada no jogo, que deixa muito a desejar, apresentando problemas tais como baixa interatividade.

SimSE [Navarro 2006], é um jogo baseado em simulação para o aprendizado de ES. O jogo permite que o usuário assuma o papel de gerente de projetos e desempenhe atividades como contratar e demitir desenvolvedores, alocar tarefas, monitorar prazo e custo, entre outras, seguindo uma dada metodologia de ES. A idéia é utilizar a simulação da gerência de projetos como meio para auxiliar o aprendizado de aspectos da Engenharia de Software.

Sendo o SimSE o ambiente que mais se alinha com o objetivo deste trabalho (melhoria de aprendizado de ES), foi realizado um estudo detalhado sobre o mesmo, através da análise de trabalhos publicados e execução de diversos protótipos do jogo. Com isso tornou-se possível identificar as principais regras e conceitos implementados neste ambiente, além de características preocupantes existentes no jogo tais como: inexistência de um modelo de processo bem definido sob o qual o projeto se realiza, uma vez que o principal objetivo é o ensino de Engenharia de Software; a não diferenciação entre empregados que trabalham com e sem horas extras, o que influencia fortemente no planejamento de custos em um ambiente real, dentre outras.

Algumas desvantagens do jogo também puderam ser identificadas, tais como: cenários estáticos com pouca interação com o usuário; falta de instruções para realização das tarefas; ensino superficial e isolado de metodologias de ES, dificultando a escolha da melhor metodologia para cada perfil de projeto e verificação tendenciosa do aprendizado através de questionários informais.

Com base nos pontos negativos observados e na existência de poucos trabalhos na literatura neste contexto, o jogo SPARSE foi desenvolvido como proposta de uma nova abordagem para melhoria do aprendizado de ES, que possa ser aplicada em contextos reais de ensino, aumentando a qualidade na formação dos alunos nesta área.

3. SPARSE (Software Project semi-Automated Tool for Software Engineering)

3.1. Definição do Modelo de Simulação

Com o intuito de criar um modelo de simulação personalizado, o primeiro passo no desenvolvimento do simulador do jogo SPARSE foi determinar o tipo de simulação a ser utilizado, que foi a Simulação baseada em Regras.

A Simulação baseada em Regras permite que o sistema armazene um histórico dos fatos acontecidos, além da possibilidade de observar o comportamento individualizado de cada variável [Barros 2001]. Isso é de suma relevância no contexto do jogo, uma vez que a análise posterior da simulação é um dos pontos chave do aprendizado. Além disso, existem diversas variáveis a serem analisadas, que influenciam diretamente no sucesso/fracasso do projeto de software tais como prazo, custo e qualidade.

3.2. O jogo SPARSE: Elementos e Regras

No simulador do jogo SPARSE estão presentes alguns elementos essenciais no desenvolvimento de qualquer projeto de software, detalhados nas próximas seções:

- *Modelo de processo*: conjunto de atividades pré-definidas e parcialmente ordenadas que um dado projeto deve seguir;
- *Projeto de software*: modelo de processo instanciado com desenvolvedores e restrições de prazo e custo, cujo objetivo final é desenvolver um produto de software e entregá-lo ao cliente;
- *Gerente de projetos*: representa o jogador, responsável por gerenciar todo o processo de desenvolvimento do projeto;
- *Desenvolvedores*: funcionários responsáveis pelo desenvolvimento do projeto, com vários atributos a serem controlados pelo gerente;
- *Ferramentas*: softwares ou técnicas que auxiliam os desenvolvedores no desenvolvimento do projeto.

O jogador possui uma gama de atividades que pode realizar ao longo do jogo, dentre elas: gerenciar os desenvolvedores, planejar o uso das ferramentas, monitorar o progresso do projeto, seguir o modelo de processo definido. Essas atividades são realizadas com o objetivo final de levar o projeto em desenvolvimento à sua completude, preservando o custo e prazo definidos, bem como a qualidade do software.

3.2.1 Modelo de Processo: Fases e Etapas

A gerência de um projeto é feita sobre um *modelo de processo* escolhido para seu desenvolvimento. Um modelo de processo é geralmente separado em *fases*, que representam diferentes estágios da produção de software. Em geral essas fases se diferem, bem como a ordem de execução das mesmas, dependendo do modelo de processo empregado (Modelo Cascata, Modelo Iterativo-Incremental, Modelo Ágil,

dentre outros). A Figura 1 ilustra as fases do Modelo Cascata, implementado nesta primeira versão do jogo.



Figura 1. Modelo Cascata.

No contexto do jogo SPARSE, existem *etapas*, que são os passos de execução de uma determinada fase. Basicamente são caracterizadas por três tipos: criação, revisão e correção. Por exemplo, um desenvolvedor pode estar alocado na etapa de criação da fase de requisitos (ex: criação do documento de requisitos), enquanto outro pode estar alocado na etapa de revisão desta mesma fase (ex: revisão de parte do documento de requisitos produzida até então).

3.2.2. Projeto de Software

O elemento *projeto de software* é o principal de todos. É por ele que passam todas as interações e comandos presentes no jogo. É também este elemento o alvo de todo o gerenciamento feito pelo jogador e também de outras tarefas automáticas do jogo.

O projeto pode ser definido através de alguns elementos que tem o objetivo de torná-lo mais real: o *prazo*, que é o tempo estipulado para a entrega do projeto; os *recursos monetários*, que representam o caixa disponível para gastos com desenvolvedores e ferramentas; e a *qualidade* do produto de software produzido, mensurada com base no número de erros existentes nos artefatos e na gerência de outros elementos, tais como prazo e recursos monetários.

3.2.3. Desenvolvedores e Ferramentas

Para o progresso do projeto são necessários *desenvolvedores*, que ficam a cargo de executar tarefas a eles atribuídas. Os desenvolvedores são como qualquer outra pessoa ou funcionário que possui um conjunto de características, que os diferem dos demais.

Como exemplos de características tratadas no jogo podem ser citados: *energia* (disposição) atual para execução de tarefas; *habilidade* na execução de cada tipo de tarefa; *experiência* que o desenvolvedor possui com relação ao projeto (o quão ele está sendo alocado em tarefas); *salário* do desenvolvedor por período de tempo alocado em alguma tarefa; e *atividades* às quais o desenvolvedor está alocado. O elemento *hora extra* deverá ser contemplado nas próximas versões do jogo, caso o desenvolvedor seja alocado em tarefas por mais de oito horas.

Além das características do desenvolvedor, as *ferramentas* representam outro fator importante, que influencia na maneira em que o projeto deve ser desenvolvido. As ferramentas são softwares ou técnicas de trabalho que agregam valor no desenvolvimento de um projeto, podendo influenciar no tempo de produção, quantidade de erros produzidos, dentre outros.

Os principais atributos das ferramentas, importantes para o simulador do jogo, são: *custo* da ferramenta, podendo ser gratuita ou não; *habilitação*, que determina se a ferramenta está em uso no projeto; *influência* que a ferramenta possui no desenvolvimento de alguma tarefa.

Além de conhecer os principais elementos de um projeto, uma boa gerência consiste ainda em saber trabalhar com as diversas situações e problemas encontrados no

desenvolvimento de um projeto de software [PMI 2008]. Dessa forma, é imprescindível que o sistema possua um conjunto de regras que manipule e simule um cenário real.

3.2.4. Regras

As Regras do simulador foram definidas com base em um conjunto de boas práticas da Engenharia de Software. Tais regras definem a maneira mais correta em que um projeto de software deve ser desenvolvido, influenciando diretamente no andamento do mesmo. Como exemplo de regra, pode-se citar: “A inspeção de um artefato deve ser feita à medida que o mesmo é produzido, para que se diminuam as chances de propagação de erros para as fases posteriores”. Mais exemplos de boas práticas podem ser encontrados no trabalho de Navarro [Navarro 2006].

Além das regras existem eventos, que podem ser aleatórios ou disparados pelas mesmas, com o objetivo de simular situações alternativas, provendo cenários próximos da realidade dos ambientes dinâmicos de desenvolvimento de software. Como exemplo de evento, pode-se citar: “O cliente requisitou o desenvolvimento de novas funcionalidades”.

O ciclo das regras pode ser visto na Figura 2. Este representa a seqüência em que as regras são executadas no sistema, e a sua influência em outras regras. Ex: A regra que define a correção dos erros (Erros Corrigidos) influencia diretamente na regra que define a propagação dos erros para a próxima fase (Erros Propagados).

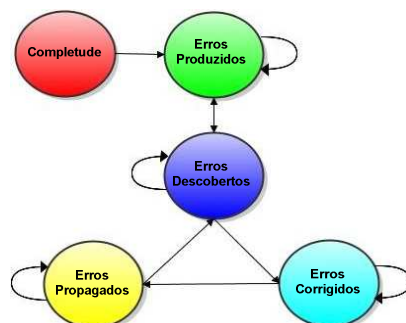


Figura 2 - Ciclo de Regras.

Existem outras regras que compõem o sistema, porém a descrição mais detalhada das mesmas não é o foco da proposta deste artigo.

3.3. Interfaces do Jogo

Atualmente, a interação do jogador com o SPARSE se dá através de uma interface tabular, na qual o jogador visualiza as informações gerais do projeto e de seus componentes. Além disso, é fornecido um resumo textual incluindo todas as suas ações no jogo, à medida que são realizadas, os eventos ocorridos, dicas sobre as regras, dentre outros. Um exemplo desta interface é mostrado na Figura 3.

Em paralelo ao desenvolvimento do simulador está sendo desenvolvido um novo padrão de interface com vários recursos gráficos em 3D. Um resultado parcial desta interface é mostrado na Figura 4. É importante frisar que este novo padrão de interface se comunicará facilmente com o simulador, devido à utilização do padrão *Observer*, na arquitetura do simulador.

The screenshot shows the SPARSE interface with a tabular view. It includes a table for developers, a table for project phases, and various control panels for tools, activity allocation, and simulation.

Nome	Energia (%)	Experiencia (%)	Salario Por Hora (R\$)	Atividades	Requisitos	Projeto	Codificacao	Teste
Gustavo	25	1	9,5	Projeto	0,1	0,2	0,2	0,1
Tiago	12	1	7,5	Codificacao	0,1	0,5	0,1	0,1
Bianca	36	1	9	Teste	0,25	0,2	0,5	0,7
Bruno	12	1	10	Requisitos	0,5	0,5	0,5	0,5
Jessica	60	1	12	Codificacao	0,8	0,7	0,3	0,2
Rodrigo	37	0,95	8		0,1	0,3	0,85	0,1

Nome	Completo (%)	Erros Descobertos	Desenvolvedores Alocados
Requisitos	95,893	64,779	[Bruno]
Projeto	98,674	28,246	[Gustavo]
Codificacao	92,668	110,368	[Tiago, Jessica]
Teste	92,01	64,118	[Bianca]

Figura 3. SPARSE: Interface tabular.



Figura 4. SPARSE: Interface com recursos gráficos em 3D.

4. O uso do jogo SPARSE por alunos de um curso de graduação

Como estratégia de elaboração de melhorias nas próximas versões do jogo e de auxílio em validações futuras, uma avaliação inicial do jogo SPARSE foi realizada. Essa avaliação foi conduzida através do uso de questionários, empregando perfis aos usuários do sistema utilizado (versão tabular exibida na Figura 3) e também dimensionando uma possível evolução dos conceitos de Engenharia de Software e Gerência de Projetos (GP) destes indivíduos após o uso do jogo.

Previamente à avaliação, foram selecionados aleatoriamente oito graduandos, cursando Bacharelado em Ciência da Computação, em diferentes etapas do curso. Esta seleção possibilitou a coleta de opiniões de diferentes públicos, relacionados a discentes com diferentes níveis de conhecimento em ES e GP.

Após a seleção dos participantes, um questionário inicial de múltipla escolha foi empregado, com o objetivo de se estabelecer o perfil de conhecimento de cada jogador. É importante frisar que nenhum tipo de instrução formal relacionada a qualquer tópico de ES ou GP foi fornecida aos estudantes, antes ou durante a avaliação, sendo exibidas apenas explicações gerais sobre o funcionamento do jogo.

Após a avaliação de conhecimento dos participantes, os mesmos foram apresentados ao software, realizando as primeiras interações como gerentes de projetos

do jogo. Para maior eficácia do experimento, cada usuário executou o processo de desenvolvimento para o projeto piloto, proposto no simulador, por quatro vezes, sendo cada decisão tomada na simulação, bem como a pontuação obtida no jogo, registrada em arquivos de *log* por usuário, para avaliações posteriores dos resultados de aprendizado obtido.

Após o uso do jogo, um segundo questionário foi aplicado a cada usuário participante do experimento. Neste questionário foram verificadas as experiências individuais dos participantes no jogo apresentado, juntamente com questões discursivas referentes ao conteúdo de ES e GP.

5. Análise de Resultados

Por meio da avaliação detalhada na seção anterior tornou-se possível coletar um conjunto de opiniões e sugestões de melhoria para o protótipo desenvolvido. Pôde-se perceber ainda uma evolução no aprendizado de conceitos de ES e GP após o uso do SPARSE pelos alunos.

Finalizada a avaliação, os questionários foram analisados, e com base nos dados obtidos foram produzidos gráficos com relação ao perfil dos usuários. Os principais são mostrados na Figura 5 e Figura 6.

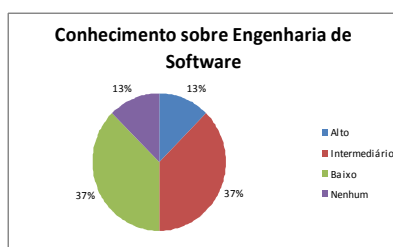


Figura 5. SPARSE: Resultados da avaliação com relação ao conhecimento sobre Engenharia de Software.

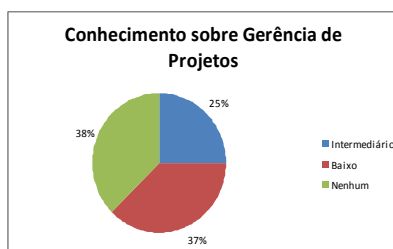


Figura 6. SPARSE: Resultados da avaliação com relação ao conhecimento sobre Gerência de Projetos.

Após a análise de perfil, os questionários referentes a opiniões sobre o jogo e conhecimento adquirido com o uso do mesmo foram analisados. Com relação às opiniões sobre o jogo foram obtidos os resultados descritos de forma resumida na Figura 7, em que cada usuário fornecia uma nota de 0 a 5 para cada característica observada.

Observando a Figura 7, percebe-se a satisfação da maioria dos jogadores com relação ao uso da ferramenta proposta, o que foi comprovado ainda por respostas discursivas, demonstrando opiniões positivas com relação ao jogo, com pequenas sugestões de melhoria.

O aumento em geral do conhecimento de ES e GP também pôde ser observado através da grande quantidade de respostas corretas relacionadas aos exercícios discursivos propostos, definidos com base em [Pressman 2007] e [PMI 2008]. Devido ao baixo conhecimento de ES e GP, anteriormente detectado pela análise de perfil, percebe-se que a corretude das respostas pode estar condicionada ao aumento do entendimento do conteúdo proposto através do uso do SPARSE.

Observou-se ainda que discentes com pouco ou nenhum contato anterior com assuntos de ES e GP obtiveram consideráveis taxas de assimilação do conteúdo pretendido, assim como outros com maior familiaridade com conceitos de ES e GP, o que pode favorecer o uso do jogo em disciplinas da área, em diferentes níveis do curso.

É importante destacar que a avaliação aqui detalhada possui limitações, dentre elas o tamanho da amostra (oito alunos). Apesar disso, foi consideravelmente importante para destacar os pontos positivos e negativos relacionados à ferramenta, de modo que novas versões possam ser aprimoradas e testadas de forma mais criteriosa, através de um experimento formal.

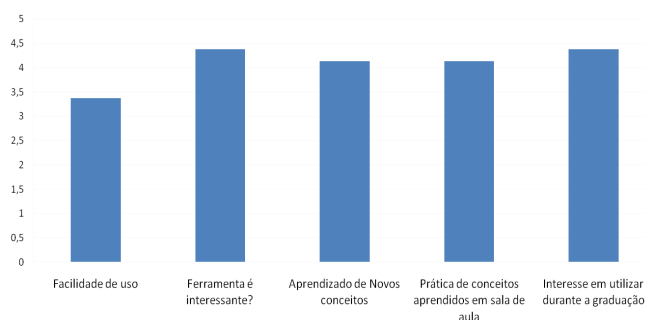


Figura 7. SPARSE: Resultados do experimento com relação à eficácia da ferramenta.

6. Considerações Finais

Este artigo apresentou o desenvolvimento e avaliação inicial do jogo SPARSE, proposto como uma abordagem prática para melhoria do aprendizado de Engenharia de Software.

A avaliação inicial através do uso do jogo por alunos de graduação foi de suma importância para adquirir informações e sugestões de melhoria para o mesmo. Vale destacar que esta avaliação serve tanto como referência na construção de novas versões, quanto como um auxílio futuro na validação formal do aprendizado, através de experimentos contrastando o uso do SPARSE com outras abordagens de ensino.

Como trabalhos futuros desta pesquisa podem ser citados:

- Produção de novas versões do jogo a partir das informações adquiridas com a avaliação inicial realizada, incluindo a versão com interface 3D.
- Validação formal do aprendizado através de experimentos com o uso do jogo SPARSE contrastado com o uso de outras abordagens, por exemplo, aulas teóricas sobre o tema, exclusivamente;
- Adaptação do SPARSE para simulação de modelos construídos com base na estrutura de WBS (*Work Breakdown Structure*), visando maior generalização do simulador, dispensando a construção de ferramentas específicas de modelagem.

Referências

- Baker, A., Navarro, E. e Hoek, A. (2006) "An Experimental Card Game for Teaching Software Engineering Processes". *Journal of Systems and Software*.
- Barros, M. (2001) "Gerenciamento de Projetos Baseados em Cenários". Tese de PhD, UFRJ, Rio de Janeiro, Brasil.
- Borges, C. J. (2005) "O Lúdico nas Interfaces das Relações Educativas", *Revista de Pedagogia*, N 12 Vol. 6.
- Catapan, A. H., Plínio, C. F., Souza, A. C., Thomé, Z. R. C., Cybis, W. D. A. (1999). "Ergonomia em software educacional: A possível integração entre usabilidade e aprendizagem", In *II Workshop sobre Fatores Humanos em Sistemas Computacionais*, Florianópolis - SC.
- Damian, D., Hadwin, A., Al-Abni, B. (2006) "Instructional Design and Assessment Strategies for Teaching Global Software Development: A Framework" in *Proceedings of the International Conference on Software Engineering*. ACM.
- Dantas, A. R., Barros, M. O., Werner, C.M.L. (2004) "A Simulation-Based Game for Project Management Experiential Learning" in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering*, Banff, Alberta, Canada.
- Drappa, A. e Ludewig, J. (2000) "Simulation in Software Engineering Training" in *Proceedings of the 22nd International Conference on Software Engineering*. ACM.
- Guedes, M. S. (2006) "Um Modelo Integrado Para Construção de Jogos de Computador Aplicado à Capacitação em Gestão de Projetos". Dissertação de Mestrado, Universidade Federal de Pernambuco. Recife, Brasil.
- Hoover, S. V., Perry, R. F. (1989) "Simulation: a problem-solving approach". Reading Massachusetts: Addison-Wesley Longman Publishing Co., Inc. Boston, EUA.
- Law, A. M. e Kelton, W. D. (2000) *Simulation Modeling and Analysis*. 3 ed.: McGraw-Hill Companies, Inc.
- Navarro, E. (2006) "SimSE: A Software Engineering Simulation Environment for Software Process Education". Tese de Doutorado do Departamento de Informática da Universidade da Califórnia, Irvine, 2006.
- Pfahl, D., Laitengerger, O., Gunther R., Dorsch, J., Krivobokova, T. (2004) "Evaluating the learning effectiveness of using simulations in software project management education: results from a twice replicated experiment". *Information and Software Technology*.
- PMI (2008) *Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos (Guia PMBOK)*, Project Management Institute, Quarta edição.
- Pressman, R. (2007) *Engenharia de Software*. 8. ed. São Paulo: McGraw-Hill
- Timm, M. I., Ribeiro, L. O. M., Lando, V. R., Azevedo, M. P., Vieira, E. (2008) "Game educacional: desafios da integração de elementos ficcionais, tecnológicos, cognitivos e de conteúdo". In: *SBGames 2008 - VII Symposium on Computer Games and Digital Entertainment*, Belo Horizonte.
- Chen, W., Wu, W., Wang, T., Su, C. (2008) "Work in Progress – A Game-based Learning System for Software Engineering Education". *IEEE Frontiers in Education Conference*.