

Uso de Jogos para o Ensino de Engenharia de Software

-- Relatos de Experiência --

Lucas Meirelles¹, Daniela Peixoto¹, Elizabeth Monsalve², Eduardo Figueiredo¹,
Vera Werneck³, Julio C. S. P. Leite², Rodolfo Resende¹, Clarindo Pádua¹

¹Departamento de Ciência da Computação, Universidade Federal de Minas Gerais (UFMG)

²Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brasil

³Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, Brasil

{lucas.meirelles, cascini, figueiredo, rodolfo, clarindo}@dcc.ufmg.br,
{emonsalve, julio}@inf.puc-rio.br, vera@ime.uerj.br

***Resumo.** Durante o ensino de qualquer disciplina é importante a aplicação prática do conhecimento adquirido para fixar o aprendizado. No ensino de Engenharia de Software nem sempre é possível utilizar-se de atividades reais de desenvolvimento e gerência de software devido à limitação de tempo do curso e à complexidade de projetos. Desta forma, uma alternativa para integrar o ensino teórico à prática é a utilização de jogos que simulam tais atividades. Este trabalho apresenta o esforço conjunto de professores e alunos de pós-graduação de três universidades brasileiras em implantar o uso de jogos educacionais para apoiar o ensino de Engenharia de Software. Além disso, o artigo relata e discute as lições aprendidas em mais de cinco anos de estudo de jogos para Engenharia de Software. Finalmente, por meio deste trabalho, nosso objetivo estratégico é angariar novos educadores interessados em compartilhar experiências no uso de estratégias de apoio ao ensino de Engenharia de Software.*

1. Introdução

Ensinar é uma tarefa com grandes desafios, pois envolve não apenas adquirir conhecimento sobre um assunto específico, mas também ser capaz de transmiti-lo a outras pessoas de forma amena e natural. Um curso típico de Engenharia de Software é constituído principalmente de aulas teóricas e de pequenos trabalhos práticos. Entretanto, é difícil para os alunos assimilarem em aulas teóricas todos os conceitos importantes da Engenharia de Software, como gerência de pessoas, gerência de riscos, gerência de tempo e controle de versões, pois alguns conceitos teóricos ficam distantes da prática [5, 6, 7, 13].

Para complementar as aulas teóricas os professores geralmente fazem uso de exercícios e trabalhos práticos de desenvolvimento. O tamanho dos grupos e limitações de tempo e recursos, entretanto, também são obstáculos para que os conceitos de Engenharia de Software sejam exercitados em sua plenitude. Além disso, instituições e educadores têm diferentes prioridades sobre quais assuntos da disciplina são mais relevantes e deveriam ter ênfase durante o curso [5].

Para suprir essas necessidades, jogos educacionais [2, 6, 8, 11, 12] têm sido utilizados para ilustrar de forma prática e intuitiva os mais diferentes conceitos

envolvidos na disciplina de Engenharia de Software. Um dos objetivos é fornecer ao professor ferramentas para transmitir conhecimento durante suas aulas de maneira natural e divertida, mas ao mesmo tempo eficaz [7]. Diversos jogos já foram propostos [2, 6, 8, 10, 11, 12] e cada um deles tem um foco específico e define regras para permitir aos alunos exercitar um conjunto de conceitos de Engenharia de Software. Entretanto, apesar de cada vez mais educadores reconhecerem a importância do uso de jogos educacionais, ainda são escassos os relatos de esforços unificados sobre a adoção de tais jogos em diversas instituições de ensino.

Assim, este trabalho tem o objetivo de relatar os resultados e observações obtidas de estudos com jogos para apoio ao ensino de Engenharia de Software feitos por professores e alunos de pós-graduação em três diferentes instituições de ensino. O nosso objetivo é comparar e discutir nossas experiências, não só entre nossos respectivos grupos, mas também com a comunidade de educação em Engenharia de Software. Além disso, buscamos a definição, junto à comunidade, de diretrizes gerais para o uso de jogos educacionais desta natureza.

O restante deste artigo é organizado da seguinte forma: a Seção 2 contextualiza o uso de jogos no ensino de Engenharia de Software. A Seção 3 discute experiências de ensino através de dois jogos de cartas. As Seções 4 e 5 analisam a interação de alunos de duas instituições com dois jogos virtuais. A Seção 6 resume as principais lições aprendidas com os estudos enquanto a Seção 7 conclui este artigo apontando direções para trabalhos futuros.

2. Jogos Educacionais para Engenharia de Software

Diversos jogos têm sido propostos para apoiar o ensino de Engenharia de Software [2, 6, 8, 10, 11, 12]. Entre os jogos educacionais existentes, podemos citar, por exemplo, o PnP [2] e o SimulES [6, 15] que são jogos de cartas físicas (i.e., jogados fisicamente). Além destes, os jogos SimulES-W [10] e SimSE [11] são propostas de jogos virtuais (i.e., jogados no computador). Estes quatro jogos são brevemente descritos nesta seção.

O PnP, *Problems and Programmers* [2], foi elaborado como um jogo de cartas que visa remover o foco da entrega de produtos. Assim, o objetivo do PnP é explicitar aspectos gerenciais de um processo de desenvolvimento de software: o Modelo Cascata [14]. Seguindo o Modelo Cascata, os jogadores têm contato com as principais etapas de desenvolvimento de software, desde a especificação de requisitos até a entrega do produto.

Inspirado pelo PnP, o jogo SimulES (Simulador de Uso da Engenharia de Software) [6] foi desenvolvido para solucionar diversos problemas identificados na versão original do jogo PnP. Por exemplo, conceitos de Engenharia de Software ultrapassados foram atualizados e contextualizados para a realidade de desenvolvimento de software no Brasil. Além disso, SimulES foi concebido para remover a dependência estrita do Modelo Cascata existente no PnP. Assim, conceitos de metodologias de desenvolvimento mais recentes, como métodos ágeis [3, 4], podem ser exercitados e aprendidos pelos jogadores de SimulES.

SimulES-W [9, 10] é a versão digital do SimulES num software implementado para Web, tendo ele incorporado os mesmos cenários do jogo de cartas. Esses cenários são apresentados no jogo virtual como telas através das quais os usuários navegam. Por

exemplo, o tabuleiro principal e os individuais são representados como telas, sendo que o primeiro é utilizado por todos. Cada tabuleiro individual pode ser utilizado somente pelo jogador responsável embora possa ser visualizado por todos. As cartas podem estar em jogo (todos visualizam) ou podem estar escondidas “nas mãos dos jogadores” para serem usadas em jogadas futuras. O SimuleS-W inclui a possibilidade de edição das cartas de problemas, de conceitos, de projetos e de engenheiros de software permitindo que o material a ser enfatizado no curso seja customizado pelo professor [10].

Diferente dos três jogos anteriores, o SimSE [11] é um jogo de único jogador e não segue a metáfora de jogo de cartas. Ao contrário, SimSE usa a analogia de um escritório em uma empresa de software no qual o jogador exerce o papel de um gerente de projetos de um time de desenvolvedores. O jogador tem a atribuição de gerenciar seus funcionários e tomar decisões que afetam diretamente o desempenho do projeto, tais como a aquisição de ferramentas, de treinamentos e a atribuição de tarefas aos funcionários. O desenvolvimento deste jogo foi feito buscando ensinar os principais conceitos de Engenharia de Software de uma maneira mais flexível. Assim, SimSE permite, por exemplo, que se escolha o modelo de processo a ser seguido. A versão atual do SimSE tem seis modelos de processos disponíveis [11]: Modelo Cascata, Incremental, Baseado em Inspeção, Prototipagem, RUP e Programação Extrema (XP).

Uma característica comum a todos estes jogos é o reconhecimento da distância entre a teoria aprendida em sala de aula e sua aplicação em uma situação real. Desta forma, o principal motivador na construção destes jogos é promover o ganho de experiência pelos alunos através da simulação de problemas reais de desenvolvimento de projetos de software.

3. Integração e Adaptação dos Jogos PnP e SimuleS

Apesar de alguns jogos, como o SimuleS-W e o SimSE (Seção 2), apresentarem algumas possibilidades para customizações (escolha do modelo de processo), ainda são raras as oportunidades para o professor customizar e adaptar os jogos existentes à sua filosofia de ensino. Neste sentido, os autores deste artigo mantêm discussões constantes sobre como tais jogos podem ser usados em diferentes momentos de uma disciplina de Engenharia de Software ou mesmo em outras disciplinas relacionadas. Por exemplo, além de Engenharia de Software, as universidades parceiras deste trabalho (UFMG, PUC-Rio e UERJ) oferecem disciplinas como Evolução de Software, Reuso de Software, Gerência de Projeto, Arquitetura de Software, Engenharia de Software Experimental, dentre outras. Todas estas disciplinas (e disciplinas semelhantes em outras universidades) podem, em maior ou menor grau, se beneficiar do uso dos jogos educacionais discutidos na Seção 2.

Entretanto, é importante que não só os professores participem desta discussão, mas que os alunos também colaborem ativamente na identificação de oportunidades para o uso de jogos educacionais. Neste sentido, uma discussão inicial foi fomentada envolvendo 23 alunos de pós-graduação em duas disciplinas da UFMG: Engenharia de Software e Reuso de Software. O objetivo desta discussão foi identificar características comuns e variáveis para modelar uma possível *linha de produtos de software* com base nas regras de diferentes jogos. Uma linha de produtos de software é um conjunto de aplicações definidas sobre uma arquitetura comum e que compartilham componentes

[16]. Os jogos considerados inicialmente foram o PnP e o SimulES, por possuírem regras semelhantes, o que facilita a identificação de características comuns.

Para direcionar as discussões em sala, foi adotado (Parte 1) um pequeno questionário abrangendo quatro perguntas e (Parte 2) um exercício prático para os alunos imaginarem um *modelo de características* [16] que modelasse a possível linha de produtos. As perguntas do questionário solicitavam que os alunos enumerassem as características que mais gostaram e as que menos gostaram nos dois jogos (PnP e SimulES). Além disso, outras duas perguntas pediam aos alunos para identificar as características que poderiam ser melhoradas ou removidas dos jogos. Por restrição de tamanho discutimos a seguir apenas alguns dos pontos mais interessantes que foram observados.

Em relação ao questionário, pode-se notar uma grande variedade de respostas, que creditamos à subjetividade das perguntas. Porém algumas características e pontos de variabilidade entre os jogos puderam ser identificadas, como ilustrado na Figura 1. A mesma apresenta um modelo parcial de características resultantes da discussão com os alunos das respostas das Partes 1 e 2 do estudo discutidas acima. Por exemplo, enquanto alguns alunos achavam que *cartas de problemas* e *cartas de conceitos* eram duas características essenciais dos jogos, outros sugeriram que estas cartas pudessem ser incluídas ou removidas dependendo do jogo. As cartas de problemas simulam problemas encontrados durante um projeto de software enquanto as cartas de conceito descrevem a aplicação de bons conceitos de Engenharia de Software. Desta forma, tais cartas foram representadas como características opcionais (círculo vazio) no modelo de características proposto para a linha de produtos (Figura 1).

Por outro lado, todos os 23 alunos que participaram do exercício foram unânimes em achar que as cartas de projeto de software são essenciais para ambos os jogos. A carta de projeto destes jogos representa uma solicitação do cliente, descrita em termos de requisitos, qualidade e orçamento. Esta característica foi representada como mandatória no modelo de características da linha de produtos (círculo preenchido). Outras características identificadas que não aparecem na Figura 1 estão documentadas em outro trabalho [1].

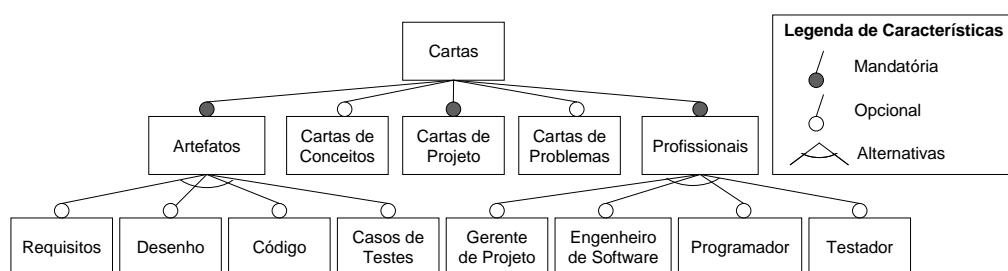


Figura 1. Modelo de características parcial com alguns pontos de variação nos jogos.

4. Experiência de Uso do SimulES-W

Um primeiro estudo de uso do recém criado jogo SimulES-W [9] foi realizado com um grupo de cinco alunos de graduação em Ciência da Computação da UERJ com conhecimentos básicos e interessados em aprender novos conceitos de Engenharia de Software. Um exemplo de tela do SimulES-W é apresentado na Figura 2 (a). Esse estudo foi realizado no dia 10 de dezembro de 2010 e teve os seguintes objetivos: (i)

analisar a interação dos alunos e do software no ambiente real, passando conceitos de Engenharia de Software e (ii) identificar pontos fortes e fracos do SimuleS-W, através da observação dos alunos e dos relatos de suas experiências com o jogo.



Figura 2. Exemplos de telas de dois jogos virtuais.

A primeira atividade realizada neste estudo foi um treinamento através de uma apresentação explicativa do jogo. A versão de tabuleiro (SimuleS físico [15]) foi usada para o treinamento e ilustração dos participantes. Após o treinamento, os alunos passaram a jogar SimuleS-W com o acompanhamento de dois instrutores. Finalizada esta atividade, foi aplicado um questionário, mostrado na Tabela 1, que permitiu receber retroalimentação dos alunos. Por exemplo, com a primeira pergunta do questionário procuramos evidências de que o jogo SimuleS-W consegue ser utilizado no ensino. Mas, verificamos que alguns aspectos de usabilidade ainda dificultam sua utilização.

Esta foi a primeira experiência de uso do jogo digital com alunos de um curso de Engenharia de Software embora alunos da PUC-Rio e da UERJ já tenham usado a versão em tabuleiro. As seguintes oportunidades de melhoria do SimuleS-W foram identificadas: (i) apresentar um menu disponível em todas as telas do jogo, (ii) evitar barras de rolagens, (iii) as mensagens entre jogadores devem aparecer em tempo real, (iv) habilitar as abas somente no momento certo, (v) tornar a interface gráfica toda na língua nativa do local, (vi) explicar melhor as cartas de conceitos e (vii) permitir que o manual de ajuda seja consultado durante o jogo. Estes pontos nos forneceram diretrizes para novos ciclos de melhorias e testes que foquem na clareza do uso do software e na qualidade do conteúdo a ser ensinado.

5. Experiência de Uso do SimSE

Foi realizado um estudo utilizando o jogo SimSE em duas turmas da disciplina de Engenharia de Software da UFMG. O objetivo deste estudo é verificar até onde o jogo pode apoiar o ensino, reforçar conceitos apresentados em sala de aula e apresentar novos conceitos. O SimSE (Figura 2b) foi escolhido devido à sua disponibilidade on-line e escalabilidade para turmas grandes, como é o caso. No total 60 alunos de graduação (16 na primeira turma e 44 na segunda) participaram deste estudo.

Foi realizado um estudo utilizando o jogo SimSE em duas turmas da disciplina de Engenharia de Software da UFMG. O objetivo deste estudo é verificar até onde o jogo

pode apoiar o ensino, reforçar conceitos apresentados em sala de aula e apresentar novos conceitos. O SimSE (Figura 2b) foi escolhido devido à sua disponibilidade on-line e escalabilidade para turmas grandes, como é o caso. No total 60 alunos de graduação (16 na primeira turma e 44 na segunda) participaram deste estudo.

Tabela 1. Questionário aplicado na experiência com SimulES-W

1.	De que maneira o jogo SimulES-W foi usável sob a óptica de um sistema de software? a) fácil de usar, b) usável, c) razoável, d) pouco usável, e) difícil usar
2.	O que poderia dizer da construção do produto de software no Tabuleiro Individual do SimulES-W, seu objetivo no jogo ficou claro?
3.	Pondo-se como um aluno, o jogo SimulES, no contexto do SimulES-W é: a) motivador, b) interessante, c) neutro, d) cansativo, e) nada a ver
4.	O que poderia dizer sobre as cartas problemas? O seu objetivo é claro?
5.	O que poderia dizer sobre as cartas conceito? O seu objetivo é claro?
6.	Pondo-se como um aluno, o jogo SimulES, no contexto do SimulES-W: a) ensina, b) informa, c) neutro, d) distrai, e) perturba
7.	Pondo-se como aluno, o jogo SimulES, no contexto do SimulES-W, é: a) deixa claro o papel da inspeção, b) lembra o conceito da inspeção, c) neutro, d) deixa dúvidas, e) confunde o aluno
8.	O que acredita é o ponto mais positivo do SimulES-W, e qual é o mais negativo?
9.	Que aspetos da navegação, aparência, execução das tarefas você melhoraria no SimulES-W?

Na primeira turma, o trabalho foi considerado como uma atividade extra-classe e os alunos fizeram um relato da experiência valendo pontos extras. Os alunos jogaram a versão do SimSE utilizando o Modelo Cascata e enviaram ao instrutor sua pontuação final e as respostas a um questionário com 15 perguntas abertas. O objetivo foi avaliar a opinião dos alunos sobre a utilização deste jogo como uma ferramenta de apoio ao aprendizado de Engenharia de Software. O questionário inclui questões sobre a avaliação da ferramenta (ex: facilidade de uso, suas características positivas e negativas) e sobre aprendizado (ex: ordem mais adequada para execução das tarefas, propósito de realizar inspeções durante o desenvolvimento do software).

Na segunda turma, os alunos jogaram em laboratório com acompanhamento de dois instrutores (o professor e o monitor da disciplina). Os alunos jogaram a versão do SimSE utilizando a Programação Extrema (XP) [3] e responderam o mesmo questionário da primeira turma. A pontuação final dos alunos que jogaram seguindo o Modelo Cascata ficou entre 27 e 99 pontos, com média de 76 e desvio padrão de 21,8. Já os alunos que jogaram pela XP tiveram pontuação entre 4 e 100 pontos, com média de 53 e desvio padrão de 17,5. Alguns alunos relataram que obtiveram uma nota consideravelmente baixa (zero em alguns casos) em sua primeira experiência com o jogo, mas conseguiram aumentar sua nota após sucessivas tentativas.

Nas respostas aos questionários, quase todos os alunos concordaram que o SimSE poderia ser incorporado à disciplina. Entretanto, alguns justificaram que isso não seria interessante por causa da simplicidade, principalmente em relação aos elementos de interface e repetição dos mesmos cenários de jogo. No mais, alunos relataram que algumas questões relacionadas à equipe, que não foram tratadas em sala de aula, puderam ser exercitadas durante o jogo como, por exemplo, os efeitos da motivação na produtividade e a importância da avaliação da experiência individual das pessoas para a alocação das tarefas. Como conclusão, verificamos que a incorporação do jogo à disciplina consiste em um recurso didático interessante para apresentar de forma lúdica questões que ocorrem em uma organização desenvolvedora de software, como alocação

da equipe, gestão de pessoas, gestão de alterações e alocação das tarefas de acordo com o processo utilizado.

6. Lições Aprendidas

Percebemos que, em todos os quatro jogos estudados, o treinamento e repetição de jogadas impactam no aprendizado dos alunos e na própria motivação em jogar. No SimSE, por exemplo, os alunos só conseguiram uma boa pontuação após várias tentativas. A utilização de elementos específicos dos jogos estudados (ex: pontuação e desafio), que representam certas características do mundo real, melhoraram a motivação e o comprometimento dos alunos. Em contrapartida, os jogos precisam ser mais dinâmicos e não prender o jogador em um ciclo de rodadas excessivamente lentas e desinteressantes.

Os jogos oferecem diferentes opções para o ensino de modelos de processos de software. Enquanto no PnP os jogadores devem seguir o modelo Cascata, no SimULES (e SimULES-W) eles são livres para seguir a metodologia preferida de desenvolvimento podendo, por exemplo, um jogador seguir o Modelo Cascata e seu adversário um método ágil. Entretanto, após algumas partidas fica claro no jogo que existe uma estratégia favorecida pelas regras que sugere um número mínimo de artefatos em cada fase de desenvolvimento. O SimSE oferece várias opções de processos mas, uma vez iniciado o jogo, ele deve ser jogado segundo um único processo.

A interface gráfica é um aspecto que deve ser bastante trabalhado nos jogos virtuais. Tanto no caso do SimULES-W como no SimSE, grande parte dos alunos apontou que questões de usabilidade, comunicabilidade e jogabilidade afetaram o seu desempenho e motivação. De forma análoga, a organização do tabuleiro e das cartas no SimULES e PnP também foi um dos fatores apontados como relevante para os jogos.

Nas perguntas sobre os jogos foram geralmente ressaltados os seguintes itens: (i) dificuldade no início da utilização de todos os jogos; (ii) forma divertida de lembrar de conceitos que foram aprendidos na sala de aula; (iii) os jogos são simples e geralmente têm objetivos claros; (iv) interessante a abordagem dos jogos de cartas, instigando e estimulando o aluno para problemas reais de Engenharia de Software usando uma metáfora intuitiva, (v) aleatoriedade do jogo pode ser um aspecto positivo em alguns casos, além da tomada de decisões corretas, e (vi) jogos são uma boa opção para completar o ensino de Engenharia de Software.

7. Conclusões

O ensino da Engenharia de Software convive com o problema de levar conceitos teóricos vistos em sala de aula para experiências práticas. Para suprir tal deficiência, podem ser utilizados jogos de apoio ao ensino, com foco em conceitos específicos. Este artigo relata nossa experiência de uso, em salas de aula de três universidades brasileiras, de quatro jogos existentes: PnP [2], SimULES [6], SimULES-W [9] e SimSE [11].

Pudemos perceber que os jogos contribuem para o aprendizado dos alunos, na medida em que eles mesclam a tarefa de aprender com o entretenimento proporcionado por um jogo. Este equilíbrio é capaz de motivar os alunos e melhorar o nível de interesse pela disciplina. Foi observado também que, mesmo apresentando um ganho significativo na motivação ao aprendizado, os jogos ainda necessitam de melhora,

principalmente nos quesitos de jogabilidade e usabilidade. A falta de coerência nas regras, a interface com o usuário e o dinamismo do jogo foram apontados como alguns dos principais problemas dos jogos avaliados.

Desta forma, este trabalho abre espaço para inúmeras oportunidades de trabalhos futuros, seja em melhorias futuras destes jogos visando torná-los mais agradáveis aos alunos, seja na metodologia de adoção dos jogos como ferramenta de ensino e aprendizagem.

Agradecimentos

Este trabalho recebeu apoio da PROGRAD/UFMG através do projeto PEG2011-84. Agradecemos Faber Andrade e Michael Dutra pelas discussões sobre PnP e SimulES.

Referências

- [1] F. Andrade. Estudo Comparativo de Ferramentas para Linha de Produtos de Software. Monografia de Conclusão de Curso em Ciências da Computação, Universidade Federal de Minas Gerais, 2011.
- [2] A. Baker, E. Navarro, and A. Hoek. An Experimental Card Game for Teaching Software Engineering Processes. *Journal of Systems and Software*, 75:1-2, 2005.
- [3] K. Beck, *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 1999.
- [4] M. Cohn. *Succeeding with Agile: Software Development Using Scrum*, Addison-Wesley, 2009.
- [5] L. Fernandes and C. Werner. Sobre o uso de Jogos Digitais para o Ensino de Engenharia de Software. FEES, SBES, Fortaleza, 2009.
- [6] E. Figueiredo, C. Lobato, K. Dias, J. Leite e C. Lucena. Um Jogo para o Ensino de Engenharia de Software centrado na Perspectiva de Evolução. *Workshop sobre Educação em Computação (WEI)*, Rio de Janeiro, p. 37-46, 2007.
- [7] C. Gresse von Wangenheim, D. Kochanski and R. Savi. Revisão Sistemática sobre Avaliação de Jogos Voltados para Aprendizagem de Engenharia de Software no Brasil. FEES, SBES, Fortaleza, 2009.
- [8] E. Kieling e R. Rosa. Planager - um Jogo para Apoio ao Ensino de Conceitos de Gerência de Projetos de Software. TCC em Ciência da Computação, FACIN, PUCRS, Porto Alegre, 2006.
- [9] E. Monsalve, V. Werneck, J. Leite. SimulES-W: Um Jogo para o Ensino de Engenharia de Software. FEES, SBES, Bahia, 2010.
- [10] E. Monsalve, V. Werneck, J. Leite. Teaching Software Engineering with SimulES-W. *Conf. on Software Engineering Education and Training (CSEE&T)*, 2011
- [11] E. Navarro and A. Hoek. SimSE: an Interactive Simulation Game for Software Engineering Education. *7th IASTED Int'l Conf. on Computers and Advanced Technology in Education*, 2004.
- [12] E. Neto. *Scrumming - Ferramenta Educacional para Ensino de Práticas do Scrum*. Pontifícia Universidade Católica do Rio Grande do Sul, 2008.
- [13] D. Peixoto, V. Batista, R. Resende, C. Pádua. Learning from Students' Mistakes in Software Engineering Courses. In *Proc. of Frontiers in Education Conference (FIE)*, 2010.
- [14] W. Royce. Managing the Development of Large Software Systems. *Proceedings of IEEE WESCON 26*, p. 1-9, 1970.
- [15] M. Serrano, M. Serrano, F. Napolitano, B. Soares. *Evolução do SimulES Versão 2.0*, Monografia em Ciências da Computação, Departamento de Informática. PUC-Rio, 2007.
- [16] D. Weiss and C. Lai. *Software Product-Line Engineering: A Family-Based Software Development Process*. Addison-Wesley Professional, 1999.