

Treinamento Experimental com Jogos de Simulação para Gerentes de Projeto de Software

Alexandre Ribeiro Dantas, Márcio de Oliveira Barros, Cláudia Maria Lima Werner
Programa de Engenharia de Sistemas e Computação
COPPE – Universidade Federal do Rio de Janeiro
Caixa Postal 68511 – CEP. 21945-970
e-mail: [alexrd, marcio, werner]@cos.ufrj.br

Resumo

O uso inadequado das técnicas de gerenciamento em projetos pode estar relacionado a uma ausência de estratégias eficientes de educação em gerenciamento de projetos de software, em que a motivação e o aprendizado experimental podem ser determinantes. Um processo de aprendizagem experimental requer um ambiente em que os estudantes possam atuar como gerentes sem os riscos e custos associados a um projeto real. O uso de simulações pode auxiliar neste processo, porém simuladores são limitados a uma representação muito abstrata dos ambientes reais de desenvolvimento de projetos. Neste artigo é proposto um jogo baseado em simulação que pode ser utilizado no aprendizado experimental para gerentes de projeto de software. O jogo é composto por um modelo em Dinâmica de Sistemas representando o projeto de software, um simulador de modelos e uma máquina de jogo que controla interações de usuário e os resultados da simulação por meio de uma interface gráfica típica de jogos. São apresentados estudos experimentais que avaliam um processo de aprendizagem baseado no jogo proposto.

Abstract

The inadequate use of project management techniques in software projects can be traced to the lack of efficient project management education strategies, where learning by experience and motivation are key issues. A hands-on learning process for project management requires an environment where students can act as managers without the costs and risks associated to real software projects. Simulation can support this process, but simulation tools lack the look-and-feel of a real project development environment. In this paper we propose a simulation-based game that can be used to provide hands-on learning to project managers. A System Dynamics model describing a software project, a simulator, and a game machine that handles user interactions and presents simulation results in a game-like fashion compose the game. We present experimental studies that evaluate an experiential learning process based on the proposed game.

1. Introdução

Segundo a literatura e pesquisas recentes [19], apesar de ser um conceito universal, a adoção de gerenciamento em projetos de software ainda é muitas vezes deficiente e inadequada. O alto número de projetos que, a cada ano, apresentam problemas de cronograma, orçamento e são cancelados pode ser consequência desta deficiência.

Uma das possíveis causas de um gerenciamento deficiente ou inadequado é a falta de compreensão das suas práticas e técnicas essenciais [5]. Geralmente, gerentes mais experientes obtêm mais sucesso na conclusão de projetos dentro de cronogramas e orçamentos planejados do que gerentes inexperientes. Isto porque o gerenciamento de projetos é uma atividade intensamente baseada em conhecimento. A habilidade e o

conhecimento tácito adquiridos durante diversos projetos anteriores permitem que o gerente mais experiente possa reagir melhor quando uma tomada de decisão é necessária. Esta necessidade de conhecimento e experiência contrasta com o cenário frequentemente observado em empresas dedicadas ao desenvolvimento de software, em que muitos gerentes de projeto ainda são promovidos de equipes técnicas devido ao seu sucesso em projetos anteriores, sem ter treinamento e educação apropriados para adquirir habilidades gerenciais [14].

Desta forma, o tempo geralmente investido nas atividades de gerenciamento é reduzido e inadequado. Nem sempre as medidas tomadas pelos gerentes, com base em sua intuição, para garantir o sucesso de um projeto fora de controle representam a solução mais adequada: o comportamento mais comum observado em projetos de software é o de "apagar incêndios". Dentre as medidas tomadas com frequência, podemos destacar aumento do cronograma, da equipe ou orçamento, pressão nos fornecedores, redução do escopo do projeto, mudança nas metodologias de desenvolvimento e da tecnologia utilizada, ou até mesmo o cancelamento do projeto.

A adoção de estratégias educacionais para preparar gerentes de projeto tem um papel importante na prevenção do uso inadequado das técnicas de gerenciamento em projetos de software, provendo a base para superar o cenário atual de projetos deficientes. Este artigo discute as estratégias de educação atuais em gerenciamento de projetos, suas deficiências e é considerada a aplicação de simulações e jogos para auxiliar o treinamento. Apresentamos um jogo, chamado *The Incredible Manager*, que foi utilizado em dois estudos experimentais para avaliar a hipótese a respeito da utilização de jogos em um programa de treinamento de gerentes de projeto. A estrutura do jogo e os resultados obtidos nos estudos experimentais são explorados neste artigo.

Este artigo está organizado em cinco seções. A primeira contém esta introdução. A próxima seção discute as deficiências apresentadas pela estratégia de ensino tradicional centrada no professor quando aplicada a cursos de gerenciamento de projetos de software e pesquisas que foram realizadas para complementar esta abordagem com outras ferramentas. A seção 3 apresenta o jogo que foi desenvolvido, sua estrutura e cada componente que a compõe em detalhes: o modelo de simulação, o simulador de modelos e a máquina de jogo. A seção 4 discute os estudos experimentais que foram executados para avaliar a utilidade do jogo em um conceito de treinamento. A seção 5 conclui o artigo apresentando as considerações finais.

2. Treinamento em gerenciamento de projetos de software

O treinamento da força de trabalho é uma necessidade crescente para atender às demandas de uma indústria cada vez mais competitiva. O mercado de treinamento foi estimado em mais de 100 bilhões de dólares no ano 2000 somente nos Estados Unidos, o que representa apenas uma parte do mercado mundial de treinamento, estimado em mais de 2 trilhões de dólares [18].

Uma das chaves para o sucesso educacional é a motivação e uma das melhores motivações para aprendizagem em gerenciamento de projetos é proveniente da experiência em atuar em projetos reais que falharam por gerenciamento insuficiente [7]. Ao analisar as situações passadas e avaliar diferentes caminhos que o projeto poderia ter tido caso decisões específicas tivessem sido tomadas em pontos particulares, um estudante pode melhorar sua habilidade em gerenciamento e na tomada de decisões. Este processo é semelhante à abordagem de estudo de casos desenvolvida no início do século XX e ainda atualmente aplicada no planejamento e administração organizacional [8].

Grande parte dos gerentes de projeto de software ainda são promovidos com base em habilidades técnicas, sem o treinamento apropriado para as novas responsabilidades. Mesmo aqueles que recebem algum treinamento, geralmente aprendem segundo estratégias educacionais tradicionais, centradas em conteúdo, que focalizam o que aprender ao invés de como aprender. O instrutor, nestes casos, é quem decide o que, quando e como o processo de aprendizagem será conduzido, geralmente por meio de aulas expositivas, livros e testes [18].

Duas características do gerenciamento de projetos de software devem ser consideradas contra a aplicação de abordagens educacionais centradas em conteúdo. Primeiro, deve ser destacado que apenas adultos são o público-alvo. Segundo, projetos de software de larga escala são complexos e seu comportamento pode não ser corretamente estimado por meio de análises baseadas em modelos mentais, percepção e intuição.

O treinamento adulto deve enfatizar mais o processo, menos o conteúdo e o instrutor [11]. Adultos precisam saber porque têm que aprender alguma coisa e aprendem melhor de forma experimental, quando o aprendizado se baseia na resolução de problemas e o tópico é de valor imediato. O aprendizado experimental e a motivação são, portanto, determinantes para uma melhor educação no gerenciamento de projetos.

Abordagens tradicionais de educação podem também não ser apropriadas uma vez que o gerenciamento de projetos depende fortemente de experiências e conhecimentos passados. Ao analisar uma decisão, o gerente procura em sua memória uma situação semelhante em outros projetos ou usa sua percepção para capturar a realidade atual e mentalmente predizer um estado futuro de acordo com as alternativas disponíveis [6]. Desta forma, esta abordagem requer que o gerente tenha experimentado situações similares no passado e tenha capacidade de percepção para relacionar as situações anteriores com os problemas enfrentados pelo projeto em desenvolvimento. Projetos de software de larga escala, entretanto, são caracterizados por dinâmica complexa e seu comportamento não pode ser eficientemente previsto por modelos mentais [20, 6].

3. Treinamento experimental

Uma vez que o comportamento de um projeto não pode ser derivado facilmente de princípios básicos (o conteúdo a ser aprendido), a abordagem educacional centrada em conteúdo deve ser complementada com mecanismos que suportem a aprendizagem experimental. A utilização de projetos-piloto é um exemplo destes mecanismos. Entretanto, é praticamente impossível criar diversos projetos reais para cada estudante devido a restrições de cronograma, orçamento e riscos associados ao fracasso de um projeto, mesmo piloto. Uma alternativa pode ser a adoção de modelos de simulação de projetos de desenvolvimento de software.

3.1. Treinamento baseado em simulações

O uso de simulação pode reduzir o tempo de treinamento, orçamento e riscos. Enquanto projetos reais podem durar semanas ou meses e suas falhas podem ter um alto custo, estudantes podem simular um modelo de projeto similar em poucas horas, focalizando sua atenção apenas em eventos relevantes durante a execução do projeto, segundo as metas educacionais previstas. Os modelos de simulação podem ser rapidamente analisados e configurados para representar diversas situações distintas que podem acontecer em projetos de larga escala, com longos cronogramas e equipes grandes, ao contrário de projetos-piloto.

O uso de simulação no treinamento em gerenciamento de projetos tem sido analisado por diversos estudos [12, 15, 17]. Pfahl et al. [17] apresentam um experimento controlado para avaliar a efetividade do uso de um modelo de simulação em educação. Em seu estudo, os participantes foram separados em dois grupos. Um grupo gerenciou um projeto de software com o auxílio de simulação. O segundo grupo atuou como grupo de controle, tendo no modelo COCOMO [4] sua ferramenta preditiva para o planejamento de projeto. Os resultados do estudo indicaram que o uso de simulações provê um melhor entendimento sobre padrões típicos de comportamento de projetos de desenvolvimento de software.

Recentemente, Barros et al. [2] conduziram dois estudos experimentais que ilustram algumas das limitações das simulações em metas educacionais. Os dois estudos consistiram na avaliação do uso de simulação no suporte à tomada de decisões em gerenciamento de projetos de software. Os participantes foram selecionados de duas diferentes universidades brasileiras (3 alunos de doutorado, 26 alunos de mestrado, 16 graduados e 4 alunos de graduação). Todos foram requisitados a gerenciar um pequeno projeto de forma a concluí-lo no menor tempo possível, atendendo a restrições de qualidade específicas.

Um emulador de projetos (um software que dita o comportamento do projeto ao longo do tempo) foi usado para representar o projeto proposto. Os participantes interagiram com o emulador, tomando decisões sobre quais desenvolvedores deveriam fazer parte da equipe, quantas horas de trabalho cada um faria por dia, se tarefas de inspeção deveriam ser incluídas no processo de desenvolvimento e quais desenvolvedores deveriam realizar cada tarefa do projeto. Metade dos participantes utilizou um simulador e modelos da Dinâmica de Sistemas [9] para analisar suas opções e avaliar suas decisões antes de aplicá-las no emulador de projetos. Os participantes restantes gerenciaram os projetos baseados apenas na sua experiência pessoal, sem o auxílio do simulador.

Os resultados dos estudos mostraram uma correlação positiva entre experiência dos participantes e sucesso no alcance dos objetivos do projeto, ou seja, gerentes experientes usando simulações parecem interpretar melhor os resultados do que gerentes inexperientes. Este resultado, embora positivo, não expressa o desejável, uma vez que o uso de simulações deveria ser capaz de oferecer relativamente maior auxílio para os gerentes inexperientes. Outra questão importante observada está relacionada com o engajamento: problemas no engajamento têm influência negativa no desempenho do participante. Destes estudos concluiu-se que para tornar a modelagem e a simulação mais úteis para gerentes inexperientes, é preciso buscar melhores formas de apresentação dos resultados da simulação, uma vez que é difícil para o analista do modelo rastrear os resultados observados para um comportamento intermediário visível do projeto

3.2. Treinamento baseado em jogos

Um dos problemas das ferramentas de simulação é a dificuldade de representar a sensação de estar em um ambiente de desenvolvimento de projetos real. Uma vez que a interação com a ferramenta não se assemelha a um ambiente real, a motivação do estudante pode estar sendo limitada. Ao utilizar simuladores, em geral, o estudante recebe números que, mesmo na forma de relatórios ou gráficos, são representações muito abstratas do que realmente está acontecendo, tornando o treinamento tão monótono quanto as formas pedagógicas tradicionais.

Um processo de aprendizagem experimental para gerenciamento de projetos requer um ambiente em que o estudante possa atuar como um gerente. Em uma situação artificial de aprendizagem, conforme visto, a motivação e o engajamento do estudante são determinantes. Alguns fatores especiais para esta motivação incluem realização pessoal,

desafio, vitória, recompensas, prazer e diversão. Os jogos são atividades em que as pessoas são absorvidas e parecem fluir, sem perceber a passagem de tempo na sua realização, de forma automática e espontânea, se sentindo livres no controle da atividade. Neste sentido, os jogos podem ser integrados aos modelos de simulação, adicionando fantasia, efeitos visuais e uma forma de interação mais atraente para os estudantes.

Alguns dos principais elementos encontrados em jogos incluem interação, flexibilidade, competição, *feedback* visual, efeitos dramáticos, usabilidade e grau de fidelidade à realidade [18]. Uma simulação, portanto, não pode ser considerada um jogo sem alguns destes seus elementos estruturais, em especial diversão e motivação, de forma a garantir o engajamento do participante no aprendizado.

De acordo com Prensky [18], a interatividade é uma característica fundamental, pois o jogador pode facilmente se entediar sem uma participação ativa. A flexibilidade permite ao jogador um nível de controle e investigação do conteúdo seguindo seu próprio caminho, de acordo com regras bem definidas. A competição representa o desafio, o conflito, a solução de problemas que requerem exploração e habilidades estratégicas que motivam o jogador na busca dos objetivos definidos. O jogo, por meio de *feedback* às ações do jogador, mede e apresenta o progresso deste na busca dos objetivos. Este *feedback* é feito por meio de efeitos dramáticos sobre os elementos que compõem uma história para o jogo. Estes efeitos (sons, imagens, animações, músicas, humor e histórias) relacionam entretenimento ao conteúdo do aprendizado, adicionando um contexto, personagens e dramatização, permitindo que o estudante tenha a chance de vivenciar um papel e atuar dentro do jogo. Para tanto, a usabilidade e o grau de realidade são fatores importantes e representam o quanto fácil e intuitivo é o jogo e como os diferentes elementos de mídia se relacionam para oferecer uma situação de proximidade com o mundo real nele representado.

Um mercado também crescente na indústria, estimado em trilhões de dólares, é o de entretenimento, especialmente jogos digitais, com vendas estimadas em 7 bilhões de dólares em 2002, apresentando um crescimento de 115% em relação ao ano de 1995 [18]. Segundo Prensky [18], a média de idade de um jogador norte-americano é de 29 anos, enquanto a média de idade da força de trabalho é de 39 anos. Adultos representam uma parte considerável do mercado de jogos de entretenimento digitais e à medida que funcionários mais antigos são substituídos por mais novos, as novas gerações de empregados terão sempre vivido em um mundo interativo, sob efeito das tecnologias digitais de entretenimento.

3.3. Efetividade do treinamento experimental

Alguns trabalhos atuais apresentam a adoção de conceitos de jogos na educação em engenharia de software, como a ferramenta SimSE [16] e o projeto SESAM [7]. Entretanto, a efetividade do uso de simulação e do aprendizado baseado em jogos é um ponto de discussão em aberto. Uma vez que os efeitos educacionais para diferentes abordagens são difíceis de serem isolados, medidos e rastreados, sua efetividade ainda não é bem documentada nem estabelecida [10]. Há muitos fatores intervenientes que devem ser levados em consideração em uma comparação entre abordagens diferentes, incluindo fatores subjetivos como a qualidade do professor em uma aula ou de um livro lido. Algumas abordagens podem ser mais apropriadas do que outras em determinadas situações e metas educacionais específicas. Ao invés de insistir em comparações entre abordagens, pode ser mais benéfico investigar comparações dentro de uma mesma abordagem, avaliando diferentes tipos de instrumentos, por exemplo. Este foi o princípio utilizado nos estudos experimentais relatados neste artigo, que utilizaram o jogo apresentado na próxima

seção no contexto de um processo de treinamento de gerentes de projeto de desenvolvimento de software.

4. O jogo *The Incredible Manager*

De forma a avaliar a abordagem do treinamento baseado em jogos, foi desenvolvido um jogo de simulação, chamado *The Incredible Manager*. Apesar da diversidade de estilos de jogos dificultar o estabelecimento de uma taxonomia bem definida, os jogos de aventura, quebra-cabeças e de simulação podem ser considerados os mais apropriados para metas educacionais de gerenciamento de projetos de software, destacando raciocínio, julgamento, tomada de decisões e pensamento sistêmico.

Ao utilizar o jogo, o estudante é requisitado a atuar como um gerente de projetos, planejando e controlando projetos de software para que estes sejam executados com sucesso, ou seja, dentro do cronograma e orçamento previstos. A construção do jogo é baseada em três elementos principais, como demonstra a Figura 1: um modelo de simulação, um simulador de modelos e uma máquina de jogo. Cada elemento é detalhado nas seções a seguir.

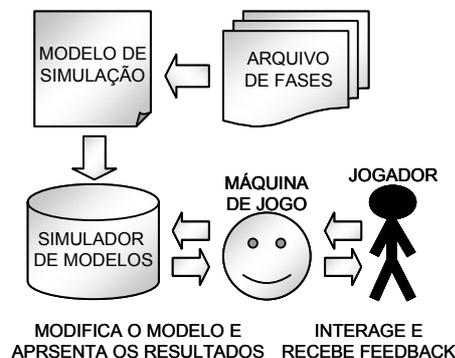


Figura 1 - Estrutura do jogo

5. O modelo de simulação

O modelo de simulação representa o mundo real e os aspectos que serão simulados e apresentados ao jogador. Diversas técnicas podem ser aplicadas para a construção de um modelo, porém os benefícios da simulação deste modelo dependem da capacidade destas técnicas representarem com fidelidade os aspectos do mundo real e da habilidade do construtor do modelo ao capturar estes aspectos. Os projetos de software, principalmente os de larga escala, são de difícil modelagem. Eles são classificados como sistemas de dinâmica complexa em função de características como [20]:

- Componentes inter-relacionados – Uma mudança em uma parte do sistema pode causar implicações em outras partes com um certo atraso, ou seja, sistemas desta categoria apresentam relações de causa e efeito distantes no tempo e no espaço.
- Comportamento dinâmico – Muitas informações importantes, como produtividade e taxa de geração de erros, variam ao longo do tempo. Além disso, as respostas a curto e longo prazo a um evento no sistema podem ser distintas.
- Ciclos de realimentação – O resultado de uma decisão pode afetar as condições que motivaram a própria decisão, de forma a reforçar ou corrigir as condições que a exigiram inicialmente.
- Relacionamentos não-lineares – Causas e efeitos nem sempre apresentam relações simples e proporcionais.

- Aspectos qualitativos – Um projeto de software não pode ser visto apenas em função dos componentes técnicos e suas relações, mas também deve considerar os aspectos socioculturais e qualitativos. Muitos destes aspectos não podem ser facilmente representados por números, mas podem influenciar no sucesso do projeto e não devem ser negligenciados.

A Dinâmica de Sistemas [9] é uma disciplina de modelagem baseada em uma visão holística para descrever e avaliar o comportamento visível apresentado por um sistema, de forma a confrontar as dificuldades encontradas em sistemas de dinâmica complexa. O comportamento é determinado pela estrutura de elementos que participam do sistema e os relacionamentos entre eles. Esta estrutura é descrita em um modelo por meio de equações matemáticas.

Esta disciplina de modelagem já foi utilizada no desenvolvimento de modelos para projetos de software [1] que se tornaram a base para revisões e extensões. Uma destas extensões é o paradigma do gerenciamento de projetos baseado em cenários [2]. Este paradigma apresenta um conjunto de técnicas de modelagem e simulação de modelos de projetos baseados na Dinâmica de Sistemas que prevê a separação entre incertezas e fatos conhecidos em modelos de projetos. Esta separação ocorre pela construção de modelos distintos (modelos de cenários) para cada aspecto incerto que pode influenciar um projeto de desenvolvimento de software.

Um metamodelo para a Dinâmica de Sistemas foi desenvolvido [2] para permitir a construção de um modelo de domínio, que define classes de elementos pertencentes ao domínio do problema, propriedades, comportamentos e relacionamentos que podem existir entre elementos destas classes. O Quadro 1 ilustra um trecho simplificado de modelo de domínio para o gerenciamento de projetos de software. O metamodelo permite a construção de modelos de projetos e modelos de cenários em um nível de abstração mais próximo dos conceitos do mundo real do que as equações matemáticas utilizadas tradicionalmente pela Dinâmica de Sistemas.

```
MODEL ProjectModel
{
  CLASS Developer
  {
    PROPERTY Expertise 0; # range [0, 1]
    PROPERTY HourlyCost 0; # in $
    PROC Productivity 1; # range [0, 1]
  };
  CLASS Activity
  {
    PROPERTY ExpectedDuration 0; # in days
    STOCK Cost 0;
    RATE (Cost) RTCost IF (Executing, Team.Cost, 0);
  };
  MULTIRELATION Precedences Activity, Activity (Successors);
  RELATION Team Activity, Developer;
};
```

Quadro 1- Exemplo de modelo de domínio¹

Com base no modelo de domínio, é possível definir um modelo operacional (modelo de projeto), em que são definidos os elementos de cada classe do modelo de domínio que participam do sistema que está sendo modelado. Os valores das propriedades de cada elemento são inicializados e os relacionamentos entre eles definidos. Por fim, o modelo de projeto é traduzido para as equações matemáticas da Dinâmica de Sistemas e pode ser

¹ Uma descrição detalhada da sintaxe da linguagem utilizada pode ser encontrada em <http://sety.cos.ufrj.br/riosim/>

simulado computacionalmente. O Quadro 2 ilustra um modelo de projeto para o modelo de domínio do Quadro 1.

```
DEFINE Exemplo ProjectModel
{
  Jimmy = NEW Developer
    SET Expertise = 0.4;
    SET HourlyCost = 20;
  Bobby = NEW Developer
    SET Expertise = 0.8;
    SET HourlyCost = 30;
  Design = NEW Activity
    SET ExpectedDuration 10;
    LINK Precedences Analysis;
    LINK Team Bobby;
  Coding = NEW Activity
    SET ExpectedDuration 25;
    LINK Precedences Design;
    LINK Team Bobby;
  ACTIVATE ProductivityLossDueCommunication
    CONNECT TheDeveloper Jimmy;
};
```

Quadro 2 - Exemplo de modelo de projeto

O Quadro 3 apresenta um cenário que, se aplicado a um desenvolvedor, determina o efeito do tamanho da equipe do projeto sobre sua produtividade individual. As duas últimas linhas do modelo apresentado no Quadro 2 ativam este cenário sobre o desenvolvedor Jimmy, que terá sua produtividade afetada pelo tamanho da equipe, ao contrário do desenvolvedor Bobby.

Os modelos, desta forma, podem ser mais facilmente construídos, modificados, integrados e expandidos para representar conhecimento de gerenciamento da literatura e da prática. Os modelos de cenário representam uma biblioteca genérica de eventos e teorias de gerenciamento que um instrutor pode integrar a um modelo de projeto para apresentar aos estudantes em uma determinada sessão de simulação com metas educacionais bem definidas. Utilizando simulações, é possível avaliar os impactos dos cenários desejados sobre o comportamento inicialmente esperado para o projeto. Para tanto, é necessário um simulador que execute estes modelos.

```
SCENARIO ProductivityLossDueCommunication ProjectModel
{
  CONNECTION TheDeveloper Developer
  {
    PROC COHFactor Count ([Developer]);
    PROC COHModifier LOOKUP (COHTable, COHFactor, 0, 30);
    TABLE COHTable 0, 0.015, 0.06, 0.135, 0.24, 0.375, 0.54;
    AFFECT Productivity Productivity * (1 - COHModifier);
  };
};
```

Quadro 3 - Exemplo de modelo de cenário

6. O simulador de modelos

O simulador de modelos é o elemento responsável por controlar os passos de simulação de um modelo, percorrendo iterativamente suas equações para calcular o comportamento de cada elemento do sistema. Ao contrário de simuladores genéricos, o simulador de modelos para um jogo de simulação deve ser interativo.

Ao utilizar simuladores genéricos, o jogador no papel de gerente deve preparar um plano (configurando as propriedades e relacionamentos dos elementos do modelo) e segui-lo até o final da simulação. Este processo de simulação, que denominamos como simulação com estrutura estática, é uma limitação da Dinâmica de Sistemas que, apesar de modelar

sistemas com comportamento dinâmico, não permite que a estrutura dos modelos, isto é, as relações entre seus elementos, seja dinâmica. A simulação com estrutura estática não representa fielmente a realidade: em um projeto de software o gerente toma diversas decisões ao longo do tempo - não apenas durante o planejamento -, modificando o plano original (e, portanto, a estrutura do modelo) para melhor controlar o projeto.

O simulador de modelos desenvolvido para o jogo é capaz de compilar um modelo construído segundo o metamodelo da Dinâmica de Sistemas para os construtores tradicionais da linguagem, simular o modelo compilado e processar eventos que alterem a sua estrutura durante a simulação. Desta forma, a simulação com estrutura dinâmica permite considerar as ações do jogador sobre a estrutura do modelo durante a execução do jogo, sem a necessidade de reavaliar o comportamento gerado pelos passos de simulação anteriores.

Como exemplo, consideremos que em um determinado instante durante o jogo uma tarefa de codificação esteja sendo executada por um desenvolvedor, porém o gerente perceba que não conseguirá cumprir o cronograma do projeto. Ele decide que deve contratar um novo desenvolvedor, mais experiente e trabalhando horas adicionais na tarefa. Tais decisões caracterizam três eventos sobre o modelo: a inclusão de um novo elemento (novo desenvolvedor), a alteração de propriedade (horas de trabalho) do elemento e a alteração de um relacionamento (desenvolvedor alocado na atividade de codificação). Uma vez interrompida a simulação e aplicados os eventos, o comportamento do modelo continua a ser simulado segundo a nova estrutura, a partir do ponto de interrupção e preservando todos os resultados obtidos previamente ao evento.

A simulação, entretanto, ainda é restrita a um modelo matemático da realidade, que não é facilmente interpretado pelos usuários. Para facilitar a interpretação e o entendimento dos resultados gerados pela simulação, torna-se necessária a construção de uma camada de mapeamento entre as equações do modelo e os conceitos do mundo real. Tal mapeamento é realizado pela máquina de jogo.

7. A máquina de jogo

A máquina de jogo é o elemento com que o jogador interage e recebe *feedback* visual dos resultados da simulação. A máquina é projetada para lidar com múltiplas fases seqüenciais, organizadas externamente em um arquivo de configuração do jogo. Cada fase representa um modelo de simulação distinto, oferecendo assim uma flexibilidade que permite a adoção de diferentes metas educacionais utilizando o mesmo jogo com diferentes modelos. O jogador inicia uma fase imediatamente após o término da anterior, mesmo que não tenha conseguido finalizá-la com sucesso. Durante uma fase, o jogador atua como gerente de projetos em um ambiente de desenvolvimento de software em que desenvolvedores (contratados pelo gerente) executam uma rede de tarefas (organizadas pelo modelo de simulação) com um cronograma planejado pelo jogador. Os personagens que são vistos no decorrer do jogo são:

- Gerente – O papel do jogador, responsável pelo planejamento do projeto e diversas tomadas de decisão.
- Desenvolvedores – A equipe que desenvolve o projeto. Cada um possui diferentes habilidades e características, como custo por hora e horas de trabalho diárias.
- Chefe – Representa todos os *stakeholders* do projeto, é responsável pela aceitação do plano de projeto e pressiona o gerente durante o desenvolvimento.

Cada fase do jogo é dividida em cinco etapas: Início do Projeto, Planejamento do Projeto, Aceitação do Planejamento, Execução do Projeto e Fim do Projeto. Cada uma das etapas é apresentada a seguir.

7.1. Início do Projeto

O início de uma fase apresenta o projeto a ser gerenciado pelo jogador. O documento de descrição do projeto inclui a descrição do produto a ser entregue, cenários especiais que podem impactar o desenvolvimento e algumas características do projeto: nome, organização e pontos de função das tarefas, demandas e restrições de qualidade, cronograma e orçamento.

7.2. Planejamento do Projeto

Nesta etapa o jogador deve construir um plano de projeto que contém a definição da equipe de desenvolvedores, cronograma e organização da rede de tarefas do projeto.

O jogador deve pesquisar e contratar desenvolvedores apropriados dentre os disponíveis no banco de currículos. Cada desenvolvedor, conforme mencionado anteriormente, possui características individuais. Uma vez definida a equipe, é preciso alocar desenvolvedores às tarefas do projeto. Cada tarefa só pode ser executada por um único desenvolvedor ao mesmo tempo e o jogador também deve definir o esforço (em número de dias) que ele deseja alocar para cada uma. O esforço em tarefas de qualidade (inspeções) é opcional: o jogador pode querer ou não realizar este tipo de tarefa.

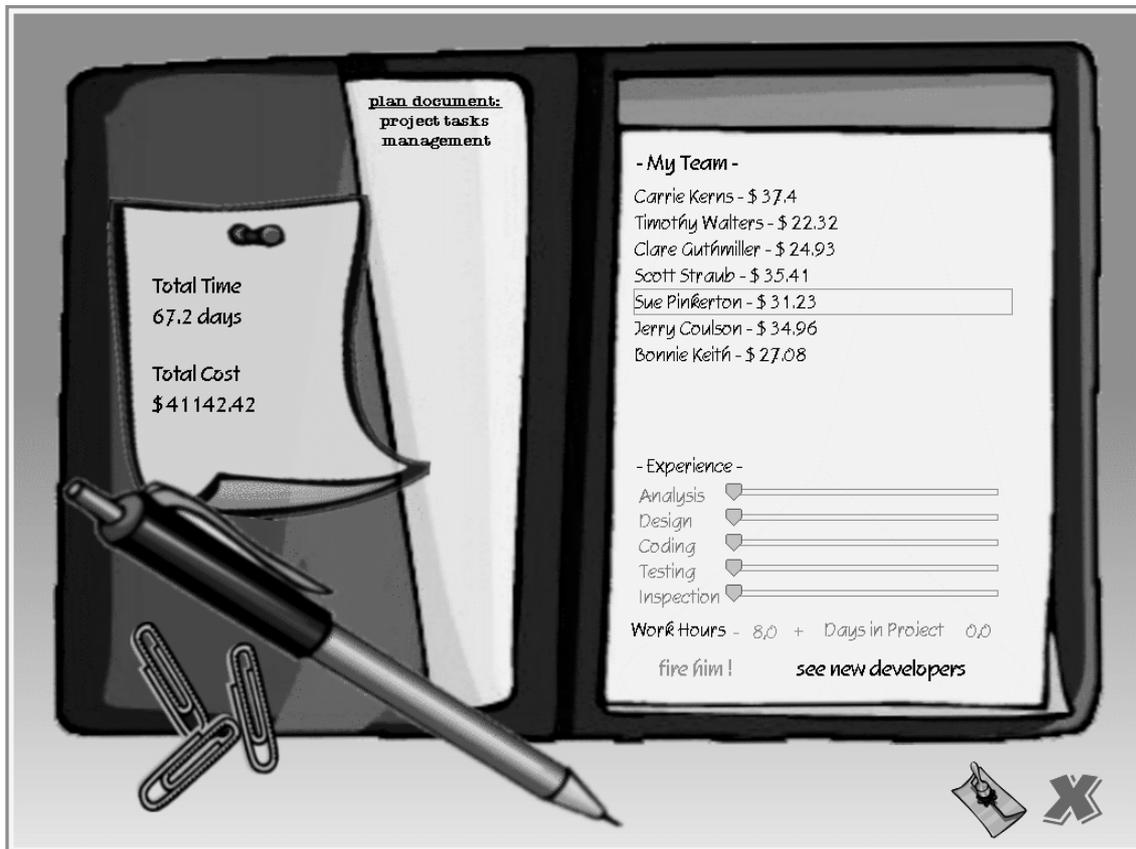


Figura 2 - Tela de planejamento de equipe

O plano de projeto pode ser modificado pelo jogador a qualquer instante durante a execução do jogo, demitindo e contratando desenvolvedores, modificando a carga horária de trabalho diária de cada um (de 8 a 12 horas de trabalho por dia), a duração das tarefas e os desenvolvedores alocados a cada uma delas. Ao final do planejamento, o plano de projeto exibe o esforço e o custo totais estimados pelo jogador para desenvolver o projeto. A Figura 2 apresenta a tela de planejamento de equipe, em que o jogador pode visualizar os desenvolvedores e suas características.

7.3. Aceitação do Planejamento

Uma vez tendo feito o plano de projeto, este deve ser enviado pelo gerente para avaliação do chefe. O plano pode ser aprovado ou não. Um plano é recusado se suas estimativas estiverem muito acima das restrições descritas no documento de apresentação do projeto. Se o plano for recusado, o jogador deve refazer o planejamento, ajustando suas estimativas de dias e orçamento até que o plano seja aprovado com sucesso. O jogador, desta forma, deve tomar importantes decisões de custo/benefício na escolha, alocação da equipe e duração de tarefas para conseguir atender às demandas propostas pelo jogo.

7.4. Execução do Projeto

Durante esta etapa, a rede de tarefas do projeto é executada segundo o plano de projeto aceito. A Figura 3 exibe o ambiente de trabalho em que o desenvolvimento acontece. As mudanças gráficas dos personagens são variações nos desenhos dos desenvolvedores e balões de pensamento. Cada desenvolvedor transita pelo escritório, indo, vindo e sentando na sua mesa de trabalho, durante o decorrer de um dia. Os balões indicam se ele está ocioso, cansado ou em pânico. Um desenvolvedor entra em pânico caso o projeto esteja atrasado, sem fundos, sem tempo ou a tarefa que ele está executando esteja atrasada. Um personagem especial, o chefe, aparece para fazer reclamações quando o projeto está atrasado, sem fundos ou sem tempo.

Uma outra forma de indicar as alterações de estados do modelo é pela emissão de relatórios. Um relatório é emitido quando um desenvolvedor começa ou termina uma tarefa, percebe que a tarefa que ele executa está atrasada e quando ele se sente cansado por trabalho excessivo. Caso o chefe perceba que o projeto está atrasado, sem fundos ou sem tempo hábil para conclusão, também são emitidos relatórios para alertar o gerente.

Na parte inferior da tela de execução, estão os recursos restantes ao projeto (dias e fundos). A coloração em vermelho indica se há deficiência em algum deles. Os recursos decrescem a cada instante de tempo, uma vez que a execução do projeto acontece em turnos contínuos (porém ajustáveis pelo jogador). Com a passagem ininterrupta de tempo, o jogador deve ficar atento ao comportamento do projeto e da equipe, de forma a tomar ações corretivas quando necessárias para controlar o projeto e garantir que ele seja concluído com sucesso. Para evitar o fracasso, o jogador pode precisar modificar o plano de projeto original durante o desenvolvimento. Desta forma, diferentes decisões tomadas por diferentes jogadores em diferentes circunstâncias fazem do jogo uma experiência não-linear, permitindo que estudantes vivenciem a experiência de gerenciar o mesmo projeto em diversos caminhos.

7.5. Fim do Projeto

A fase termina quando os recursos disponíveis para o projeto (dias e fundos) acabam sem que todas as tarefas previstas para o projeto tenham sido concluídas. Neste caso, a fase e o

projeto foram finalizados com fracasso. Quando todas as tarefas são concluídas dentro das estimativas de recursos previstas pelo planejamento, o projeto é finalizado com sucesso.

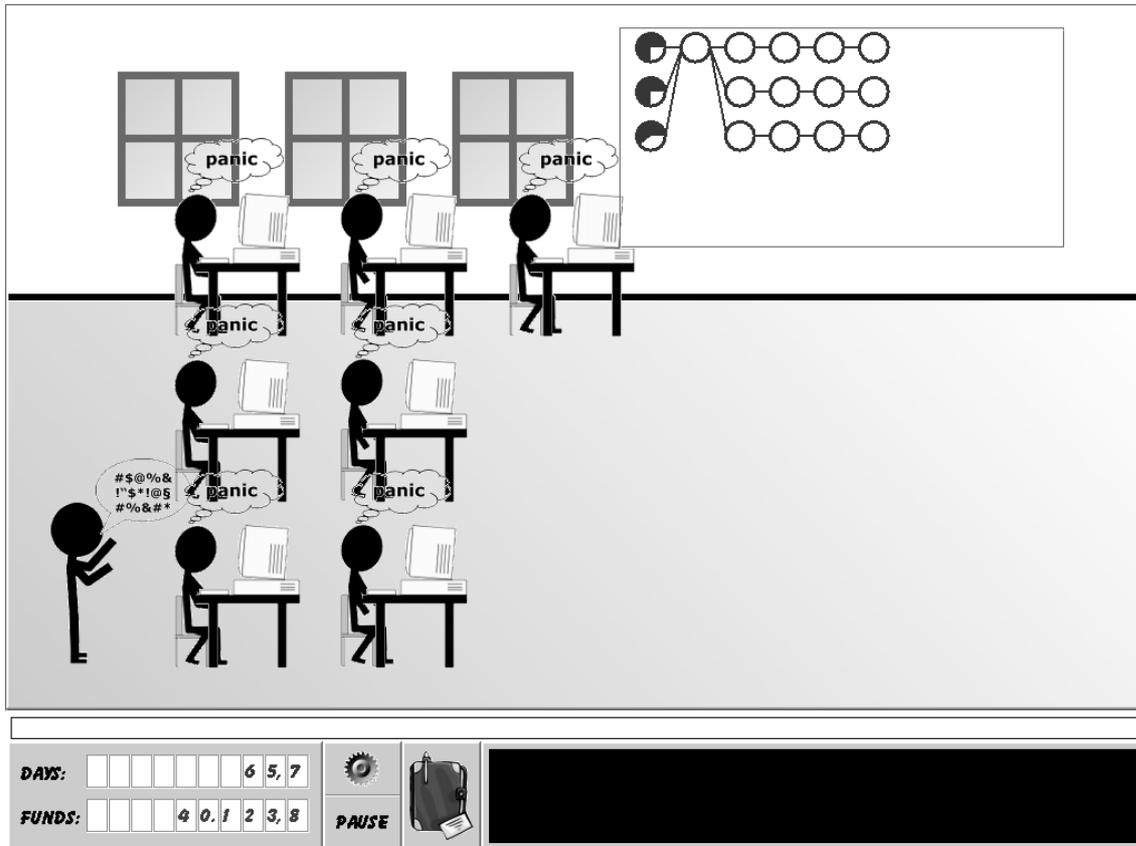


Figura 3 - Tela de execução

8. Avaliação do treinamento baseado em jogos

Dois estudos de caso experimentais foram conduzidos para avaliar a adoção do jogo *The Incredible Manager* em um processo de treinamento. O Quadro 4, baseado no modelo GQM [3], descreve o objetivo dos estudos experimentais.

*Analisar a utilização de um jogo de simulação dentro de um conceito de treinamento
 Com o propósito de caracterizar a utilidade do jogo e continuidade do desenvolvimento
 Referente aos ganhos obtidos por sua utilização e as percepções sobre sua utilização
 Do ponto de vista do pesquisador
 No contexto do gerenciamento de projetos de desenvolvimento de software.*

Quadro 4 - Objetivo dos estudos experimentais

O processo de treinamento utilizado nos dois estudos foi baseado no processo proposto por Mandl-Striegnitz [13], que divide o treinamento em duas sessões de simulação e duas sessões de discussão intercaladas, complementando o aprendizado que tradicionalmente seria apoiado apenas por livros-texto. Em um estudo realizado com base neste processo de treinamento, os participantes apresentaram melhora de desempenho na segunda sessão de simulação em relação aos aspectos capturados na primeira sessão de discussão. Na avaliação qualitativa do treinamento, os participantes demonstraram entendimento sobre as funções de gerenciamento de projetos e as dificuldades de gerenciar um projeto de software com sucesso. Eles consideraram as sessões de discussão como o ponto crucial do

processo de treinamento estabelecido. O procedimento planejado para execução dos estudos é resumido a seguir:

- a) Sessão de treinamento;
- b) preenchimento do questionário quantitativo (formação, experiência);
- c) primeira sessão de simulação;
- d) primeira sessão de discussão;
- e) segunda sessão de simulação;
- f) segunda sessão de discussão; e
- g) preenchimento do questionário qualitativo.

O procedimento não foi aplicado totalmente por limitações de tempo: apenas uma sessão de simulação e uma sessão de discussão foram aplicadas. Assim, não houve a repetição prevista para estas sessões. Inicialmente, os participantes foram treinados em conceitos importantes em gerenciamento (como estimativas por pontos de função) e na utilização do jogo *The Incredible Manager*. Após o treinamento, os participantes foram requisitados a jogar individualmente uma fase do jogo (sessão de simulação). Quando todas as simulações terminaram, um instrutor e todos os participantes se reuniram para uma sessão de discussão. O instrutor apresentou cenários e abordagens comuns em gerenciamento de projetos segundo conhecimento da prática e da literatura. Apresentando exemplos positivos e negativos para situações específicas de tomada de decisão com que os participantes se confrontaram durante o jogo, foi permitido a estes melhor interpretar suas ações e desempenho durante a sessão de simulação. Os participantes também puderam relatar suas percepções e sensações no papel de gerente de projetos do jogo.

O primeiro estudo foi conduzido com 7 participantes (1 aluno de doutorado e 6 alunos de mestrado) de uma turma de um curso de gerenciamento de projetos de software em um curso de pós-graduação de uma universidade federal brasileira na região sudeste do país. O segundo estudo foi conduzido com 8 participantes (6 alunos de mestrado e 2 alunos de graduação) de um laboratório de desenvolvimento de software de uma indústria conveniada a uma universidade brasileira na região sul do país. Em questionário quantitativo preenchido antes da primeira sessão de simulação, cada participante avaliou sua experiência em desenvolvimento de software, habilidade e interesse em gerenciamento de projetos a partir de uma escala variando de zero (nenhum) a cinco (muito). A Tabela 1 apresenta os valores médios indicados pelos participantes em cada estudo e a média para ambos.

Tabela 1 - Características médias dos participantes

Característica do participante	média geral	estudo 1	estudo 2
Experiência em desenvolvimento de software	3,5	3,1	3,8
Habilidade em gerenciamento de projetos	1,9	1,7	2,0
Interesse em gerenciamento de projetos	4,3	4,3	4,3

A sessão inicial de treinamento dos participantes durou aproximadamente 20 minutos nos dois estudos. A primeira sessão de simulação durou entre 50 e 120 minutos (respectivamente, o tempo gasto pelo primeiro e pelo último participante para terminar a simulação) no primeiro estudo e entre 55 e 140 minutos no segundo o estudo. Apesar do número total de pontos de função do projeto ser mantido entre os dois estudos, o número de tarefas do projeto no segundo estudo foi reduzido de 20 para 15, na tentativa de diminuir a duração da sessão de simulação e viabilizar a realização de uma segunda sessão. Tal tentativa, entretanto, não surtiu efeito.

Com base nos históricos de ações dos jogadores capturados pelo jogo durante a sessão de simulação, a Tabela 2 apresenta exemplos de algumas médias sobre a tomada de decisões e ocorrência de determinados estados que podem ser coletados durante o jogo e utilizados para futuras análises sobre o treinamento. Possíveis discrepâncias entre os valores em uma comparação entre os dois estudos podem estar relacionadas com a redução da rede de tarefas de 20 para 15 tarefas, na execução do segundo estudo.

Tabela 2 - Médias de tomada de decisões e estados durante o jogo

Decisões/Estados	estudo 1 (# vezes)	estudo 2 (# vezes)
Contratação de desenvolvedor	12,4	10,8
Demissão de desenvolvedor	5,6	4,2
Mudança de alocação de desenvolvedor	7,4	8,4
Replanejamentos aceitos	12	18,6
Desenvolvedores cansados	1	2,2
Desenvolvedores exaustos	0,4	0,2
Tarefas extrapolaram o tempo	3,2	3,4
Tempo em tarefas de inspeção (em dias)	4,24	1,7

Apenas um dos participantes concluiu o jogo com sucesso (no primeiro estudo). Apesar do pouco sucesso na conclusão da fase do jogo, os relatos dos participantes foram considerados muito positivos. O processo de treinamento baseado em jogos de simulação foi considerado motivante, dinâmico, prático e divertido. Os participantes apontaram outros aspectos importantes como pressões psicológicas (da passagem contínua de tempo e dos efeitos visuais apresentados), alto grau de dificuldade como um desafio motivante para o jogador e o entretenimento obtido durante a execução do treinamento sem a perda de engajamento na busca das metas propostas pelo jogo. A Tabela 3 resume alguns dos resultados obtidos em questionário qualitativo preenchido pelos participantes nos dois estudos, após a sessão de discussão.

Tabela 3 - Avaliação do treinamento baseado em jogos

	aumentou	foi indiferente	reduziu	
A habilidade em gerenciamento de projetos após o treinamento	100%	0%	0%	
O interesse em gerenciamento de projetos após o treinamento	87%	13%	0%	
	boa	foi indiferente	ruim	
A avaliação do treinamento baseado em jogos de simulação	100%	0%	0%	
	todas	nenhuma	muitas	poucas
As lições apresentadas no treinamento foram aprendidas ?	0%	0%	100%	0%
	foi	foi muito	foi pouco	não foi
O treinamento foi considerado divertido ?	47%	53%	0%	0%

As principais limitações relatadas pelos participantes foram relacionadas à simplicidade do modelo de simulação para projetos de software. O modelo atual é incapaz de representar diversas situações e eventos inesperados do mundo real, como múltiplos desenvolvedores trabalhando juntos em uma mesma tarefa, demissões, interações sociais, questões psicológicas de desenvolvedores, questões organizacionais, entre outras. Esta simplificação está relacionada à dificuldade de obtenção de conhecimento formalizado para descrever tais situações e eventos no modelo de simulação. Alguns participantes demandaram também a adoção de uma interface multi-usuário, facilidades de ensino a distância e algum

tipo de ferramenta de rastros explicativa para as ações, conseqüências, lições aprendidas e rotas alternativas para as tomadas de decisões do jogador durante o jogo. Este tipo de ferramenta ajudaria os usuários a avaliar seu próprio desempenho depois da execução do jogo. Tais aspectos serão futuramente explorados no contexto desta pesquisa.

9. Considerações finais

Neste artigo, foi analisada a adoção de mecanismos práticos para complementar as estratégias tradicionais de educação centradas no instrutor e no conteúdo. O foco atual do trabalho se baseia no treinamento de gerentes de projeto de software, uma vez que a falta de conhecimento sobre as técnicas de gerenciamento e o uso inadequado destas técnicas podem ser considerados fatores determinantes que inibem o sucesso de um projeto.

A introdução de jogos de simulação pode ser um mecanismo benéfico para uma situação de aprendizado experimental, como a necessária para estudantes de gerenciamento de projetos. Os jogos oferecem aos estudantes a oportunidade de experimentar as conseqüências de executar ou negligenciar importantes funções de gerenciamento, confrontar-se com questões complexas que devem ser resolvidas durante o decorrer do desenvolvimento do projeto, testar diferentes abordagens e soluções, aprendendo e observando suas conseqüências.

Com a evolução dos modelos de simulação, as limitações atuais do jogo proposto serão diminuídas, representando situações mais reais do dia-a-dia do desenvolvimento de projetos de software, aumentando, o número de cenários de treinamento e refinando a transferência de conhecimento. As dificuldades de construção de modelos formais abrem uma demanda para ferramentas gráficas que aumentem o nível de abstração destes modelos, tornando a construção mais intuitiva e flexível. O simulador de modelos e máquina de jogo propostas são flexíveis para lidar com diferentes modelos desenvolvidos a partir do metamodelo de gerenciamento de projetos existente. Entretanto, para acompanhar a evolução do metamodelo (e a construção de modelos em outros domínios), a máquina de jogo apresentada pode ser estendida para representar mais transições de estados e maior *feedback* gráfico, enriquecendo a percepção e o entretenimento do jogador.

Além do modelo de simulação, muitas outras áreas de pesquisa podem ser destacadas no caminho de trabalhos futuros: evoluções pedagógicas sobre o processo de treinamento baseado em jogos, evoluções artísticas sobre a usabilidade e apresentação multimídia do jogo, pesquisas sobre rastros das ações do jogador, suas conseqüências, desempenho, caminhos alternativos e pesquisas psicológicas sobre aspectos cognitivos e motivacionais relacionados à aprendizagem e ao treinamento baseado em jogos.

Agradecimentos

Os autores agradecem à CAPES o auxílio financeiro.

Referências

1. ABDEL-HAMID, T.K., MADNICK, S.E., Software Project Dynamics - An Integrated Approach, Prentice-Hall, Englewood Cliffs, 1991.
2. BARROS, M.O.; WERNER, C.M.L.; TRAVASSOS, G.H., Supporting Risk Analysis on Software Projects. In: The Journal of Systems and Software, v. 71, n. 1-2, p. 21-35, 2004.

3. BASILI, V.R., CALDIERA, G., ROMBACH, H.D., “Goal Question Metric Paradigm”, IN: Marciniak, J.J., Encyclopedia of Software Engineering, vol. 1, New York, NY: Wiley, p. 528-532, 1994.
4. BOEHM, B.W., ABTS, C., BROWN, W.A., CHULANI, S., CLARK, B.K., HOROWITZ, E., MADACHY, R., REIFER, D.J., STEECE, B., Software Cost Estimation with COCOMO II, Upper Saddle River, Prentice Hall PTR, 2000.
5. BROWN, N., “Industrial-Strength Management Strategies”, IEEE Software, v. 13, n. 4, p. 94-103, 1996.
6. DOYLE, J.K., FORD, D.N., RADZICKI, M.J., TRESS, W.S., Mental Models of Dynamic Systems [online, 01/07/2003], <http://www.wpi.edu/Academics/Depts/SSPS/Faculty/Papers/27.pdf>.
7. DRAPPA, A., LUDEWIG, J., Simulation in Software Engineering Training. In: Proceedings of the International Conference on Software Engineering, Limerick, Ireland, p. 199-208, June 2000.
8. FORRESTER, J.W., System Dynamics and the Lessons of 35 Years, [online, 05/02/2004] <http://sysdyn.clexchange.org/sdep/papers/D-4224-4.pdf>, 1991.
9. FORRESTER, J.W., Industrial Dynamics, Cambridge, MA: The MIT Press, 1961.
10. GRÖBLER, A., NOTZON, I., SHEHZAD, A., Constructing an Interactive Learning Environment (ILE) to Conduct Evaluation Experiments. In: Proceedings of the 1999 Conference of the International System Dynamics Society, Wellington, New Zeland, July 1999.
11. KNOWLES, M., Andragogy in Action, Jossey-Bass, San Francisco, CA, 1984.
12. MAIER, F.H, STROHHECKER, J., “Do Management Flight Simulators Really Enhance Decision Effectiveness”. In: Proceedings of the 1996 Conference of the International System Dynamics Society, Cambridge, USA, July 1996.
13. MANDL-STRIEGNITZ, P., “How to Successfully Use Software Project Simulation for Educating Software Project Managers”. In: Proceedings of the 31st Frontiers in Education Conference, Reno, Nevada, USA, October 2001.
14. MANDL-STRIEGNITZ, P.; LICHTER, H., A Case Study on Project Management in Industry: Experiences and Conclusions. In: Proceedings of the European Software Measurement Conference (FESMA 98), p. 305-313, Antwerp, Belgium, May 1998.
15. MERRIL, D., Training Software Development Project Managers with a Software Project Simulator, Master Thesis Proposal, [online, 01/07/2003], <http://www.eas.asu.edu/~sdm>, Arizona State University, Tempe, AZ, USA, 1995.
16. OH, E., VAN DER HOEK, A., “Towards Game-Based Simulation as a Method of Teaching Software Engineering”. In: Proceedings of the 2002 Frontiers in Education Conference, Boston, MA, USA, November 2002.
17. PFAHL, D., LAITENBERGER, O., DORSCH, J., RUHE, G., “An Externally Replicated Experiment for Evaluating the Learning Effectiveness of Using Simulations in Software Project Management Education”, Empirical Software Engineering, v. 8, p. 67–395, 2003.
18. PRENSKY, M., Digital Game-Based Learning, McGraw-Hill, 2001.
19. STANDISH GROUP, The Chaos Chronicles. The Standish Group International, 2003.
20. STERMAN, J.D., System Dynamics Modeling for Project Management. MIT System Dynamics Group, Cambridge, MA, USA, 1992.