

A Simulation-Based Game for Project Management Experiential Learning

Alexandre Dantas, Márcio Barros, Cláudia Werner
COPPE / UFRJ – System Engineering and Computer Science Department
Caixa Postal: 68511 – CEP: 21945-970 – Rio de Janeiro – Brazil
{alexrd, marcio, werner}@cos.ufrj.br

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. An experiential learning process for project management requires an environment where students can act as managers without the costs and risks associated to an unsuccessful software project. 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 experiential 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. System Dynamics limitations to support a game-like user interface are discussed. Also, we present an experimental study that evaluates an experiential learning process based on the proposed game.

1. Introduction

Project management can be considered an universal concept, but according to the software engineering literature and recent researches [3][11][16] its adoption in software projects is still inadequate and deficient. The high number of software projects that are cancelled each year and the number of project presenting schedule and cost overruns [3][16] may be consequences of this lack of project management.

It is widely accepted that experienced project managers perform better than inexperienced managers in concluding their projects successfully, that is, within their planned schedules and budgets. Project management is strongly dependent on knowledge and still many project managers are promoted from technical teams due to their successes in previous projects without proper training and education to acquire management skills [11].

Thus, education strategies adopted to prepare project managers play an important role in preventing from inadequate use of management techniques on software

projects, providing the basis to leverage the present scenario of so many faulty projects.

This paper discusses current project management education strategies and their deficiencies. We consider the application of simulation and games to support management training. We have developed a game, namely *The Incredible Manager*, and used it in two experimental studies to evaluate our hypotheses concerning the usefulness of games to a project management training program. This paper presents the game structure and results obtained from the studies.

The paper is organized in eight sections. The first one comprises this introduction. The next section discusses the deficiencies presented by the traditional professor-centric education strategy when applied to project management courses and some research that has been made to complement this approach with other tools. Section 3 presents the game that we have developed and its architecture. The following sections present the major components that compose this architecture: Section 4 presents the simulation model, Section 5 presents the simulation model, while Section 6 describes the game machine. Section 7 discusses the experimental studies that were executed to evaluate the game usefulness. Section 8 concludes the paper by presenting our final considerations.

2. Experience in Education

One of the keys for educational success is motivation and one of the best motivations for learning project management comes from taking a role in real projects that failed due to insufficient management [5]. By analyzing past situations and evaluating a different path that the project could have taken if specific decisions were made in particular points, a student can enhance his management skills and ability to make decisions. This is similar to the *case study approach* that was developed early in the 20th century and is currently applied in organization planning and administration [6].

However, most of our current project managers are developers who were promoted considering technical

skills, without proper training to assume their new responsibilities. Even those who receive some training usually learn through traditional educational strategies, which are content-centric: they focus on “what to learn” instead of “why to learn”. The instructor decides what, when and how learning will be conducted, usually by using classes, textbooks and tests [15].

Two characteristics of software project management present difficulties to the application of the content-centric educational approach. First, it should be noted that only adults undertake project management: so, project management training is adult training. Second, large-scale software projects are complex elements and their behaviour is often too complex for mental analysis.

Concerning adult training, pedagogical studies [9] have shown that the content-centric approach is not adequate for adult learning, since adults prefer to learn based on experience and learn better when they can apply to solve their current problems. Thus, learning by experience and motivation are key issues for better management education.

Concerning complexity, the traditional education approach may not be adequate since project management strongly depends on past experiences and knowledge. While analyzing a decision, a manager usually seeks in his memory for a similar situation in other projects or uses his perception to capture current reality and mentally predict its future state according to available alternatives [4]. This approach requires that the manager has experienced similar situations in the past.

In the learn-by-error approach to management training (implicitly taken when no formal management education is provided to novice managers), this experience comes from participating in failing projects. In the case-study approach, this experience comes from creating and analysing descriptive models of software projects.

However, large software projects are characterized by dynamic complexity in the form of feedback loops, delays, and cause-and-effect relationships distant in time. Their behaviour cannot be efficiently predicted by mental models [17]. Such interpretation often leads project managers to wrong decisions. Since project behaviour cannot be easily derived from basic principles (the content to be learned), the content-centric approach must be complemented by mechanisms that support experiential learning.

Mentoring novice managers through pilot-projects is an example of such mechanisms. However, it is rarely possible to create real projects for manager trainees due to practical constraints on schedule, budget, and risk.

Another alternative could be the adoption of simulation models of software development projects. Simulation can reduce training time, budget and risks. While real projects can last for months and their failures may have a high cost, students can simulate a similar

project model in a few hours, focusing their attention on relevant events occurring throughout the project execution and hiding the details that may confuse the trainee while learning a major lesson. Simulation models can be quickly analyzed and configured for several distinct development situations that could only happen in large projects, with long schedules and large teams.

The use of simulation to support project management education has been analyzed by several studies [10][12]. In a recent paper [14], Pfahl et al. present a controlled experiment to evaluate the effectiveness of using a simulation model in education. In this study, subjects were separated in two groups. One group managed a software project with the aid provided by a simulation model. The second group acted as a control group, using the COCOMO model as a predictive tool for project planning while the experimental group used a simulation model. The results of the study indicate that the use of simulation models provides a better understanding about typical behaviour patterns of software development projects. However, the unique use of simulation models is insufficient to project management education. Simulation is usually a predictive approach: models try to capture some specific real world issues so as simulation can present good insights about the results obtained from particular decisions made. Results are mostly represented by numbers or graphics that are abstract representations of what is really happening within the model during the simulation.

Recently, we have conducted two experimental studies that illustrate some simulation drawbacks for educational goals [2]. Both studies consisted of evaluating the use of simulation to support decision-making on software project management. Subjects were students from two different universities (3 D.Sc. students, 26 M.Sc. students, 16 B.Sc. students and 4 B.Sc.). They were asked to manage a small project with a major objective of concluding it in the lowest schedule as possible, while attending to specific quality restrictions.

A project emulator (that is, a software that dictates project behavior overtime) was used to represent the proposed project. Subjects interacted with the emulator, making decision about which developers should take part in the project team, how many hours should each developer work per day, if inspections should be included in the development process, and which developer should accomplish each project activity. Half of the subjects used System Dynamics models [7] and simulation to analyze their options and evaluate their decisions before applying them in the project emulator. The remaining subjects managed the project based on their own experience, without the aid provided by the simulator.

The results of these studies show us positive correlation between subject experience, interpretation difficulties and success in attending to project objectives.

This was an unexpected result because modeling and simulation were supposed to provide more help to inexperienced managers. Another important issue observed relates to subject engagement. The lack of engagement had negative influence over the subjects' performance. To make modeling and simulation more useful for inexperienced managers, we shall look for better ways to present simulation results. It is usually difficult for a model analyst to trace model observed results to intermediate behaviour.

A problem with simulation tools is their lack of a real project development environment look-and-feel. Since the interaction with the project environment does not resemble a real situation, student's motivation can be limited while using simulation tools. An experiential learning process for project management requires an environment where students can act as managers. Besides, in an artificial learning situation, student motivation and engagement play an important role. Some special drivers for such motivation include self-realization, challenge, victory, rewards, pleasure, and fun. In this sense, games can be integrated to simulation models, adding fantasy, visual effects, and a more compelling interaction model for students. Digital games are also a growing market to adults: the average American player age is 29 years while the average task-force age is 39 years [15]. However, playing is usually considered to be the opposite of working.

Some current research works present the adoption of game concepts in software engineering education, such as the SimSE Tool [13] and the SESAM Project [5]. However, the effectiveness of simulation and game-based learning is a discussion point. Since the educational effects of different approaches are difficult to isolate, measure and trace, their effectiveness is not well documented and established. There are also many disturbing factors that must be taken into account in a comparison, including subjective factors such as the quality of a teacher or a book. Some approaches may be more suitable according to some specific situations and educational goals [8].

3. The Incredible Manager

To evaluate the game-based learning approach, we have developed a simulation game, called *The Incredible Manager*. The diversity of game styles makes it difficult to establish a game taxonomy, but we consider adventure, puzzle, and simulation-games well suited for educational goals aiming at reasoning, judgment, decision-making and system thinking.

By using the game, a student is asked to act as a manager, planning and controlling software projects with success, i.e. within the planned schedule and budget estimates. The game construction is based on three main

elements, as can be seen in Figure 1: a simulation model, a simulation machine, and a game machine, which will be detailed on the following sections.

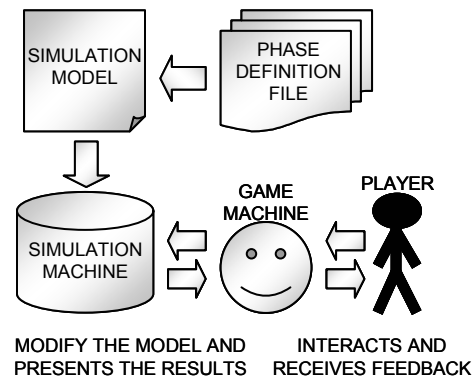


Figure 1. The game structure

4. The Simulation Model

The simulation model represents the world and the aspects that will be simulated and presented to the player. However, software development projects are difficult to model since they are classified as systems of complex dynamics [17].

Addressing these difficulties, System Dynamics [7] is a modeling discipline based on a holistic view to describe and evaluate the visible behaviour presented by a system. Such behaviour is determined by the structure of the elements that participate in the system and the relationships among them. Such structure and relationships are described in the model through mathematical equations. This modeling discipline has already been used in the development of software project models [1], which became a base for subsequent reviews and extensions by other authors.

One of these extensions is the scenario-based project management paradigm [2], which separates uncertain aspects from known facts in project models. This separation occurs by building distinct models (namely scenario models) for each uncertain aspect that can influence a software project. The models can be more easily developed, modified, integrated and expanded to embrace management knowledge from the technical literature and practice.

Scenario models provide a library of generic management events and theories that an instructor can integrate to a project model and present to management trainees during a simulation session. By using simulations, it is possible to evaluate the impacts of the desired scenarios over the expected project behaviour.

5. The Simulation Machine

The simulation machine is the element responsible for controlling simulation steps, iteratively calculating model equations to evaluate system elements' behaviour. Different from ordinary simulators, the simulation machine for a game must be interactive. Using ordinary simulators, a student playing the role of a manager should prepare a plan (configuring model elements and relationships) and follow it until the end of the simulation. This static structured simulation does not represent with confidence the reality: during a software development project, the manager makes decision all the time during the development process – not only during the planning phase –, modifying the original plan (and thus, the model structure) to better control the project.

The simulation machine developed in our work is able to translate and simulate System Dynamics models and to process events during simulation. This dynamic structured simulation can take into account player actions over the model structure during the game run without rebuilding the behaviour generated by previous simulation steps.

6. The Game Machine

The game machine is the element that the player interacts with and receives visual feedback from the model simulation. It is able to deal with continuous game phases. Each phase represents a separate simulation model, configured externally in a game configuration file. This flexibility allows the adoption of several different educational goals using the same game. The player starts the subsequent phase immediately after finishing the preceding one, even if the later was concluded without success.

During a phase, the project development takes place with hired developers executing a net of project tasks (defined in the model). The characters who take place in the game are:

- Manager - The player's role, responsible for project planning and several decision-making;
- Developers - The team to develop the project. Each one has different skills and characteristics such as hourly cost and work hours per day;
- Boss - Represents all the project stakeholders and is responsible for the project plan acceptance and project pressure during development.

Each game phase is also divided into five steps: Begin Phase, Project Planning, Planning Acceptance, Project Execution and End Phase.

6.1. Begin Phase

The beginning of a phase presents the project to be managed by the player. The project description document includes the description product to be delivered, special scenarios that may impact the development and project characteristics: tasks and its function points, quality, schedule, budget demands and constraints.

6.2. Project Planning

In this step, the player is asked to develop a project plan to be executed. The player must select and hire appropriate developers from those available in the market. Once the team is defined, developers must be assigned to execute the task network. Each task must be executed by only one developer and the player must determine the effort (number of days) necessary to complete each task. The effort on quality assurance activities, such as inspections, is also up to the player decision: the player can even remove these activities from the project plan.

The player can modify the project plan at any time during project execution, firing and hiring developers, modifying their work-hours or modifying the estimated duration of tasks. The project plan resume shows the overall budget and duration estimates to the project.

6.3. Planning Acceptance

Once the project plan is ready, it must be send to the stakeholders for acceptance. The plan can be approved or not. A project plan is refused if its overall estimates are over the constraints described in the project presentation at the begin phase. If the project plan is refused, the player must plan for it again until it is accepted.

6.4. Project Execution

The network task of the accepted project plan is then executed by the allocated team. Figure 2 illustrates the office room where development takes place. The time and funds available for development, as shown in the bottom on the screen, are the ones requested in the accepted project plan. Project execution runs in continuous turns, consuming project resources. The player must be aware of the project behaviour and take corrective actions when necessary. Visual effects and project reports show the game characters (and model elements) state, such as exhausted developers, late tasks, project without funds, and so on.

To avoid finishing the resources before project completion, the player may need to modify the original plan on the fly. According to these decisions, different players can live the experience of managing the same project in different ways.

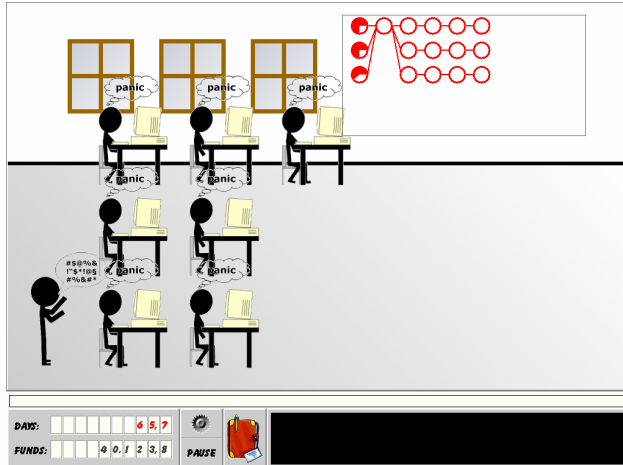


Figure 2. The office room

6.5. End Phase

The phase ends when the project resources are done without project completion (failure) or when all the tasks are done and the project is completed with success.

7. Game-based Learning Evaluation

In the software project management context, to depict the game utility and the improvements that our research should develop, two runs of a case study were conducted to evaluate the adoption of *The Incredible Manager* game within a training concept. The training was divided into simulation and discussion sessions, as stated in [11]. However, only one session of simulation and discussion was applied.

During the simulation session, the subjects were asked to play one phase of the game. When the simulations were finished, an instructor and all subjects participated in a discussion session. The instructor presented common scenarios and approaches of project management, positive and negative examples for specific decision-making situations, allowing the students to better interpret their actions and performance during the simulation session.

The first study was conducted with 7 subjects (1 D.Sc. student and 6 M.Sc. students) from a software project management course of a Brazilian university. The second study was conducted with 8 subjects (6 M.Sc. students and 2 B.Sc. students) from a laboratory for industrial software development within a different university in Brazil. All subjects received training in project management topics (e.g. function point estimates) and in the game utilization. The training session last 20 minutes in both runs. The first simulation session last between 50 and 120 minutes for the first run and between 55 and 140 minutes for the second run. Although the overall project function points have been kept, the number of project

tasks was reduced from 20 to 15, in order to reduce the second study duration, however, without success.

Only one of the subjects reached the end of the game with success (in the first study). Despite the failure, the subject's feedback was considered positive. The training concept with the simulation-game instrument was considered motivating, dynamic, practical and enjoyable. Subjects pointed out some important aspects such as psychological pressures (from continuous-time turns and compelling visual effects), high difficulty as a motivating challenge to the player and the entertainment factor while executing the game without losing the engagement to achieve the goals. Tables 1 resumes the results of the two runs.

Table 1. Game-based Learning Evaluation Results

	Raised	Indifferent	Reduced	
<i>PM Skill</i>	100%	0%	0%	
<i>Interest in PM</i>	87%	13%	0%	
	Good	Indifferent	Bad	
<i>Game-based Training</i>	100%	0%	0%	
	All	None	Lots	Few
<i>Presented Lessons Learned</i>	0%	0%	100%	0%
	Yes	No	Much	Little
<i>Was the training fun ?</i>	47%	0%	53%	0%

The main limitations and drawbacks reported by subjects were related to simplifications that were made to allow the creation of a simulation model for software projects. For instance, our current model is unable to represent real-world situations, such as multiple developers working together in a single task, social interactions among developers, psychological and organizational issues. Some subjects demanded the adoption of a multi-user interface, distance-learning facilities in the game, and some kind of wizard to trace and explain the actions, consequences, lessons learned, and alternative routes for decision-making during the execution of the game. Such wizard would help users to evaluate their own performance after executing the game.

8. Final Considerations

In this paper we analyzed the adoption of practical mechanisms to complement the traditional content-centric education strategies. The current focus is on training software project managers, since the lack of knowledge of management techniques and the inadequate use of management techniques is considered to be a root factor that inhibits project success.

Simulation-based games seem well suited to be introduced in an experiential learning situation, such as

required by manager trainees. They give to the student the opportunity of experimenting the consequences of executing or neglecting important project management functions, confront himself with complex issues that must be resolved during project development, and test different approaches and solutions of project management, learning by observing their consequences.

With the evolution of project simulation models, many limitations of current simulation-based game will be addressed to provide a more realistic situation, increasing the number of training scenarios and enhancing knowledge transference. The difficulties of formal models development open a special demand for graphical tools increasing the abstraction level to real world concepts, turning the development of complex models more intuitive and flexible.

To keep up with software project models evolution, the simulation machine presented in this paper should be extended to show more state transitions and graphical feedback, enriching the player perception and entertainment. The simulation machine is able to deal with different models developed upon an existing project management meta-model.

Besides the simulation model, many other research areas can be highlighted: pedagogical evolutions to the training concept with games, art evolutions over game usability and multimedia presentation, research on traces over player actions and performance, and psychological researches about cognitive and motivational issues related to game-based education.

Acknowledgements

The authors would like to thank all the subjects involved in the studies and CAPES for the financial investment in this work.

References

- [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, pp. 21-35, 2004.
- [3] BROWN, N., "Industrial-Strength Management Strategies", *IEEE Software*, v. 13, n. 4 (julho), pp. 94-103, 1996.
- [4] DOYLE, J.K., FORD, D.N., RADZICKI, M.J., TRESS, W.S., *Mental Models of Dynamic Systems* [online]. Available at <http://www.wpi.edu/Academics/Depts/SSPS/Faculty/Papers/27.pdf>. [01/07/2003].
- [5] DRAPPA, A., LUDEWIG, J., *Simulation in Software Engineering Training*. In: *Proceedings of the International Conference on Software Engineering*, pp. 199-208, Limerick, Ireland, June, 2000.
- [6] FORRESTER, J.W., *System Dynamics and the Lessons of 35 Years*, [online]. Available at <http://sysdyn.clexchange.org/sdep/papers/D-4224-4.pdf>, 1991. [05/02/2004].
- [7] FORRESTER, J.W., *Industrial Dynamics*, Cambridge, MA: The MIT Press, 1961.
- [8] 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 Zealand, July 1999.
- [9] KNOWLES, M., *Andragogy in Action*, Jossey-Bass, San Francisco, CA, 1984.
- [10] MAIER, F.H., STROHECKER, 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.
- [11] 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)*, pp. 305-313, Antwerp, Belgium, May, 1998.
- [12] MERRIL, D., *Training Software Development Project Managers with a Software Project Simulator* [online]. Master Thesis Proposal Arizona State University, Tempe, AZ, USA. Available at <http://www.eas.asu.edu/~sdm>, 1995. [01/07/2003].
- [13] 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.
- [14] 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, pp. 367-395, 2003.
- [15] PRENSKY, M., *Digital Game-Based Learning*, McGraw-Hill, 2001.
- [16] STANDISH GROUP, *The Chaos Chronicles*. The Standish Group International, 2003.
- [17] STERMAN, J.D., *System Dynamics Modeling for Project Management*. MIT System Dynamics Group, Cambridge, MA, USA, 1992.