Educational Games Development: Issues and Challenges

Daniela C. C. Peixoto
Centro Federal de Educação Tecnológica
Department of Computer
Brazil
cascini@decom.cefetmg.br

Rodolfo F. Resende
Universidade Federal de Minas Gerais
Department of Computer Science
Brazil
rodolfo@dcc.ufmg.br

Clarindo Isaías P. S. Pádua
Universidade Federal de Minas Gerais
Department of Computer Science
Brazil
clarindo@dcc.ufmg.br

***Abstract* – The rapid technological advance allowed an increasing number of educators to use computer games for educational purposes. More and more simulation games are being employed in education and training with the aim of helping to attract and sustain students' engagement. It is believed that these games will bring overall improvement and, in particular, improved pedagogical outcomes. Besides the advantageous aspects of educational games, varieties of challenges were reported by previous researches. Some examples include the difficulty to customize these games to educational purposes, the use of complex technologies, and the lack of design frameworks with associated validation. In this paper, we describe our experience in creating two simulation games in the Software Engineering field. We present the problems that we faced and the proposed solutions during the design and development. We guided this research by a list of game attributes that are linked to learning. Our contributions include (i) a description of the simulation games; (ii) the discussion of the development activities, and (iii) the problems and proposed solutions. The outcomes of this study can be used as a basis when developing new simulation games as well as a starting point for the analysis and improvement of processes that already exist.**

Keywords— simulation games; software engineering; development processes; learning attributes

I. INTRODUCTION

Recently, there has been an increasing interest in the use of games as valuable learning tools. The potential benefits of using games were highlighted, among others, by Boyle and colleagues [1], Connolly and colleagues [2] and Braghirolli and colleagues [3]. With the games, students can practice activities that are infeasible to practice during an academic course, due to restrictions of time and resources. Thus, using games for educational purpose has been considered an important approach [4].

In the Software Engineering field, there are a number of researches that discusses the use of simulation-based games as educational tools [5], [6], [7], [8]. Using simulation-based games, it is possible to practice some Software Engineering skills in a more realistic environment, increasing students' motivation and engagement. It is also possible to reduce the risks of failure and the high costs of an actual practice. In addition, students can work at their own pace [8].

Besides the great advance in the researches of games as learning tools, the theoretical base of educational games is still incipient [9]. It is important to identify what are the attributes that will affect learning, in order to incorporate these attributes into the game development process. Numerous researchers have proposed a list of elements that can contribute to learning [9], [10], [11], [12]. For example, Wilson and colleagues [11] identified 18 game elements and their effects on the learning outcomes, including, among others, conflict, challenge, control and surprise. Bedwell and colleagues [9] reorganized these 18 attributes into nine categories.

The creation process of a simulation game relies often on a multidisciplinary work involving, among others, technology, design, and psychology knowledge. This is a complex task and, in the recent years, a few frameworks have been proposed to help during this process [4], [13], [14], [15], [16], [17]. Yet, with these frameworks, it is not possible to make an association among the game requirements, game attributes, and activities in the development process [18]. In this work, we present how two Software Engineering simulation games were created, highlighting how some game attributes were incorporated into the development process. Our main contribution is the set of aspects of our experience in the development of the simulation games, showing the problems that we faced and the proposed solutions. This discussion is guided by a list of selected game attributes that are related to learning. The presented results can be used as an aid in the design of new games and can also be used to analyze the development processes of existing simulation games.

The remainder of this paper is structured as follows. Section 2 outlines a description of the main concepts considered in this article, i.e. game attributes, simulation games and development frameworks. Section 3 describes the methodology used in this research. Section 4 shows how the games attributes were represented in the simulation games. Section 5 presents a discussion of the results and Section 6 concludes the paper with the final considerations and future work.

II. BACKGROUND

In this section, we present: (A) game attributes that impact on the learning results; (B) a brief description of two simulation games, SPIAL and MScrum; and (C) an overview of frameworks used for the creation of educational games. These two games were chosen because they were developed in our laboratories and we have available all documentation.

A. *Game Attributes*

Several attempts have been made to identify what are the game attributes that foster learning. In this area, the literature is incipient and there is a lack of researches that relate different game elements to specific learning outcomes [9].

Malone [10] argues that game engagement primary attributes are Challenge, Curiosity, and Fantasy. Garris and colleagues [19] presented six core features that are needed for promoting learning: Fantasy, Rules/Goals, *Sensory Stimuli*, Challenge, Mystery, and Control. Wilson and colleagues [11] conducted a comprehensive literature review in order to evaluate specific game attributes and their impact on learning outcomes. They identified 18 attributes, including attributes such as Conflict, Challenge, Mystery, and Safety. A more recent work of Bedwell and colleagues [9] reorganized the 18 attributes presented by Wilson and others [11] into nine categories: Action Language, Assessment, Conflict/Challenge, Control, Environment, Game Fiction, Human Interaction, Immersion, and Rules/Goals. They used a literature review and subject-matter expert (SME) judgment. The aim of Bedwell and colleagues work was to reduce the overlap among similar game attributes.

Despite all these advances, a more robust research is still needed in order to identify how different game elements are related to specific learning outcomes [9].

B. Simulation Games

B.1 SPIAL

SPIAL [20], [21] (*Software Process Improvement Animated Learning Environment*) is a desktop-based, graphical, interactive, and adaptable simulation game (Figure 1). The game goal is to improve the Software Engineering learning, using a simulation approach. The knowledge model constructed within SPIAL represents Software Process Improvement (SPI) concepts. These concepts were based on CMMI-DEV version 1.3 [22] framework. The SPI knowledge model is coupled with a software development process model. The player alternates between the software development project and the SPI project. The message sent to the player is related to how process improvement affects the activities of a software development project. Thus, SPIAL allows students to practice SPI techniques and the best practices of Software Engineering.

SPIAL is a single-player game in which the player takes on the role of a manager of an SPI group in a software development organization. The player is given a process improvement task and he or she can interact with other stakeholders (high level management, project manager, team member, consultant, and customer) represented as non-player characters, i.e. a character controlled by the computer (see Figure 1). The type of improvement that is required is stated at the beginning of the game and can include, for example, cost and defect reduction and productivity improvement. In order to complete the task, the player can make investments for improving specific process areas of a software development project. A good investment strategy will result in improvement of process areas and a bigger budget for further investments. The player can visualize project estimations, indications of process areas capability level and decide in which process areas to invest. During the development project, the player can visualize the effects of his/her selections on the outcomes (productivity, defect, cost, and time-to-market measures) and, if needed, change his/her investments. The outcome is a score that represents how close the results are to the initial proposed target.

The learning context assumed in SPIAL takes into consideration that the game will be played in a classroom. At the beginning, instructors should provide the basic concepts about SPI and software development to the students. The feedback mechanisms are useful resources for helping players understand concepts related to software process improvement.

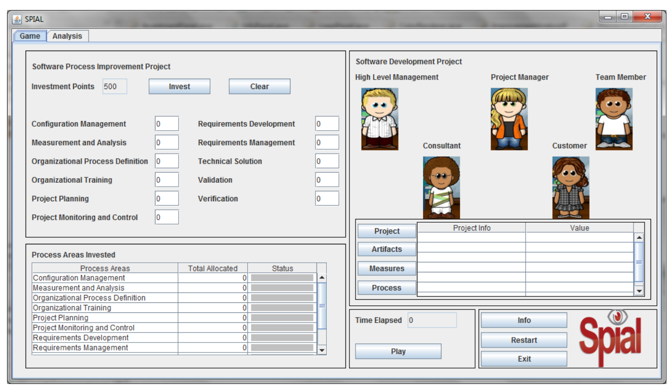

Fig. 1. SPIAL Graphical User Interface.

B.2 MScrum

MScrum (*Multiplayer Scrum Simulation Game*) is a web-based multiplayer simulation game, which simulates the use of Scrum methodology [23] in a software development project. The knowledge model conveys the idea that the several phases of a Scrum-based project can be understood in the context of playing simulation games. In addition, the game scope covers other practices used in an agile software development project. The game goal is to present to students how a Scrum agile project works.

Players have the role of a Scrum Master. At the beginning, players select the members of the team from a pool of resources (see Figure 2). Players can choose the team members according to their experience level and their costs. During the development, problems may occur (see Figure 3) and players can interact with the team members. The interaction occurs through selection of answers to specific questions. The simulation rewards correct answers and penalize unjustifiable ones. Scrum meetings occur at the end of the day (daily meeting) and at the end of the sprint (sprint review meeting). During these meetings, players can also interact with the team members, answering specific questions. As a multiplayer game, different players can communicate and share experiences using instant messaging. They can also visualize a rank of scores.

The student model subjacent to MScrum assumes that students need some prior knowledge of agile methods and basic understanding of agile practices, such as, pair programming. Students, as in SPIAL, can use the feedback mechanisms in order to understand how an agile project works.

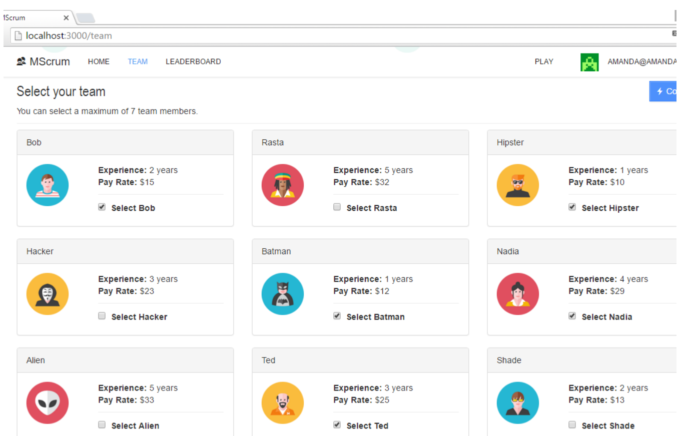

Fig. 2. Initial MScrum Interface – Team selection Interface.

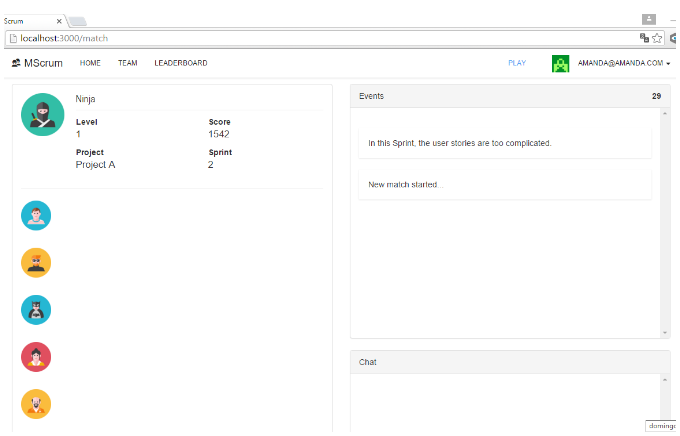

Fig. 3. Example of an event that occurs during a Sprint.

C. *Framework*

The educational game frameworks encompass learning theories and best practices of game development [18]. In the literature, we can find some frameworks.

van Staalduinen and Freitas [17] defined a game-based learning framework that can be used to develop new games or to evaluate existing ones. In this framework, the first step is the definition of the goals and learning objectives/contents. Then the instruction elements associated to the learning cycle (user learning, user behavior, player's feedback and user engagement) are defined. Finally, it includes two types of assessments: debriefing and system feedback.

A work from Marcos and Zagalo [15] proposes a development model of serious games based on the processes of digital arts. The model encompasses six phases: Concept Design, Narrative Design, Experience Design, Game Design, Game Implementation, and Game Deployment Planning.

Sommeregger and Kellner [16] proposed six design stages for the creation of educational adventure games, including Conceptual Design, Game Design, Implementation, Testing, Validation and Project Management.

Kirkley and colleagues [13] developed a model named Simulation-game Instructional Systems Design process (SG-ISD) which supports a specific instructional system design. SG-ISD allows formative and summative evaluations.

Loh [14] described a 10-steps instructional development model. It is a detailed model which includes: determining target audience and learning content, determining the amount of funding and time available, writing game narratives, selecting the game development kits and game bundle, designing the video game and the game mechanics, designing the interactive learning instruction, developing, executing beta testing and usability testing, public release, and assessing the efficacy.

Kickmeier-Rust and colleagues' work [4] defines an interdisciplinary approach in order to design educational games. The focus is on the psychological and pedagogical elements, such as analysis of curricula and definition of the background story, according to the educational aims.

III. METHODOLOGY

In order to analyze how game attributes fits into the game development processes, we have studied two simulation games. The first simulation game, SPIAL, had its development concluded in 2013. After this stage, we conducted several experiments with students [24]. In SPIAL development, we adopted an incremental and iterative approach, where each step involves different knowledge. The second simulation game, MScrum, started its development in 2015. In this game, we adopted an agile development approach. So far, we implemented most of the simulation engine and the interface prototypes.

In this research, we used the model presented by Bedwell and colleagues [9] as a basis for our discussion. The reason for this choice is that according to Landers [25] this model has been carefully designed. Also, this model presents a matrix associating specific attribute categories to learning outcomes.

After selecting the simulation games and defining the attributes model, we conducted the following activities, described in the next sections: (1) we analyzed how the attributes were represented in SPIAL and MScrum; (2) we discussed what were the students observations during the SPIAL experiments; (3) we identified problems and solutions according to these observations; and (4) we described lessons that we have learned.

IV. Game Attributes

We used the process stages described in Sommeregger and Kellner work [16] for the analysis of the development activities in SPIAL and MScrum. These stages are specific for the creation of educational games and include conceptual project, design, implementation, testing and assessment. Considering the games attributes presented below, we can map most of them on the conceptual project and the design stages. We observed that important factors, such as, performance and usability, that are crucial for the development of an educational game, were not identified as an attribute, in most of the considered works.

A. Action Language

According to Bedwell and colleagues [9], the Action Language category consists of “language” and “communication” attributes. “Language” refers to the method by which players present their intentions to the system. “Communication” refers to specific communication rules between the player and the system. The interface is an important element in this category, because it mediates the interaction.

In SPIAL, the Action Language attributes are represented by the investment and the system feedback elements. Players should inform the investments for improving specific process areas of a software development project. Considering the system-to-user interaction, the non-player characters communicate the effects of the player’s actions through bubbles over their heads. Examples of messages include "Since there is dependency among process areas some of your investments may not be effective", or "Poor investment decisions result in a reduction of business value, and a reduction in the number of investment points."

In MScrum, the Action Language attributes are expressed by the selection of the team members and the players’ response to the questions that appear during the game. The system communicates to players the project status using, for example, the *Burndown* chart [23].

In the current versions of SPIAL and MScrum, the interaction elements are quite simple. The players provide the input data and the system presents some feedback using text messages or graphs. Until now, our development efforts were concentrated on defining a suitable game behavior, which supports our idea of simulating a real software organization.

Figure 4. Stakeholders’ communication in SPIAL.

B. Assessment

According to Bewdell and colleagues [9], the assessment category is related to the content and nature of the feedback presented to the player, during or after the game. It is composed of two attributes: “assessment” and “progress”. The “assessment” attribute is a measure of achievement. “Progress” attribute shows what goals have been achieved. Both attributes can guide players’ actions. Feedback is a critical element that impacts on both attributes. According to Garris and colleagues [19] feedback is a central component of the judgment-behavior feedback cycle. Because it permits players to interpret the software, it also supports players’ decisions about interactive goals and provides information for novice players during the learning process.

In both simulation games we adopted a “trial and error” strategy, since the interface signs are limited in their message communication. This is an expected aspect in simulation games, but in some cases the miscommunication gets in the way of the player’s learning process. We were challenged to find a balance between the information displayed in the interface and the information provided through other media (e.g. manual, tutorials, and so on), in a way that the game is still inviting and challenging. Regarding the type and the instant of the feedback, we present the score only at the end. The consequence of this is that good results are normally obtained only after several game plays. In contrast, there exist informative feedbacks that are presented throughout the whole game helping the player’s progress. It varies, for example, in signs: bubbles, graphs, tables, and animated elements.

C. Conflict/Challenge

According to Bewdell and colleagues [9], this category refers to the presentation of problems and the difficulty of solving them. The Conflict/Challenge category includes the “adaptation”, “challenge”, “conflict”, and “surprise” attributes.

Simulation games are more enjoyable and fun when they provide sufficient challenge for the player [10]. In both games, we experimented with different settings in order to establish a balance between tasks considered easy and difficult. The students were challenged to solve the problems that appear during the game play (e.g. increasing number of defects, constraints on budget and time). Overcoming these challenges enables students to succeed. These challenges were defined according to a mathematical framework, which maps the actual state of the variables and present, to the player, the next state and the events triggered during the transition. The uncertainty of the events can be considered a surprise element.

Adaptation of the learning environments can assist students with different learning styles, different levels of initial knowledge and different expectations and objectives [26]. In SPIAL and MScrum, adaptation can be done only in a static way, not during the game (dynamic adaptation). In order to adapt these games, instructors need to change an XML file that specifies the simulation model. We did not model a situation that can cause conflict in these games. However, the problems presented during the game may cause players' disagreement due to the selection of the most suitable answer/action.

D. Control

According to Bewdell and colleagues [9], this category refers to "control" and "interaction (equipment)" attributes. The first attribute, "control", represents the amount of control that users exert on the content or game play. The second attribute, "interaction (equipment)", represents the degree of changes in response to player's actions.

In both games, players provide inputs as a reaction to the prompted events. The simulation games determine the next events based on the players' inputs. For example, in SPIAL, incorrect investments will increase the defects number, and after some time steps, the investment points will reduce. Players' decisions affect the final score. The design insured that players do not have a tight control in these games.

E. Environment

According to Bewdell and colleagues [9], environment refers to the "location" attribute that consists of the settings where the player is immersed.

In both simulation games, we tried to involve players in an environment similar to a software development organization. The events, the non-player characters, and the interface elements are inspired in a real software development organization. The goal is to provide students an experience of a software development project and process within the academic environment. In MScrum, for example, we represented most of the Scrum elements: ceremonies, artifacts, daily problems, team members and stakeholders.

F. Game Fiction

According to Bewdell and colleagues [9], Game Fiction represents how the game world is presented to the player. It is composed of the "fantasy" and "mystery" attributes. "Fantasy" refers to elements that are distinct from the real world and "mystery" represents the unknown situations in the game.

SPIAL and MScrum capture the most important aspects of a physical software development organization. The simulation games worlds are tightly coupled with their aims and objectives. In the virtual organizations, players assume a Scrum Master role, in MScrum, and an SPI group manager role, in SPIAL. Players monitor the project and the employees within a delimited time frame. The combined effects of players' actions represent the mystery attribute. Although with some guidance, players need to discover how the game objectives can be achieved.

G. Human Interaction

According to Bewdell and colleagues [9], this category refers to human interaction throughout the course of the game. This can happen in the "real space (interpersonal interaction)" or using "game media (social interaction)".

SPIAL is a single player and MScrum is a multiplayer game that allows interaction among players. In MScrum, they can interact using an instant messaging communication mechanism. Any information can be shared, since the game does not control this communication channel. Other kind of interaction is through the ranking information. A player has a dynamic view of other players' score value.

H. Immersion

According to Bewdell and colleagues [9], Immersion describes the affective and perceptual experience of a game. It consists of the attributes: "pieces/players", "representation", "*sensory stimuli*", and "safety". The "pieces and players" are objects or people included in the game narrative. The "representation" attribute consists of the players' perceptions during the game. The "*sensory stimuli*" refers to the visual and auditory *stimuli* implemented in the game environment. "Safety" refers to playing the game and observing the results without great risks.

In our games, players can interact with other non-player characters (NPC) representing software development stakeholders. In MScrum, the interaction happens between the Scrum Master (player) and the team members (NPC). In SPIAL, the interaction happens among the SPI manager (player) and other NPCs representing high level manager, project manager, team member, consultant, and customer. In both games, the player perception is almost the same, he/she is immersed in a software development organization, which is improving its processes (SPIAL) or developing a project (MScrum). The visual *stimulus* can be seen when the system sends a feedback to the players via text messages or graphs. Graphs allow players to control metrics, such as, quality, scope, and cost. With respect to safety, the designers and players did not raise any concerns.

I. Rules/Goals

According to Bewdell and colleagues [9], the Rules/Goals category refers to having the definition of clear rules and goals. The attributes, "rules" and "goals", describe the methods and the reasons for which player interact within the game.

One important design concern in both simulation games was the clear specification of the goal. Thus, at the beginning of each game, players are showed their objectives. On the other hand, this is not true for the rules. The Software Engineering rules are not explicitly presented to the players in these games, but they can search for their descriptions on the game interface. These rules correspond to an important learning process, where players become skilled in anticipating and correcting the effects of choosing incorrect actions. If the players learn these rules, they would eventually be in a position to achieve a good score and understanding the dynamics of a Software Engineering team.

In order to define these rules, we have searched for the best practices on textbooks and other Software Engineering simulation games. These rules are used to teach the best practices of Software Engineering to students, rewarding or penalizing their actions. This set includes rules that are imprecise (rules related to unspecified characteristics and values) and rules beyond the Software Engineering area, including a wider range of business processes rules. From this set, we selected the ones related to the SPIAL and MScrum game context.

V. DISCUSSION

Developing and evaluating a simulation game is a not simple task. In this section we discuss the problems and solutions that we faced during the development and assessment of SPIAL and MScrum.

A. Development process

During SPIAL creation, we did not follow a documented formal development process. We investigated the problems found in other simulation games and we tried to overcome them. In these investigations, we observed that crucial points were missing, such as the specification and execution of comprehensive validations, mainly regarding the learning outcomes. In SPIAL, we carried out distinct validations [24]: experiment, user test, and semiotic inspections with aim to evaluate the learning results.

In MScrum, we followed agile practices. During the development, we carefully included activities in order to cover the game attributes described in this research. Furthermore, we also investigated user interaction attributes specific for multiplayer games. We conducted a literature review and we identified the following attributes: sociability [27], communication [28], competitiveness [29] and collaboration [28], [29]. In order to address these interaction attributes, we decided to phase our efforts. In these initial efforts, we included an instant messaging communication mechanism and a ranking score. In the current development stage, our main challenge is the definition of an assessment process that will address these attributes.

During our research on educational game frameworks, we identified that most of them emphasize the initial development phases (conceptual project and design), and they did not invest sufficiently in different types of tests or assessments. This is also true in the development process of other Software Engineering simulation games [30], [31]. In these games, the assessments are mostly restricted to small pilot tests or subjective evaluations [24]. Usually the evaluation methods cover few characteristics of the game environment, varying according to the needs and experiences of its designer. Therefore, we recognized the need of a more effective understanding of the educational contributions, mainly regarding the learning outcomes.

B. Assessments

Since MScrum is under development, we did not evaluate it yet. On the other hand, in SPIAL, we used the student appraisal, like in the other games, but we also brought other evaluation techniques. Part of our approach was inspired by SimSE [8], since it is an example of simulation game with extensive evaluation of the learning process it promotes. We evaluated SPIAL using three methodologies supported by UGALCO framework [24]: (1) Semiotic Inspection Method; (2) Experiment; and (3) User Test. These evaluations are detailed in Peixoto et al. [24] and briefly described below.

In the first evaluation, using the Semiotic Inspection Method (SIM) [32], an inspector carried out the assessment, playing the role of a student. After conducting the SIM steps, a unified analysis was produced, highlighting the main communication breakdowns. According to the inspector, important aspects to understand the core behavior were missing, such as the reason why sometimes investments do not produce any improvement. We produced a new game version, with some modifications, in order to provide more guidance to players.

Then, in the second evaluation, we carried out two experiments with undergraduate students of a Software Engineering course. In the first experiment, our aim was to gain a better understanding of SPIAL effectiveness as an educational tool. We also determined the strengths and weaknesses of SPIAL through the feedback of the students who played it. In total, 11 undergraduate Computer Science students participated in this pilot experiment. Each student answered a background and a pre-test questionnaire, before playing SPIAL, and a post-test, after playing it. In general, the students were able to understand how the simulation game works, they found SPIAL quite enjoyable, and they had fun during the game play. They learned new concepts and reinforced concepts taught in a Software Engineering course.

In the second experiment, we applied a specific framework [24] in order to evaluate the game experience, adaptivity, learning experience and usability dimensions. Students answered specific questions for each dimension, with respect to the Understandability/Usability dimension an example of question is "how easy was the understanding of the game inputs and the outputs?". In total, 15 students participated in this experiment. The highest scored attribute was the learnability, i. e. the students understood how the game play works (Table I). The least scored attribute was the learning goals, and this reflects the fact that players did not feel that they gained a great amount of new knowledge.

TABLE I. AVERAGE RESULTS FOR UGALCO'S ATTRIBUTES [24] .

Dimension	Attribute	Avg.
Game Experience	Challenge	0,47
	Competence	0,68
	Immersion	0,70
	Positive affect	0,65
Adaptivity	Cognitive and motivational aspects	0,60
Learning Experience	Learning goals	0,46
	Content appropriateness	0,63
Usability	Operability	0,45
	Understandability	0,63
	Learnability	0,77
	Attractiveness	0,47
	Satisfaction	0,62

The user test occurred in a one-to-one setting – one subject and one evaluator. As corroborating with the previous experiments, students reported that they reinforced SPI concepts playing SPIAL, but they were not sure about what new concepts were presented to them.

Besides these evaluations, we were not able to trace the real learning effects considering these assessments. A more elaborated evaluation mechanism, using pre and post game playing questionnaires with specific questions, should be done. We also failed to adapt the game as students became more proficient. Students reported that they were not anymore challenged and interested in playing the game after discovering its rules and behavior.

C. Other issues

Besides the development processes and assessments, another issue that we identified was the absence of extensive investments on the interface design. Most of the game developers we investigated were not experts on this subject. Usually, they focused on the game behavior and let the interface improvements as future work.

In SPIAL and MScrum, we decided to use simple interface elements. In SPIAL, the reason was the reduced development time. In MScrum, the reason was the technology complexity. The difficulty in creating a multiplayer game is considerable, both in the requirements definition as in the technology understanding.

We faced other challenges, such as finding the appropriate balance in this particular teaching and learning context, between the real and virtual world. The simulation game is a case study of a particular reality where players play roles of characters in a particular situation following a set of rules and interacting with other characters. The settings, players' role, players' educational profile and the learning objectives are important design aspects related to the virtual world where the game occurs. As simulation games, we represented the basic concepts that could link students to the idea of a real development organization.

Another challenge was the amount of control that players have over the game. With limited control, students can be stuck during the game without being able to progress. Since games are complex and a simple change could significantly impact on the learning results, we decided to allow players to have only limited control over the game. Therefore, they basically inform a few data and wait for the system feedback.

D. Lessons learned

The following lessons were learned during the development process employed for the creation of SPIAL and MScrum.

D.1 Development Process

As developers of instructional tools, designers need to address a set of characteristics and to deal with complex technologies. A framework for developing educational games could help in this difficult task. These frameworks would help to clarify what kind of learning outcomes the students can expect, the identification of core game attributes, and the activities to be carried out. In this work, we showed that even without employing these frameworks, we achieved satisfactory results. However, our development experience was inspired on other simulation games development.

The lesson learned is the importance of having game development frameworks in order to assist the simulation game creation. The frameworks found on the literature consolidate the best development practices, avoiding the rework of collecting this information.

D.2 Assessments

The assessments of learning effectiveness in Software Engineering simulation games employ a mixed-method approach, covering a few characteristics of the game environment. In the literature, we found relatively few studies that have definitively assessed the effectiveness of the educational game. Even in the development frameworks, there is little emphasis on these activities. After SPIAL evaluations, we can affirm that there is a need to invest in the definition of more effective learning evaluation mechanisms.

The lesson learned is that the assessments need to be planned during the initial development phases and strictly executed. Well-rounded assessments can provide the basis for the improvement of this research field.

D.3 Game Attributes

Game attributes provide the basis for determining what makes an educational game effective. There are a number of researches describing game attributes. However, there is a need to collect more robust evidence that associates these attributes to the expected learning outcomes. In MScrum, we were careful in the incorporation of these attributes. We also identified and defined specific user interaction attributes, commonly found in a multiplayer game.

The lesson learned is that appropriate development practices should check whether the games attributes have been addressed. This provides researches with an important guidance in order to organize development efforts.

VI. CONCLUSION

This work described the development activities of two Software Engineering simulation games: SPIAL and MScrum. We based our discussion on nine categories of game attributes. These two games addressed most of these attributes.

Our experience with this study has highlighted some important aspects that can be used to enhance the field of simulation games development. First, there is a need of more investments on educational games assessments. There have been relatively few studies that implement effective evaluations. Second, a development framework, specific for educational games, can be a useful guidance tool. Third, developing a multiplayer game is much more complex than a single player game. This includes not only the interface design, but also the technology and assessments methods. During our literature review, we did not find comprehensive assessments for multiplayer games. Finally, designers should invest sufficient time in defining important issues, such as the interface signs and the feedback types. We observed that most of the problems detected during SPIAL assessments could be traced back to these items.

As a future work, we plan to propose a game development process that will incorporate these relevant issues. We are working on improvements related to the UGALCO framework and we will employ these improved analyses in a context similar to this research. We also plan to address these issues in the MScrum development, including a comprehensive definition of validation mechanisms for multiplayer games.

ACKNOWLEDGMENT

This work was supported by CAPES, CNPq and FAPEMIG, entities of the Brazilian government.

REFERENCES

[1] E. A. Boyle, T. Hainey, T. M. Connolly, G. Gray, J. Earp, M. Ott, T. Lim, M. Ninaus, C. Ribeiro, and J. Pereira, “An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games,” Computers & Education, Volume 94, March 2016, pp. 178-192.

[2] T. M. Connolly, E. A. Boyle, E. MacArthur, T. Hainey, and J. M. Boyle, “A systematic literature review of empirical evidence on computer games and serious games,” Computers & Education, Volume 59, Issue 2, September 2012, pp. 661-686.

[3] L. F. Braghirolli, J. L. D. Ribeiro, A. D. Weise, and M. Pizzolato, “Benefits of educational games as an introductory activity in industrial engineering education,” Computers in Human Behavior, Volume 58, May 2016, pp. 315-324.

[4] M. D. Kickmeier-Rust, D. Schwarz, D. Albert, D. Verpoorten, J.-L. Castaigne, and M. Bopp, “The Elektra Project: Towards a new Learning Experience,” In M. Pohl, A. Holzinger, R. Motschnig, & C. Swertz (Eds.), M3 – Interdisciplinary aspects on digital media & education, 2006, pp. 19 - 48.

[5] L. Vos, “Simulation games in business and marketing education: How educators assess student learning from simulations,” The International Journal of Management Education, Volume 13, Issue 1, March 2015, pp. 57-74.

[6] A. Drappa, and J. Ludewig, “Simulation in software engineering training,” in Proceedings of the 22nd International Conference on Software Engineering, pp. 199–208, Limerick, Ireland, 2000.

[7] A. Jain, and B. Boehm, “SimVBSE: Developing a game for value-based software engineering,” in Proceedings of the 19th Conference on Software Engineering Education and Training, pp. 103 –114, Turtle Bay, Hawaii, 2006.

[8] E. O. Navarro, “SimSE: A Software Engineering Simulation Environment for Software Process Education,” PhD thesis, Donald Bren School of Information and Computer Sciences, University of California, Irvine, USA, 2006.

[9] W. L. Bedwell, D. Pavlas, K. Heyne, E. H. Lazzara, and E. Salas, “Toward a Taxonomy Linking Game Attributes to Learning: An Empirical Study,” Simulation & Gaming, Volume 43, no. 6, December, 2012, pp. 729-760.

[10] T. W. Malone, “What makes things fun to learn? Heuristics for Designing Instructional Computer Games,” In Proceedings of the 3rd ACM SIGSMALL Symposium and the First SIGPC Symposium on Small systems (SIGSMALL '80), 1980, pp. 162-169.

[11] K. A. Wilson, W. L. Bedwell, E. H. Lazzara, E. Salas, C. S. Burke, J. L. Estock, K. L. Orvis, and C. Conkey, “Relationships Between Game Attributes and Learning Outcomes: Review and Research Proposals,” Simulation & Gaming, Volume 40, no. 2, April, 2009, pp. 217-266.

[12] R. Garris, and R. Ahlers, “A game-based training model: development, application, and evaluation,” In Interservice/Industry Training, Simulation and Education Conference, Orlando, FL, 2001.

[13] S. E. Kirkley, S. Tomblin, and J. Kirkley, “Instructional Design Authoring Support for the Development of Serious Games and Mixed Reality Training,” In Interservice/Industry Training, Simulation and Education Conference, 2005.

[14] C. S. Loh, “Researching and developing serious games as interactive learning instructions,” International Journal of Gaming and Computer-Mediated Simulations (IJGCMS), Volume 1, no. 4, 2009.

[15] A. Marcos, and N. Zagalo, “Instantiating the creation process in digital art for serious games design,” Entertainment Computing, Volume 2, Issue 2, pp. 143-148, 2011.

[16] P. Sommeregger and G. Kellner, “Brief Guidelines for Educational Adventure Games Creation (EAGC)” In IEEE Fourth International Conference on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), pp. 120 - 122, March 2012.

[17] J.-P. van Staalduinen and S. de Freitas, “A Game-Based Learning Framework: Linking Game Design and Learning Outcomes,” Learning to play: exploring the future of education with video games, Springer, 2011.

[18] M. Ahmad, L. A. Rahim, and N. I. Arshad, "A review of educational games design frameworks: An analysis from software engineering," International Conference on Computer and Information Sciences (ICCOINS), 2014, Kuala Lumpur, 2014, pp. 1-6.

[19] R. Garris, R. Ahlers, and J. E. Driskell, “Games, Motivation, and Learning: A Research and Practice Model,” Simulation & Gaming December 2002, Volume 33, no. 4, pp. 441-467.

[20] D. C. C. Peixoto, R. M. Possa, R. F. Resende, and C. I. P. S. Pádua, “Challenges and Issues in the Development of a Software Engineering Simulation Game,” in Proceedings of the 42nd ASEE/IEEE Frontiers in Education Conference, Seattle, Washington, USA, 3-6 October, 2012.

[21] D. C. C. Kupsch, “SPIAL: A Tool for Software Process Improvement Training,” PhD Thesis, Universidade Federal de Minas Gerais, Brazil, 2012, SPIAL is available at: www.dcc.ufmg.br/~cascini.

[22] CMMI, “CMMI® for Development, Version 1.3,” Technical Report CMU/SEI-2010-TR-033, Software Engineering Institute, 2010.

[23] K. Schwaber, and M. Beedle, “Agile Software Development with Scrum (Series in Agile Software Development),” Pearson, 1st Edition, 2001.

[24] D. C. C. Peixoto, R. F. Resende and C. I. P. S. Pádua, "Evaluating Software Engineering Simulation Games: The UGALCO Framework," In Proceedings of the 44th ASEE/IEEE Frontiers in Education Conference, Madrid, Spain, 2014, pp. 1-9.

[25] Richard N. Landers, “Developing a Theory of Gamified Learning: Linking Serious Games and Gamification of Learning,”Simulation & Gaming, December 2014, Volume 45, no. 6, pp. 752-768.

[26] P. Moreno-Ger, P., D. Burgos, I. Martínez-Ortiz, J. L. Sierra, and B. Fernández-Manjón, “Educational game design for online education,” Journal of Computers in Human Behavior, September 2008, Volume 24, Issue 6, pp. 2530–2540.

[27] E. Ye, C. Liu, and J. A. Polack-Wahl, "Enhancing software engineering education using teaching aids in 3-D online virtual worlds," In Proceedings of the 37th ASEE/IEEE Frontiers in Education Conference, Milwaukee, WI, 2007, pp. T1E-8-T1E-13.

[28] J. B. Hauge, and J. C.K.H. Riedel, “Evaluation of Simulation Games for Teaching Engineering and Manufacturing,” Procedia Computer Science, Volume 15, 2012, pp. 210-220.

[29] W. Admiraal, J. Huizenga, S. Akkerman, and G. ten Dam, “The concept of flow in collaborative game-based learning,” Computers in Human Behavior, Volume 27, Issue 3, May 2011, pp. 1185-1194.

[30] T. Galvão, F. Neto, M. Bonates, and M. Campos, “A Serious Game for Supporting Training in Risk Management through Project-Based Learning,” Virtual and Networked Organizations, Emergent Technologies and Tools, Communications in Computer and Information Science, Volume 248, pp. 52-61, 2012.

[31] C. S. Longstreet, and K. Cooper, “A Meta-model for Developing Simulation Games in Higher Education and Professional Development Training,” in Proceedings of the 17th International Conference on Computer Games, pp. 39-44, Louisville, Kentucky, USA, 2012

[32] C. S. de Souza, C. F. Leitão, R. O. Prates, S. A. Bim, and E. J. da Silva, “Can inspection methods generate valid new knowledge in HCI? The case of semiotic inspection,” International Journal of Human-Computer Studies, Volume 68, Issue 1-2, pp. 22-40, 2010.