INCORPORATING PLAYFUL ACTIVITIES IN THE SOFTWARE ENGINEERING TEACHING

Carlos Mario Zapata Jaramillo
Universidad Nacional de Colombia
cmzapata@unal.edu.co

María Clara Gómez Álvarez
Universidad de Medellín
mcgomez@udem.edu.co

ABSTRACT

Software engineering comprises the application of a systematic and disciplined approach to develop software products. Such an approach includes the development of tools, methods, and theories in order to support the software development and management processes. Software engineering teaching has been focused on theoretical concepts and technical issues; communication, teamwork, and negotiation skills have been traditionally left aside. In this context, playful activities appear as an alternative for promoting the learning-by-doing approach as a way to increase the assimilation of concepts by the students. In this paper we propose four playful activities applied to a group of undergraduate students belonging to the system engineering program. We also describe the feedback from the students who participated in these activities. Promising results arise in order to replicate this type of activities in other software engineering subjects and similar knowledge areas.

INTRODUCTION

The so-called *software crisis* leads to a different way of thinking about the development of software products. In fact, by the end of the 60's, the arrival of a new era was promoted by several researchers and practitioners, searching for new ways of working in the recently created software engineering. Several tools, methods, and theories were created for this recent engineering, to the extent of applying a systematic and disciplined approach for developing software products and managing the software development process.

Since the very beginning of the software engineering, the teaching process has been focused on technical issues—e.g. modeling and programming—and theoretical concepts. Some other issues involving human factors—e.g. ethics, teamwork, communication, negotiation skills, competition, and problem solving—have been left aside. Most of the time, software engineering has been taught by using lectures and toy practical projects, i.e. professor-centered activities. Nowadays, some other trends in teaching are gaining importance, and students are now considered the focus of their own learning. In this context, student-centered activities are considered as important issues for teaching software engineering, and playful activities can play a crucial role for promoting the learning-by-doing approach needed by the students.

In this paper we propose four playful activities to be applied to a group of undergraduate students belonging to the system engineering program. We include the following: systems fair; software catastrophes—eight bases; information system case studies—creative presentations; and lottery about ethics in information systems. With such activities, we aim to streamline the software engineering teaching process.

This paper is organized as follows: first, we present the fundamental concepts that support this work; then, we propose different playful activities for teaching software engineering; after that, we discuss some results of the playful activities application; finally, we summarize conclusions and future work.

THEORETICAL FRAMEWORK

SOFTWARE ENGINEERING

The most widely accepted definition of software engineering has been provided by the IEEE: "The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software" (IEEE, 1990). Commonly, software engineering is promoted in order to teach students some concepts and practices related to the software development process, aiming to satisfy previously elicited requirements (Laplante, 2007).

Due to the need for understanding software development as a systematic process, several universities working in this area have included in their curriculum an area focused on software engineering. As a matter of fact,

countries like USA and UK are demanding professionals certified on this area. Thus, some institutions dedicated to teaching preparation courses for such certifications are increasing day by day (Wyrostek, 2008).

Software engineering is a foundational area for any system engineer. It is a cross-sectional area comprising several fields of knowledge such as: computer science, information tecnologies and information security, among others (ACM, 2005). However, the learning process in this discipline is a non-trivial activity because some level of abstraction is required for dealing with complex problems. Such level should be acquired by the students by using active-learning strategies, *i.e.* strategies based on the active participation of the students (Morell, 2009).

PLAYFUL ACTIVITIES AS LEARNING STRATEGIES

Bushell—as cited by Dempsey *et al.* (1996)—defines gaming as an interactive activity intended to replicate expected real-world conditions. Such an activity stimulates the decision-making process. Some behavioral rules are accepted by the players in the competition, whom make decisions affecting themselves and the other players (Dempsey *et al.*, 1996). A playful activity should exhibit the following features in order to be considered "educational" (Burgos *et al.*, 2006): (i) a premise to-be-solved as a starting point; (ii) at least one true solution; (iii) at least one learning strategy—e.g., new knowledge generation, previously acquired knowledge reinforcement, skill exercise, concept discovery, creative development, or experience sharing.

According to the previous definitions, some authors propose the following advantages for using playful activities in the teaching-learning process:

1. Learning-by-doing approach. Students develop communication skills and consider emotions in the learning process. Consequently, they promote learning by peer interaction (Kober & Tarca, 2000).
2. Development of critical thinking, communication skills, debate, and decision making. From a practical point of view, learning-by-play activities are substantially different from the theoretical approach of lectures (Baker *et al.*, 2005).
3. Increase of the learning speed and improvement of the knowledge acquisition (Klassen & Willougby, 2003). Team working promotes students to achieve consensus on tasks, goals, and methods (Lainema, 2004).

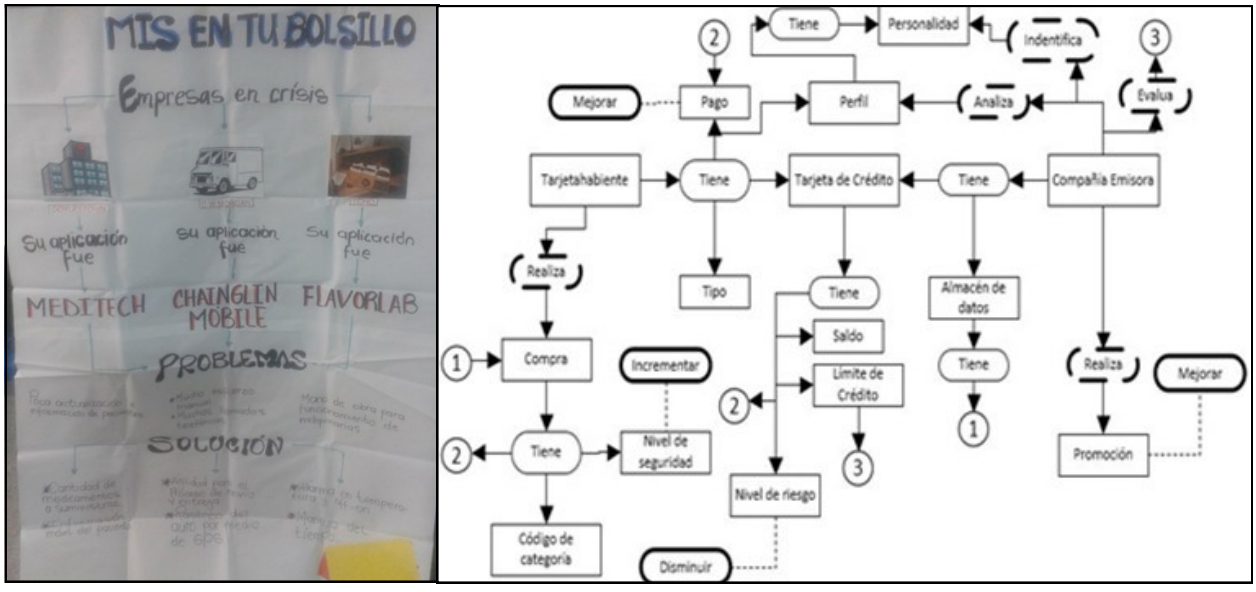
PLAYFUL ACTIVITIES IN THE SOFTWARE ENGINEERING TEACHING

COURSE CONTEXT

The playful activities were applied in the 64-hour information systems course during February to May 2013, belonging to the third semester of the System Engineering Program in the Medellín University. Students are eighteen years old in average.

The course begins with an introduction to the general systems theory focused on the explanation of principles, components, and systems properties. Then, the concepts of information systems and organizations are presented, emphasizing the importance of software application development for optimizing business processes and

EXHIBIT 1
EXAMPLES FOR REPRESENTING TWO CASE STUDIES.



achieving competitive advantage and leadership. Next, the work is centered on business process modeling for real organizations searching for the identification of problems and bottlenecks, and proposing a software solution that improves the analyzed process. Finally, information technologies comprising emerging technologies—e.g., business intelligence, cloud computing, and Web 2.0—are covered.

The professor assigns a project to be developed throughout the semester for practicing the above mentioned concepts.

PLAYFUL-ACTIVITY DESIGN

Previous design and preparation of the playful activities is one of the critical success factors for incorporating playful activities in the classroom. Three main aspects should be considered for designing an education-oriented playful activity:

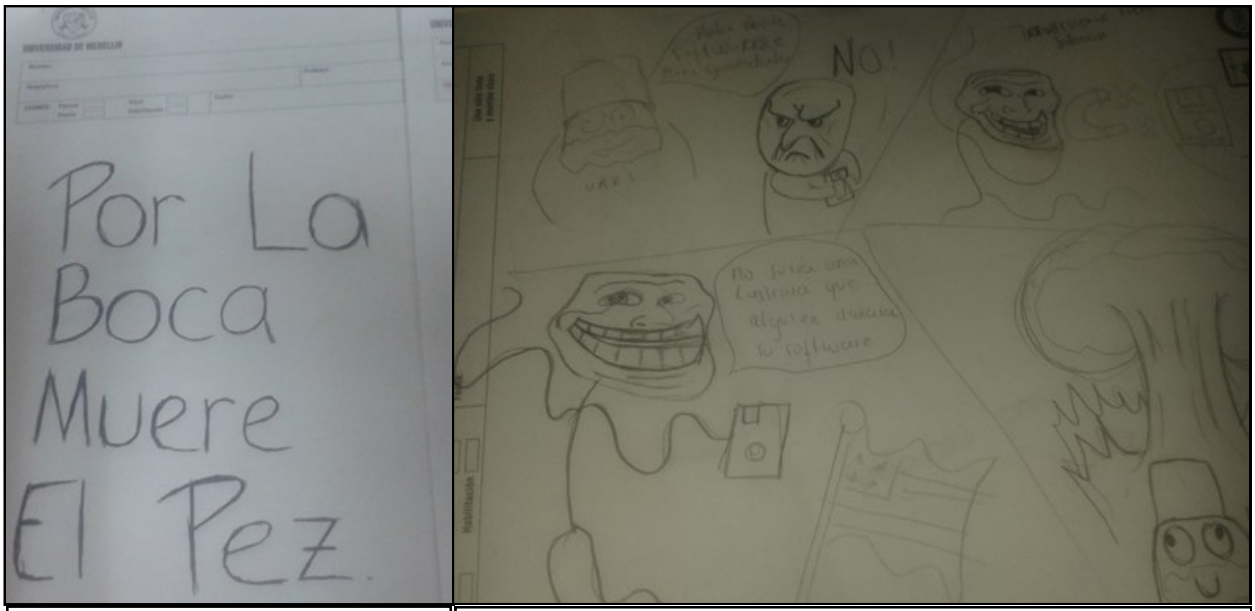
a. Defining the issues to be covered by the playful activity. Professor has a set of issues to be presented in the classroom by using several teaching strategies.
b. Establishing the purpose of the playful activity. An education-oriented playful activity should exhibit at least one of the following purposes: *teaching* (presenting the general concepts of an issue for the first time to a group of participants); *reinforcement* (presenting again an issue to a group of participants in order to reinforce the appropriation of the basic concepts by using a different strategy from the theoretical approaches), *development of creations* (using a playful activity for promoting the generation of several solutions for a problem), e*xperience sharing* (analyzing the decision making process of each participant, searching for experience sharing and collective identification of conclusions and lessons learned).
c. Defining the instructional/educational goals of the playful activity. The instructional goals expected to be achieved by the participants at the end of the playful activity related with concepts assimilation and skills development should be defined.

In this section we present the design of four playful activities incorporated in the course described above searching improve the teaching/learning process. This is the first course of the emphasis on Software Engineering in the Systems Engineering Program of the Medellín University.

SYSTEM FAIR

- Issue—components of a system according to the general systems theory.
- Purpose—reinforcing the knowledge about components of a system from a practical approach.
- Instructional goal—by the end of this activity, the student will be able to select a system and identify its main components, such as inputs, outputs, and

EXHIBIT 2
EXAMPLES OF SOFTWARE CATASTROPHES
REPRESENTATIONS DEVELOPED BY THE STUDENTS



Saying	Cartoon

structure, among others.

Following are the steps of this playful activity:

1. Organizing 4-student teams.
2. Providing each team with cardboard and two markers.
3. Each team has half an hour for:
   a. Selecting a system not included in the class examples.
   b. Identifying and explaining its components: (1) elements; (2) relationships; (3) structure; (4) goals, (5) inputs, (6) outputs; (7) environment.
   c. Drawing a graphical explanation of the components in the cardboard—text-only explanations are not allowed.
   d. Designating a team leader.
4. After the half hour, the team leader is responsible for exposing in two minutes the drawing to their classmates in some parallel classroom sessions.
5. Professor controls time and designates the next speaker.

INFORMATION SYSTEM CASE STUDIES—CREATIVE PRESENTATIONS

- Issue—real cases for incorporating information systems in organizations in order to solve problems and competitive advantages for leadership.
- Purpose—reinforcing the importance of information systems for organizations by using successful real cases.
- Instructional goals—by the end of this activity the student will be able to:
- Address an information system case study and identify the business challenges—problems—and the technological solution related.
- Speak about the answers for the questions in a case study by using alternative ways to an oral slide-based exposition.

The scheme followed for implementing this playful activity is summarized below:

1. Organizing 3-student teams.
2. Each team selects randomly one out of four case studies related to green computing, customers traceability through credit card transactions, order tracking by internet access, and business mobile applications.
3. Teams have fifteen days for answering the questions of the case study and identifying the problems and solutions from the perspective of information systems. In addition, students should prepare a creative presentation for their classmates. In this context, "creative presentations" are not intended to be slide-based presentations. Some options used by the students were, among others:

EXHIBIT 3
EXAMPLE OF THE BOARD FOR THE LOTTERY ABOUT ETHICS IN INFORMATION SYSTEMS

- Home video made by the students.
- Pre-conceptual schemas—conceptual-map-type diagrams—for presenting the case in a single diagram.
- Drawing-based cardboards representing the problem and its solution.

Exhibit 1 shows two examples of creative presentations made by the students for the assigned case study.

SOFTWARE CATASTROPHES—EIGHT BASES

- Issue—Software Catastrophes.
- Purpose—reinforcing the importance of competent software engineers committed with the development of applications. In this way, we can avoid risking human lives and wasting large economic investments. Also, we can socialize experiences related to real cases of failures in the software industry.
- Instructional goal—by the end of this activity, the student will be able to describe several software catastrophes by identifying their causes and consequences. Also, he/she will be trained for practicing some communication skills.

The following are the steps for implementing this activity:

1. Organizing 8-student teams.
2. Each team has a week for selecting a software catastrophe and planning some idea for the sake of explaining it.
3. The day of the presentation, each team is located in one of eight numbered "bases," supplied with sheets, markers, colors, and stationery tape.
4. Each team selects a base and has two minutes for representing the catastrophe by using the following strategies:

Base 1: Hieroglyph
Base 2: Saying
Base 3: Cartoon
Base 4: Draw only with animals
Base 5: Free draw
Base 6: Draw only with flowers
Base 7: Acrostic
Base 8: Logo

Once the representation is completed, each team posts the representation by using the stationery tape.

5. Finally each team has a minute for observing the representation of each base, identifying the catastrophe and choosing the best representation.

In the Exhibit 2, we can see two representations made by the students during the playful activity.

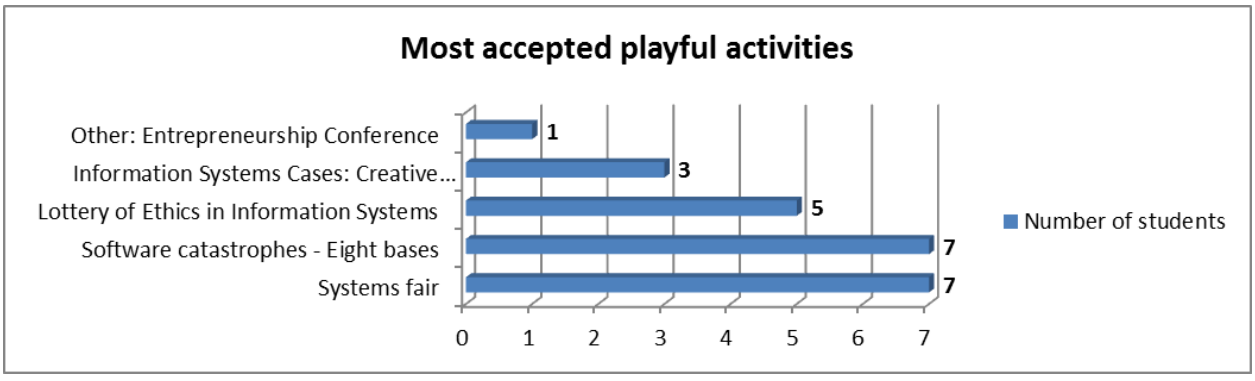
LOTTERY ABOUT ETHICS IN INFORMATION SYSTEMS

- Issue—ethics in information systems.
- Purpose—teaching the fundamental ethical concepts in the development and management of organizational information systems.
- Instructional goal—by the end of this activity, the student will be able to define some of the main concepts about ethics, e.g., patent, intellectual property, and copyright, among others.

The steps followed for the design and implementation of this playful activity are summarized below:

1. Reading the material associated with Ethics and Information Systems.
2. Identifying relevant concepts to be included in the lottery.
3. Developing a list of concept-definition pairs.
4. Design the boards with the selected concepts (see Exhibit 3).

EXHIBIT 4
MOST ACCEPTED PLAYFUL ACTIVITIES FOR THE STUDENTS BELONGING TO THE INFORMATION SYSTEMS COURSE



5. Once the lottery has been designed, the professor promotes the lottery practice in the classroom as follows:
   a. Organizing 2-student teams.
   b. Each team receives a board and a card with the team number.
   c. The professor starts to read the definition of a concept and the first team to raise the numbered card has the opportunity to guess the concept. If the answer is right, the team receives the card with the concept; otherwise the opportunity is given to another team.
   d. The winner team is the one guessing most of the concepts.

RESULTS

Some retrospective should be conducted by the end of the playful activities to the extent of obtaining feedback from the students. We aim to validate the fulfillment of the instructional goals and collect the student perception about the playful activity focused on realism, fun level, and simplicity, among others. In this section we analyze the answers.

Question 1: Select the playful activity you enjoyed most:

- ○ Systems Fair
- ○ Software Catastrophes—Eight bases
- ○ Lottery about ethics
- ○ Information Systems Cases—Creative Presentations
- ○ Other? Which?_____

The most accepted playful activities by the students are: Systems Fair, Software Catastrophes bases and the Lottery about ethics (see Exhibit 4). This result is probably due to the fact that such activities promote expressing concepts by using different strategies from verbal language. In the beginning, some difficulties arise, but later the students could use their creativity and skills for creating non-verbal representations.

Question 2: The learning level achieved by practicing playful activities in the classroom, according to your point of view is:

- ○ Excellent
- ○ Good
- ○ Fair
- ○ Deficient

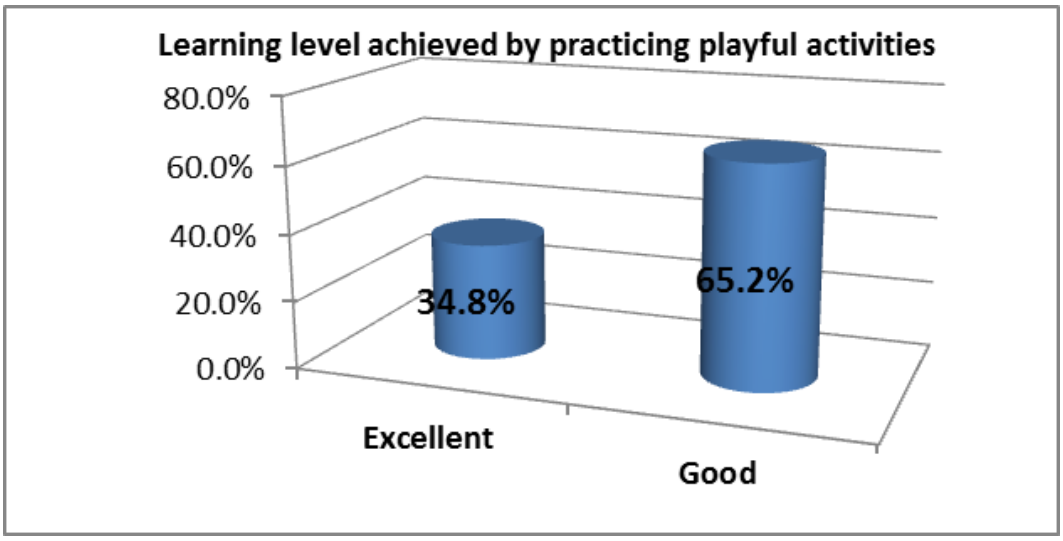
Students had a high level of acceptance related to the practice of playful activities in the classroom, as shown in Exhibit 5. This fact reinforces the value the students assign to the incorporation of such activities in the classroom as an strategy for improving the teaching/learning process.

Question: The amount of playful activities used in this course during the semester, according to your point of view is:

- ○ Excessive
- ○ Adequate
- ○ Insufficient

This question also reinforces the high level of acceptance of the playful activities by the students: most of them (21 of 23 in Exhibit 6) consider *adequate* the amount of this type of activities to be applied within the course and two of them consider the amount *insufficient*. None of the

EXHIBIT 5
LEARNING LEVEL ACHIEVED BY PRACTICING PLAYFUL ACTIVITIES



students consider *excessive* the amount of playful activities, leading to an implicit acceptance of the playful activities by the students, since they are well-designed and previously tested for avoiding improvisation.

Question 4: Do you think the playful activities in the classroom are useful for your learning process?

- Yes
- No

In this question the answer was unanimous, since all of the students answered affirmatively emphasizing the utility of playful activities in their learning process. This fact is consistent with the answer to the second question, i.e., students consider their learning level as either good or excellent.

Question 5: Would you like the incorporation of playful activities in other courses of the system engineering curriculum?

- Yes
- No

In this case the situation is identical to the previous question because all of the students agreed on wishing to incorporate playful activities in other courses of the curriculum. This fact represents a challenge for professors, because it represents an invitation to continually rethink the courses for gaining awareness about the current learning styles of the students.

Question 6: What would you change to the playful activities practiced in the classroom?

74% of students (17 of 23 in Exhibit 7) consider no changes are necessary for the playful activities. However, six students mentioned some changes to be considered:

- Investing more time for the application of the playful activities, since sometimes was very limited.
- Predicting the enthusiasm generated by the playful activities. When competition is involved, some mechanisms to avoid disorder are needed. Also, we need more objective criteria of valuation.
- Increasing the participation and discussion by the students for gaining more feedback.
- Increasing the complexity of the playful activities and incorporating more real life examples in practicing them.

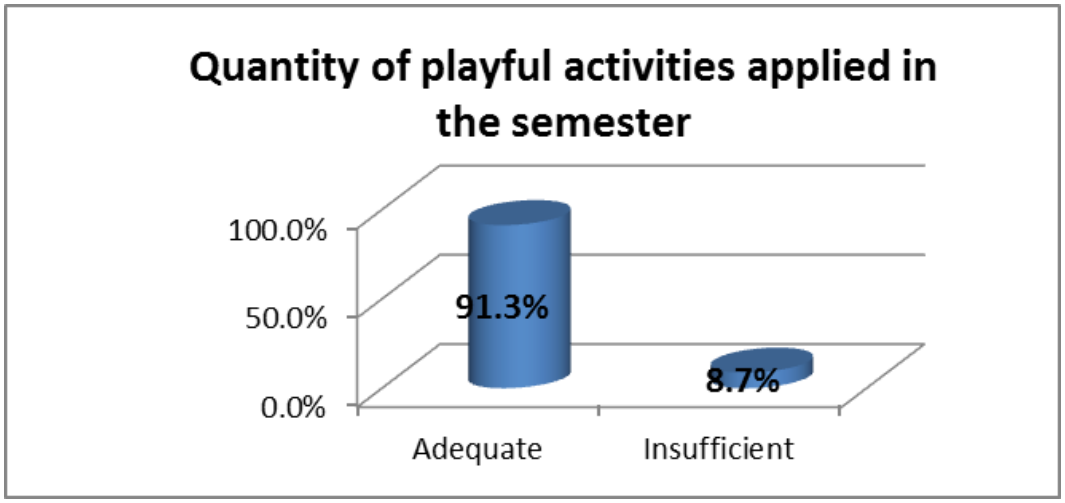
Be advised that one of the students suggested: "What we really need is... more playful activities!" Answers lead us to think that by practicing playful activities in the classroom we can improve the learning process and increase the speed of learning and the recall of concepts.

CONCLUSIONS AND FUTURE WORK

Nowadays, students are surrounded by an environment full of information and interactive media. For this reason, we need to improve the traditional teaching strategies by adding some other alternatives, where the student can play an active role instead of only being a spectator. In addition, playful activities allows for the students to develop other key skills for a software engineer: leadership, good communication, and negotiation. The playful activities promoted in this paper were well accepted by the students and promoted the discussion in the classroom.

Future work is devoted to promote this initiative in some other courses of the software engineering curriculum. We aim to complement the well-known teaching strategies,

EXHIBIT 6
QUANTITY OF PLAYFUL ACTIVITIES APPLIED IN THE SEMESTER



instead of replacing them. We want to increase student motivation, promote meaningful learning, and develop social skills in the training process of the students.

REFERENCES

ACM (2005).*Computing Curricula 2005: The Overview Report*.

Baker, A., Navarro, E. & Van Der Hoen .(2005). An experimental card game for teaching software engineering processes. *The Journal of Systems and Software*, (75), 3–16.

Burgos D., Tattersall C. & Koper R. (2006). Can IMS Learning be used to Model Computer-based Educational Games?. *Binaria Magazine*

Dempsey J., Rasmussen K. & Lucassen, B. (1996). The Instructional Gaming Literature: Implications and 99 Sources. *COE Technical Report*, 96(1), College of Education. University of South Alabama, USA.

IEEE.(1990). *IEEE Standard Glossary of Software Engineering Terminology*.IEEE Std 610.12-1990, pp. 1 -84

Klassen K. & Willoughby K. (2003). In-Class Simulation Games: Assessing Student Learning. *Journal of Information Technology Education*, 2, 1–13.

Kober R. & Tarca A. (2000). For fun or profit? An evaluation of a business simulation game. *Accounting Research Journal*, 15, 98–111.

Lainema T. Redesigning the Traditional Business Gaming Process—Aiming to Capture Business Process Authenticity. *Journal of Information Technology Education*, 3, 35–52.

Laplante, P.A. (2007). *What Every Engineer Should Know about Software Engineering*: Taylor & Francis.

Morell, T. (2009). *¿Cómo podemos fomentar la participación en nuestras clases universitarias?*. Universidad de Alicante. Departamento de Filología Inglesa

Wyrostek,W.E. (2008). *The Top 10 Problems with IT Certification in 2008.*

EXHIBIT 7
CHANGES TO BE INCORPORATED IN THE PLAYFUL ACTIVITIES