

Communication and traceability game: a way to improve requirements elicitation process teaching

El juego de la comunicación y la trazabilidad: una manera de mejorar la enseñanza de la educación de requisitos

*Carlos Mario Zapata Jaramillo**

*Grupo de Investigación en Lenguajes Computacionales, Escuela de Sistemas, Bloque M8A Oficina 310, Universidad Nacional de Colombia, Medellín, Colombia

(Recibido el 9 de julio de 2009. Aceptado el 18 de mayo de 2010)

Abstract

Requirements elicitation process is important to guarantee the quality of software applications. In software engineering, requirements elicitation process has been taught by means of traditional methods. Games have been used to improve teaching of some issues in software engineering, but elicitation has not been the center of this way of teaching. As a way to deal with problems in requirements elicitation process teaching, we create in this paper “Communication and Traceability Game” and we summarize the results of applying this game to several groups of students.

----- *Keywords:* Requirements elicitation process, teaching by games, communication, traceability, active learning

Resumen

El proceso de educación de requisitos es importante para garantizar la calidad de las aplicaciones. Este proceso se suele enseñar, en ingeniería de software, empleando métodos tradicionales. Los juegos al interior de la ingeniería de software se vienen usando para mejorar la enseñanza de algunos temas particulares, pero el proceso de educación no se abordó hasta el momento. Como una manera de lidiar con los problemas en la enseñanza del proceso de educación de requisitos, en este artículo se propone el “juego de la comunicación y la trazabilidad” y se compendian los resultados de la aplicación de este juego a varios grupos de estudiantes.

---- *Palabras clave:* proceso de educación de requisitos, enseñanza mediante juegos, comunicación, trazabilidad, aprendizaje activo

* Autor de correspondencia: teléfono: 57 + 4 + 425 53 50, fax: 57 + 4 + 425 53 65. correo electrónico: cmzapata@unal.edu.co. (M. Zapata)

Introduction

Software Engineering is a discipline that applies software development methods in order to improve the quality of the resulting software applications [1]. One of the first stages of software engineering is the requirements elicitation process, which comprises requirements capture and transformation into software specs [2]. Software Engineering teaching has been traditionally made by means of lectures and practical projects, as a way to practice the needed skills for this process [3]. However, requirements elicitation process demands some skills of its practitioners, which are difficult to develop when is taught by traditional means [4]. For example, “communication” issues are almost impossible to teach by means of lectures, due to the fact that they need big amounts of expertise in order to be used in an appropriated way in the requirements elicitation process. Something similar occurs to “traceability” issues, which are complicated even to the most experienced software analysts. Practical projects exhibit contradictions when used for teaching this kind of issues: they need analyst’s experience to be properly conducted, but they are conducted in a simulated way by non-experienced students with learning purposes.

Some researchers in several fields of knowledge are proposing the use of games for improving the learning process. For example, in management, games like “beer game” [5] and “beefeater” [6] are proving to be effective tools to simulate the behavior of actors in some environment. In Software Engineering, games like “Problems and programmers” [3], “Requirements Game” [7], and “Consistency Game” [8] have been used to teach some of the skills needed by the future software engineers, but they lack the communication and traceability component that is crucial for a good requirements elicitation process.

We create, in this paper, “Communication and Traceability Game”, a special game that reinforces teaching of two of the most common problems in requirements elicitation process: requirements traceability and communication among stakeholders.

The structure of the rest of this paper is the following: in Section 2 we describe the notions of requirements elicitation process (focusing on communication, and traceability) and games for teaching, in Section 3 we present previous work in software engineering games, in Section 4 we define “Communication and Traceability Game”, in Section 5 we summarize the results of playing the game in several groups of students. Finally, in Sections 6 and 7 we conclude and we present future work.

Theoretical framework

Elicitation process requirements

Software specs are the result of a carefully conducted process. In this process, called “requirements elicitation process”, analysts interview stakeholders (people with some concern in the software development process) in order to gather information about the future software application [2]. The gathered information is then translated into several diagrams, which reflect the needs and expectations of the stakeholder. Diagrams are intermediate products of the software development process, and they are representations of the required software application.

Requirements elicitation process exhibit problems like differences in the language of the actors of the process (stakeholders are domain experts while analysts are modeling experts), informal approaches to the information gathering process, and misconceptions of the stakeholder about the software to be constructed [4]. These problems lead to a poor-quality requirements elicitation process, and they need to be solved if you want to build the software that stakeholders need.

Pressman [1] defines customer communication as a set of “tasks required to establish effective communication between developer and customer” (page 36). In this sense, the above mentioned problems are results of ineffective communication practices among software development teams (including stakeholders as team members) and often this is an overlooked issue in requirements elicitation process [4]

According to Pressman [1] (page 511) traceability is the “The ability to trace a design representation or actual program component back to requirements.” Traceability problems arise as a consequence of misunderstanding information during requirements elicitation process and, consequently, can be related to communication issues of such process.

Games for teaching

Huizinga [9] showed some of the benefits of using games as a didactic method. Games are funny experiences, Huizinga says, and we can create complete simulated worlds by means of games. Some other researchers proposed games as a way to improve the teaching process. They explore the advantages of games for this process: the simulation of real experiences with safety and reality, with the purpose of reproducing the behavior of actors in the simulated environment, the reinforcement of knowledge by means of vivid experiences, and the power of funny activities to keep the attention of the players.

The use of games in software engineering teaching

Traditionally, lectures and practical projects have been the preferred strategies for teaching software engineering [3]. These strategies have proved to be useful to spread concepts and to practice technical skills, but the reinforcement of managerial and team work skills is difficult to achieve with such strategies. Games are non-traditional strategies, and they belong to an “active” way of practicing some skills [10].

Management is probably the most common source of games for teaching purposes. The “beer game” [5], for example, as described by Senge, recreates the world of inventory in order to permit simulation of the behavior of several actors belonging to this environment. In the same line of thought, the microworld “beefeater” [6] permits to its practitioners to simulate the environment of rapid-food restaurants.

Related to software engineering, there are few game experiences. The first of such experiences is “Problems and programmers” [3], a card game for practicing software development issues. Other experiences are “Requirements game” [7] and “Consistency game” [8], two in-class non-technological games for teaching the first phases of software development.

Despite the efforts of these games, communication and traceability are issues still untreated in teaching game experiences.

Communication and traceability game

We try, with this game, to improve the teaching experiences related to requirements elicitation process and, more specifically, related to communication and traceability, two of the most important issues of such process. Next, we define the main features of the game.

Goal of the game

The aim behind this game is to correctly finish (and within the lower possible time) a set of artifacts related to the development of a software application. In this case, the artifacts are three use case diagrams, one class diagram, and one graphical user interface. Also, the model behind these artifacts is the well-known youth card game Yu-gi-oh!®. We use this game because is easy to understand by young people, whom are the target practitioners of the game.

Materials

- Game board: Is a spreadsheet of rows and columns labeled as we show in figure 1. The game board can be built in paper, carton, wood, or any other material.
- Game tokens: Are several diagramming elements belonging to every diagram. Tokens have to be disposed on the game board when the game is played. Game tokens have different shapes depending on the target diagram:

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							

Figure 1 Game board of the traceability and communication game

- Use case diagrams: tokens can be actors, use cases, communication lines, and system boundary lines.
- Class diagram: tokens can be class names, sets of attributes, sets of operations, and association lines.
- Graphical user interface: tokens can be heading lines, monster zone cards, magic zone cards, hand cards, deck zone cards, and cemetery zone cards.
- Artifact specs: Are the finished drawings of the diagrams and the graphical user interface. These drawings are tailored to a board with similar cells to the figure 1. The five available artifacts for this game are shown in figure 2.

Game rules

- Players are divided into 5-member teams. A captain and a leader must be identified. The rest of the members are recognized as co-workers.
- The physical location of the game must be organized as shown in figure 3. For every cycle of the game, the game director must place one of the artifact specs in the diagram zone. Tokens belonging to all of the diagrams are located into the store, in no particular

order. Game board must be located into a desk between the captain and the leader of every team.

- At the beginning of the first cycle, game director puts the first artifact specs in the diagram zone and returns to their position in the game.
- The captains go to the diagram zone to see and memorize the artifact specs. Then, captains return to the desk and inform the leaders about the tokens needed to replicate artifact specs and their labeled positions in the game board.
- Simultaneously, leaders give, also by voice, instructions to the co-workers, in order to gather all the needed tokens to finish the specs.
- Co-workers go to the store and take one token per trip. Then, they return to the location of the leader and give him/her the token. This process must be repeated until all of the needed tokens are completed.
- At their locations, captain and leader discuss the position of the tokens by using the labels of the game board cells. When the leader believes that the artifact specs in his/her game board are exactly the same than artifact specs in the diagram zone, he/she calls the game director to verify if this is true. If so, game director register the order of finishing the artifact specs. The cycle of the game ends with the order of all the teams ending this task.
- The game rules are repeated by four more cycles, changing the artifact specs in every cycle. The winner team will be that with the lowest average order to finish all of the artifact specs.

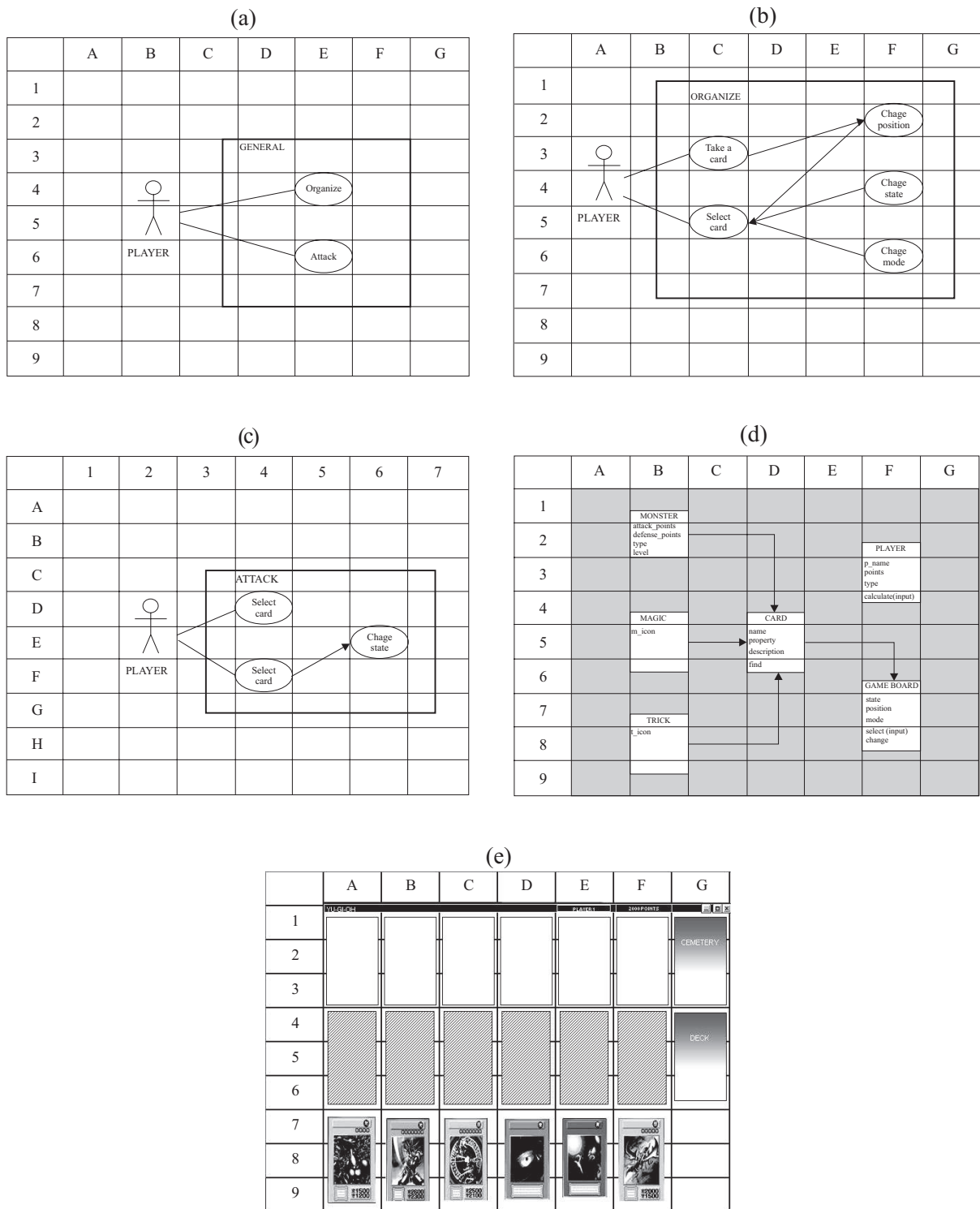


Figure 2 Available artifacts of the traceability and communication game

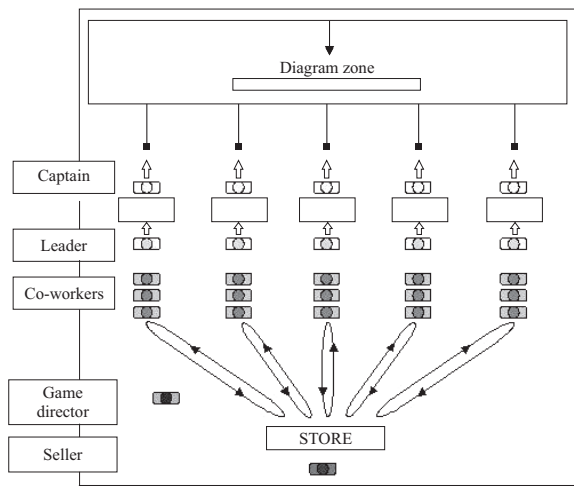


Figure 3 Physical location of the game

Results of playing “communication and traceability game”.

This game is a simulation of a real requirements elicitation process, because the way players drive

the process of finishing the artifacts is similar to the interviews among analysts and stakeholders in order to specify software specs. Here, communication is represented by the information and the instructions that players say to each other. Traceability inside the game is represented by the capability of recognizing the same model on every artifact. As a consequence, the game is designed to reach two goals: prioritizing the importance of communication and understanding the main ideas behind the concept of traceability. The game has been played by two groups of people belonging to the Universidad Nacional de Colombia, as a part of the course “Requirements Engineering”. Some of the features of the groups are summarized in table 1. The amount of players is still low (37 people) and the features of the two groups are still so similar, but we expect that this experience can be replicated in several environments to explore the differences with the current players.

Table 1 Features of the players of “communication and traceability game”

Semester	Course	Number of students	Level
2006-03	Requirements Engineering	18	6th to 9th level of Systems and Informatics Engineering.
2008-01	Requirements Engineering	19	8th and 9th level of Systems and Informatics Engineering.

Feedback to the game was obtained from a survey that players must answer at the beginning and at the end of the game. The requests were the following:

- (1) Please mention three aspects that you believe are fundamental on requirements elicitation process.
- (2) What do you understand when we mention the word “traceability”?

The requests are related to every one of the mentioned goals of the game and they are intended to “measure” the features of the main concepts of the game. In this way, first request points at the importance of communication and

the second one points at the understanding of the traceability concept. Answers to the first request are summarized in table 2. From this table, we can deduce the changes in the opinion of the players after the game. For example, the aspect “communication” mentioned as fundamental by 27% of the players before game increased its participation to 68% after the game. This change is probably due to the difficult situation that the players must affront during game, in which they could not freely communicate with their partners in the game and, consequently, they suffered delays in completing the diagrams. “Consistency”, the second most mentioned aspect by players before game (30%), felt to 16% of the opinions after the game. This

fact does not mean that this aspect has lower importance in requirements elicitation process, but it must be combined with other aspects like “communication”. “Problem understanding” and “information clarity”, the two most important aspects mentioned by players before game, after the game they were still important, but with a

lower participation of the opinions. “Regarding” is a new opinion of the players after the game. This aspect seemed not to be important before the game, but the situations that players must go through the game were sufficiently convinced to suggest this new aspect as important to requirements elicitation process.

Table 2 Summary of answers to the first request

<i>Main aspects of the requirements elicitation before game</i>	<i>%</i>	<i>Main aspects of the requirements elicitation after game</i>	<i>%</i>
Problem understanding	67	Communication	68
Consistency	30	Problem understanding	43
Information Clarity	27	Information Clarity	24
Information completeness	27	Regarding	19
Communication	27	Consistency	16
Modeling	17	Completeness	16
Representation Artifacts	13	Interpreting	16
Requirements	13	Coordination	16
Actors	13	Organization	14
Processes	10	Representation artifacts	11
Leadership	7	Traceability	5
Traceability	3	Development method	5
Responsibility	3	Leadership	5
Persistency	3	Team work	5
Planning	3	Requirements	5
Time period	3	Modeling	3
Costs	3	Persistency	3

The answers to the second request were analyzed by a software engineering professor, in order to assign a grade mark from 1 to 5 depending on the quality of every answer of the players, perceived against the provided definition of traceability. In this context, the design representations are the use case and class diagrams, and the software component is the graphical user interface. Similarly, the requirements are the information

and instructions provided by the players during the game. The results were an average grade mark of 1.6 before the game and an average grade mark of 2.7 after the game. This means an increment of 69% in the perceived grade mark assigned by a professor to the traceability definition. Note that grade mark average is still low compared to the upper value (5), but we must remember that the goal of the game is to improve

the teaching experiences about requirements elicitation process, not to replace the traditional teaching of this issue.

Conclusions

Communication and traceability are two of the most important concepts of requirements elicitation process. These concepts have been traditionally taught by means of lectures and practical projects, which emphasize on technical skills rather than social and managerial skills. Following collaborative learning trends, common in several fields of knowledge like management, we created in this paper “communication and traceability game” and we applied it to two groups of Requirements Engineering students.

The results of the application of the game are promising because the practitioners of the game have reinforced the importance of communication among participants of the software elicitation process and they have clarified the definition of traceability. Despite of some difficulties experimented during the game application, we conclude that this game can be used as an alternative and “active” learning experience for complementing the traditional way of teaching software engineering.

Future work

Some work still has to be done in this area. For example, the way in what we select the main variables and elements of the game was completely subjective, linked to the experience of the author in requirements elicitation processes. Gaming is a good learning strategy to complement traditional teaching, but we need to create a more objective methodology to define the main features of the game, in order to improve the quality of the designed games.

Related to “communication and traceability game”, we need to play the game with another type of players, for example people from the software industry, in order to compare the results with more skilled people in requirements

elicitation process. It is important to note that this game is currently used in a special project with children called “science garden”. The main goal of this project is showing several aspects of science research to children, and software engineering is one of the areas of interest of the project.

Acknowledgments

I wish to thank the work of three excellent students behind this game: Natalia Gil, Yudi Jiménez, and Hernán Soberón. Their support in conceptualizing and creating the game was crucial to the results of applying this game. This work is founded by the Vicerrectoría de Investigación from Universidad Nacional de Colombia, under the project: “*Software Boulevard un juego de estrategia en Web para la enseñanza de competencias de gestión en Ingeniería de Software*”, Project number 9766.

References

1. R. Pressman. *Software Engineering. A Practitioner's Approach*. Ed. McGraw Hill. New York. 2001. pp. 3, 36, 511.
2. J. C. Leite. “A survey on requirements analysis, Advanced Software Engineering Project”. *Technical Report RTP-071*, Department of Information and Computer Science. University of California at Irvine. 1987. pp. 26.
3. A. Baker, E. Navarro, A. Van der Hoek. “An experimental card game for teaching software engineering processes”. *The Journal of Systems and Software*. Vol. 75. 2005. pp. 3-16.
4. M. Christel, K. Kang. “Issues in Requirements elicitation”. *Technical Report CMU/SEI-92-TR-012 ESC-TR-92-012*. Software Engineering Institute. 1992. pp. 10-23.
5. P. Senge. “The Fifth Discipline”. *The art and practice of the learning organization*. New York. Doubleday. 1990. pp. 27-54.
6. Strategy Dynamics: “Beefeater Restaurants Microworld”. <http://www.strategydynamics.com/products/beefspec1.asp>. Consultada el 8 de Julio de 2009.
7. C. Zapata, G. Awad. “Requirements Game: Teaching Software Project Management”. *CLEI Electronic*

- Journal*. Vol. 10. 2007. <http://www.clei.cl/cleiej/paper.php?id=133> Consultada el 10 de Noviembre de 2009.
8. C. Zapata, M. Duarte. "El juego de la consistencia: una estrategia didáctica para la Ingeniería de Software". *Revista Técnica de Ingeniería de la Universidad del Zulia*. Vol. 31. 2008. pp. 1-10.
 9. J. Huizinga. "Homo ludens. *A study of the play-element in culture*". Ed. Beacon Press. Boston. 1955. pp. 1-10.
 10. M. Pivec, O. Dziabenko, I. Schinnerl. "Aspects of game-based learning". *Proceedings of I-KNOW 03, the Third International Conference on Knowledge Management*. Graz. Austria. 2003. pp 217-224.