The use of Comic Strips in the teaching of Software Engineering

Diomara Martins Reigato Barros, Luiz Ricardo Begosso
Fundação Educacional do Município de Assis
Centro de Pesquisas em Informática
Assis, Brazil
barrosdiomara@gmail.com, begosso@femanet.com.br

José Augusto Fabri, Alexandre L´Erario
Universidade Tecnológica Federal do Paraná
Departamento Acadêmico de Computação- DACOM
Cornélio Procópio, Brazil
{fabri,alerario}@utfpr.edu.br

Abstract - **Computer Science students usually carry practical activities for the identification of software requirements and for understanding the organization business rules. Within this context, during the last two years we have conducted a project with software industry and Computer Science students, using comic strips to support the software requirements specification. We created a method of scenario simulation for the use of comic strips and a language for requirements specification using comic strips. We also proposed a Guide for comic strips to support students and professionals creating software requirements. For the academia, we conducted two experiments, to verify the students' understanding of software requirements using comic strips and to get students to create comic strips for specifying software requirements using our methodology. The results of these experiments were evaluated very positively. This work aims to present our methodology, the grammar language and a guide for the creation of comic strips, and the results of the experiments conducted with Computer Science students.**

Keywords – Comic Strips; Software Requirements; Software Engineering; Computer Science Education.

I. INTRODUCTION

Teaching concepts of elicitation of software requirements to Computer Science students has been a major challenge, because students often have difficulty understanding user needs.

It is known that the correct understanding of the requirements by the software developers is a critical factor for the success of the project, however several studies emphasize that documenting requirements is considered a great challenge [1], [2].

In this project, we propose the use of comic strips in the teaching of software requirements specification, to facilitate the understanding of the user's needs by Computer Science students.

Comic strips are advancing to the digital environment [3], and stand out as an efficient method for teaching concepts in several areas of Information Technology, such as database [4] and computer networks [5].

The identification of system requirements is not an easy task for students to understand. If they have incorrect, incomplete, inconsistent or ambiguous understanding of requirements, then it can affect negatively the quality of software to be developed.

One factor that contributes to increasing the difficulties in the understanding the software requirements is the gap between the real needs and the demand of the users. Misunderstanding the identified requirements is one of the main factors for failure, rework, schedule delays and cost increases during the software development process [6].

When students must transform the user requirements into technical specifications, they should pay close attention and do a great deal of interpretation. However, these technical specifications of requirements are not easily understood by students.

Within this context, this work aims to introduce the use of Comic Strips on helping in the teaching of software requirements specification, as well as to evaluate the results of the application of this technique.

For achieving this objective, we created a specification language for the creation of comic strips, which we named “the grammar”, and it is used for the specification of software requirements. Then we defined a guide to support students when creating their comic strips. Finally, we conducted two experiments with our Computer Science students.

In the next sections, we will discuss the background and related works, as well as the methodology we defined for the application of the experiments and then the results obtained.

II. BACKGROUND AND RELATED WORKS

Techniques of visual requirements and ludic methods can be used to increase awareness and understanding of user requirements.

Some studies have proposed the use of tools for supporting the requirement-gathering activity, such as the use of tabular views, quantitative visualizations of risk with tables, modeling

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

of requirements business processes, and the use of mind mapping. These tools have been effective for facilitating the communication between stakeholders in a software development environment [7].

Commercial Requirements Engineering tools also support the visualization requirements, such as graphical use case definition, requirements specification through scenarios, and storyboards for requirements validation [7].

Visual notations are an integral part of the language of Requirements Engineering and have dominated the research and practice of Requirements Engineering since its inception [8]. The main reason for the use of visual notations is to exploit the power of human visual processing and thus optimize human communication and problem solving.

Software development tends increasingly to be a distributed or global process with participants geographically dispersed. This scenario requires attention to three aspects identified as physical distance, temporal distance and cultural distance. It is acceptable to argue that these features will have an impact on the software process, especially at the times when there are demands for more communication and collaboration between team members. In [9], authors presented an experiment carried out in a university environment, with the purpose of getting software requirements, as well as analyzing the use of university environments to carry out these validations.

In global software development, projects usually have teams and developers distributed; communication and cooperation between stakeholders are critical for the project success. In [7], authors presented a proposal to merge the stakeholders during the requirements gathering. The idea was to stimulate the parties and increase their knowledge about the requirements, with online collaboration support and the use of visualization techniques in a web-based environment. A prototype platform was implemented and submitted to an objective-based assessment. The evaluation results show that it accomplishes the proposed objectives, which include staff involvement and a better understanding of the requirements.

Considering that existing approaches for requirements specification are not completely effective, another strategy for documenting requirements includes the use of storytelling-based methods [10]. This strategy leads to specify the user needs in a complete, consistent and correct manner.

On [10], authors have proposed a case study with the aim of investigating how storytelling can be effective in the induction and development of requirements. They reported on an experiment that involved twenty-five specialists from various industrial companies, to gather software requirements using storytelling technique for a ticket machine case. The authors investigated the effectiveness of using a storytelling technique compared to a traditional brainstorming one; they concluded that the quality and details of the requirements gathered using the storytelling approach were better than those gathered using traditional approaches such as brainstorming.

In [11], authors argue that the use of Comic Strips for teaching computing concepts motivates students to participate more during the classes.

In [12], the author is a biology professor, who has used pictures and drawing to bring students closer to the natural world and to help them deepen their understanding of form. The author concluded that he could transfer the visual narrative of biology, using drawing and comic strips to anatomy classes and to surgical residents and medical students.

In [5], the author used comic strips as a supplementary material for the course of Computer Networks, to facilitate the students' understanding about the concepts. The method was applied to the students of the third year of Computer Engineering and it was observed that the learning degree for the students exposed to comic strips was better, compared to the students taught with traditional methodology. The use of comic strips adds amusement elements to the teaching-learning process.

According to the studies described on this section, visual notations and comic strips representations play an important role in communicating with end users and clients.

In the next section, we will describe our first experience using comic strips for the specification of requirements.

III. REQUIREMENTS SPECIFICATION USING COMIC STRIPS

We carried out two controlled experiments with professionals of software industry to verify that we can use comic strips for the specification of software requirements [13]. Our main purpose on these experiments was to find out if IT professionals can specify software requirements using comic strips. We also intended to check if they can analyze the comic strips written by their colleagues and identify actors, objects, processes and software requirements.

Our conclusion with these experiments was that it is possible to specify software requirements using comic strips. By our results, we could see that the identification of the process was a simple task for the professionals; they were able to identify the actors, but 63% of the participants had difficulties in identifying the objects. For example, some stories were created wrongly and some objects were connected to speech bubbles. We believe that some difficulties were because these experiments were executed without establishing a formal method for creating the comic strips.

So, we decided to create a grammar for comic strips language and a guide for the creation of comic strips. This methodology will be described in the next section.

IV. METHODOLOGY

A language is characterized as a communication protocol, which has syntactic and semantic rules. The language must have an alphabet, composed of valid symbols [14]. In our work, we propose to use a language for comic strips, defined by a set of active characters, forms of communication, objects and textual elements.

We have defined a grammar to describe the syntax of the language for comic strips, which we named by CSG – Comic Strips Grammar – and has been formally described as:

CSG = (Rules, Alpha, CSR, Story), where:

Rules = {story, characters, communication, objects, text, Pc, Po, C, O, Ot, Pt, T}

Alpha= {

}

CSR = is the grammar rules presented on Figure 1.

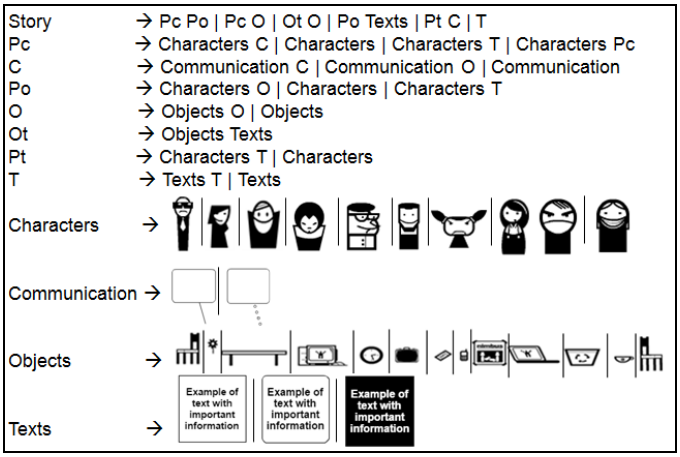


Figure 1 – Grammar defined for Comic Strips

Each line of Grammar defined in Figure 1 represents a Grammatical Rule, where each rule will be deployed in a chain of symbols (terminals and non-terminals).

The rule defines the structure whose name is to the left of the arrow and this structure is defined by choosing an option to the right, separated by the vertical bars.

Grammatical rules determine the valid chains of symbols by making use of derivations. A derivation is a sequence of substitutions of rules by choices to the right of grammatical rules. A derivation begins with a single name of Rule, which we call the initial symbol of grammar. At each step in the derivation, a single substitution is made based on a choice of grammatical rule.

As it is defined in the formalization of grammar, "Story" is the initial symbol. It is through this rule that the language begins to be described. In this way, "Story" can be derived into five options that are separated by the vertical stroke in Grammar.

According to the defined grammar, the comic strip of Figure 2, created from the grammar, can be syntactically analyzed.

Figure 2 – Comic strip of CSG Grammar

A derivation of the CSG Grammar is presented in Table 1, in accordance to the grammatical rules.

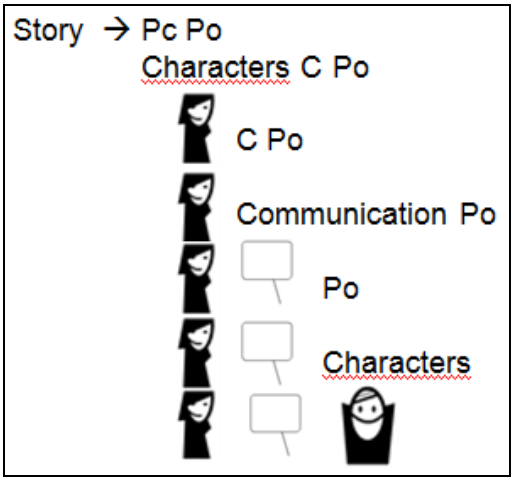


Table 1 – Derivation of the CSG from Figure 2

We decided to establish a guideline for the creation of the comic strips, to facilitate its production, which will be described below.

A story needs to be well told, so that it can be well understood. We know that a comic strip needs to have a beginning, middle and an end. So, it is possible to synthesize the three acts to construct comics in: 1 - Context, 2 - Problem and 3 - Solution.

In the first act "Context", it is necessary to situate the reader about the theme, characters, space and time. This process can be divided into 4 steps: 1) What is the theme of the Comic Strip, 2) Who are the characters, 3) Where is the Comic Strip and 4) When is the Comic Strip.

In the second act "Problem", as in storytelling, it is necessary to describe in detail the reasons for the story with all the procedures involved and the difficulties. When it comes to documenting software requirements using Comic Strips, this "Problem" act should be well described, so that the reader can easily understand which problem needs to be solved. In this act, we should describe the details of the process.

In the third act "Solution", we must bring the reader a solution or conclusion to the problem. In this phase, we conclude the requirements specification and highlight the main requirements reported in the story.

After defining the Guide, we synthesized the three acts for the construction of comic strips in a summary, to make the construction of the stories easier for students. This summary is described on Table 2.

Acts for building comic strips	**Development of the Comic Strips Software Requirements**	**Information**
1 – Context	1°) **What** is the theme of the Comic Strip?	Set a Theme. Which knowledge domain will be implemented?
	2°) **Who** are the characters?	Define the actors that will interact with the system.
	3°) **Where** is the story?	Describe the environment, the scenario in which the actors are inserted. Use objects, decorations, geographic space.
	4°) **When** is the story?	Determine at what time or age the comic strip occurs.
2 – Problem	**Describe** the process, the details of the scene.	Describe the information that derives from the environment and the business.
3 – Solution	**Propose** a solution for the problem.	Finish the story. Finalize the requirements specification.

Table 2 – Summary for the Guide for construction of Comic Strips

After defining the grammar for comic strips language and the guide for the creation of comic strips, we decided to conduct two experiments with Computer Science students, to verify how students would respond to these concepts. These experiments will be described in the next two sections.

V. First Experiment with Students

We conducted this first experiment to verify if students can understand software requirements written in comic strips. We created the comic strips according to the guideline specified in Table 2 and based on the Grammar defined in Figure 1.

The experiment was applied for 77 students in the first year of Computer Science. So, these students did not have previous knowledge or experience in software requirements.

The experiment was carried out from an execution plan divided into the following steps: Definition of the Environment, Definition of Subjects, Definition of the Sample and Execution of the Experiment.

The experiment was executed in a computer lab, where students had access to computers with internet connection. We defined the following steps for carrying out this experiment:

1: Bring students together in the classroom.

2: Request each student to fill in the Characterization form, containing personal information, if he works or worked with Software Development, work place, software development experience time and if any software is used in the company where he works.

3: Provide the students with a sheet containing a comic strip (Figure 3) and another sheet containing the questionnaire with questions related to this comic strip (Table 3).

4: Ask each student to analyze the comic strip and answer the questionnaire described in step 3.

5: Collect questionnaires and comic strips.

1 – What is the theme of the story?
2 – Identify the department of the company where the story occurs.
3 – Does the story report a dialogue between an employee and a client?
4 – What kind of task the characters develop?
5 – Point the number of the comic story you identified the answer to the question #4.
6 – Point the number of the comic story that identifies the difficulty faced by the characters in the story.
7 – What is the main difficulty in their job?
8 – Is there a solution for the difficulty found in the story? Describe the solution.

Table 3 – Questionnaire about Comic Strips

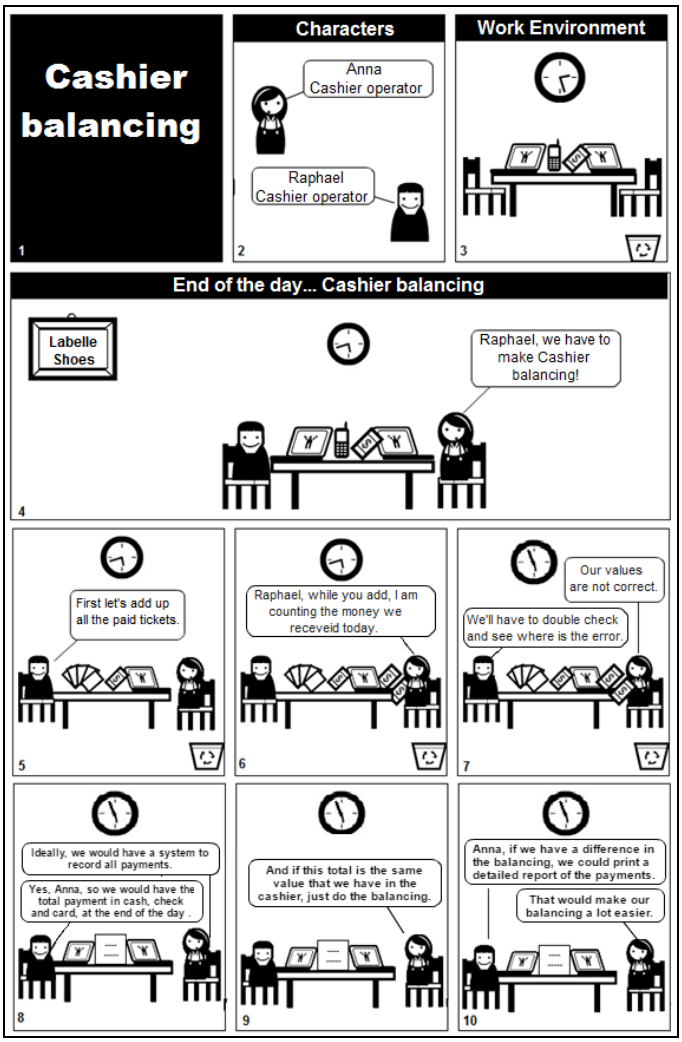
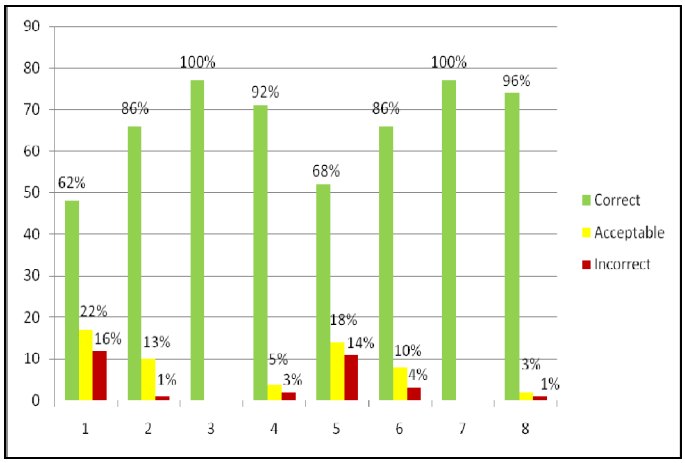


Figure 3 – Comic strip used in the experiment with students

The purpose of this experiment was to present to these 77 students a specification of software requirements which was documented using comic strips. The story specifies a cashier closure problem in a store's cashier where the characters are cashier operators and have troubles checking the cashier at the end of the day because the balance is done by hand.

After collecting the questionnaire with the answers of the 77 students, we analyzed how they interpreted the comic strips and how they understood the system requirements. Students answered the questions described on Table 3 and their responses are summarized on Graph 1.



Graph 1 – Students' understanding about Software Requirements

As we can see in Graph 1, students had good understanding about the software requirements specified using comic strips. When we analyze students' answers to questionnaire described on Table 3, we see that: 62% of students identified the theme of the story; 86% of students identified where the story occurs; 100% of students identified if story reports a dialogue between an employee and a client; 92% of students identified the task developed by the characters; 86% identified where is the difficulty faced by the characters in the story; 100% of students identified the main difficulty of characters' job; and 96% identified the solution for the problem in the story.

After conducting this first experiment with students, we decided to carry out a second experiment, with the purpose of evaluating the possibility of specifying software requirements using comic strips. This experiment will be described in the next section.

VI. SECOND EXPERIMENT WITH STUDENTS

We conducted this second experiment with students in the last year of Computer Science. Our main purpose was to verify if students could specify software requirements, creating comic strips according to the guideline specified and based on the Grammar defined for comic strips.

For this case, we worked with 24 students. As those students were at the end of the last year of Computer Science, they had already finished the Course Conclusion Work and were almost entering the labor market. For these reasons, we considered those students as pre-professionals.

In this experiment, students received a copy of the Grammar defined for Comic Strips and we explained them the rules of Grammar and some examples of requirements, specified using comic strips.

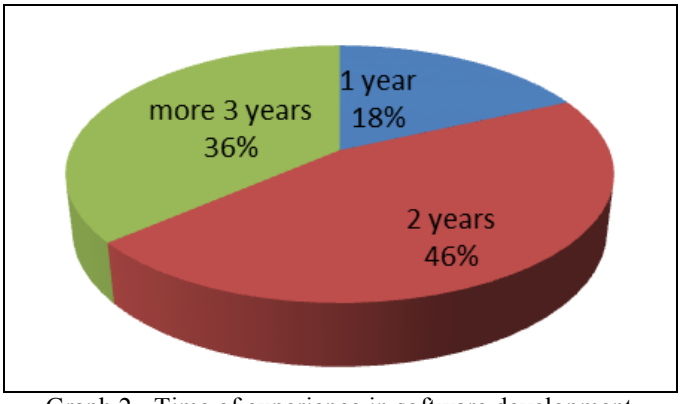
We also presented to the students the guidelines for the creation of comic strips, as defined in section IV of this paper, with the purpose of supporting the students when creating

their comic strips. Then we asked the students to write the requirements specification using comic strips. We defined the following steps for carrying out this experiment:

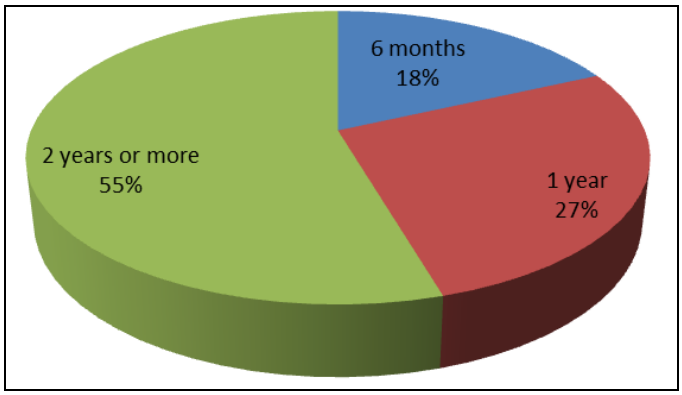
1: Bring students together in a classroom with computers and internet connection.
2: Introduce students to the Grammar and the Guide for the language of comic strips.
3: Request each student to fill out the Characterization form.
4: Provide the students with a sheet containing the guidelines for creation of comic strips.
5: Ask each student to go to the website http://stripgenerator.com and create a user account.
6: Introduce the students to the comic strips development website (http://stripgenerator.com).
7: Ask each student to develop a comic strip, within their work environment, with the objective of mapping a business process, highlighting the requirements throughout the story. This story should be in accordance with the Grammar presented.
8: Ask each student to save the comic strips and send it to the professor.
9: Ask the students to answer two questions: 1 - Did you understand better the concept of Requirements Specification using comic strips? 2 - What do you think about the experience of Specifying Software Requirements using comic strips?

It is important to point out that all students who participated in this experiment had already been working or internships in IT companies. From the characterization form, it is worth to highlight the answers for two questions: "Time of experience in software development" (Graph 2) and "Time of experience in requirements specifications" (Graph 3). With these answers, we can identify the students' experience time in software development and requirements specifications.

We can see that 82% of students had more than two years of experience in software development (Graph 2), and 82% of students had more than one year of experience in requirements specifications (Graph 3). These data are important for analyzing the comic strips created by the students, and the answers to the questions about specifying requirements using comic strips.



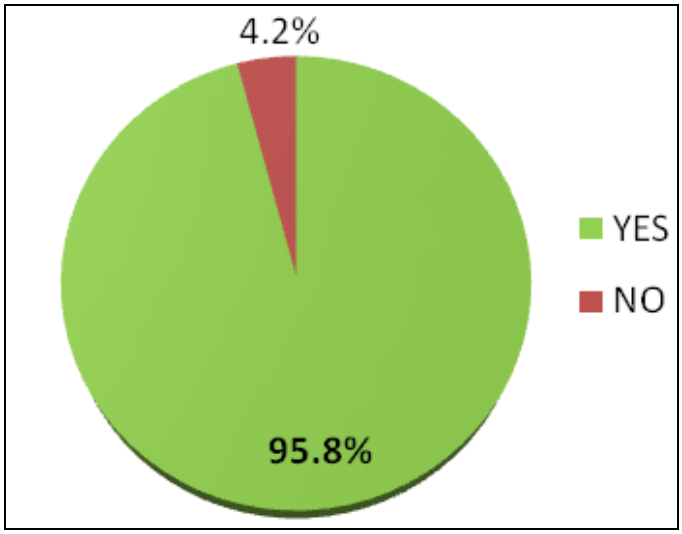
Graph 2 - Time of experience in software development



Graph 3 - Time of experience in requirements specifications

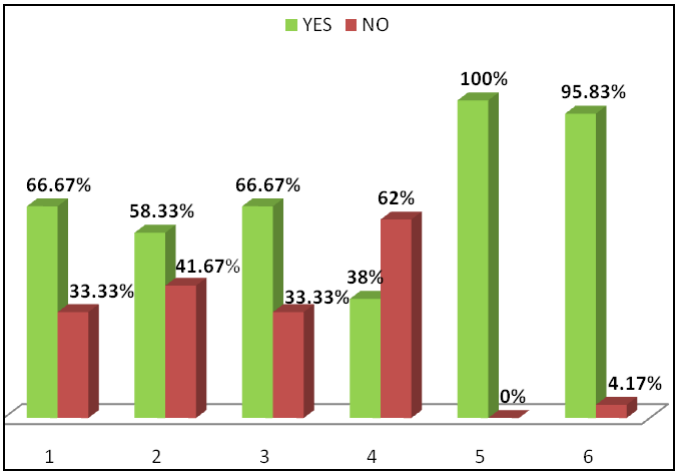
When we received the 24 comic strips developed by the students, we made an analysis of their work. First, we investigated if the comic strips were in accordance with our Grammar, if it was possible to do a syntactic analysis of the story. Then, we evaluated if the comic strips were in accordance with our Guidelines.

The results of our analysis if the students' works were in accordance with our Grammar are presented on Graph 4. We can see that 95.8% of the works were in accordance with our Grammar. Only one comic strip (4.2%) was not written in conformity with the Grammar, because there was a speech balloon connected in an object and the grammar does not have a rule that accepts balloons of communication with objects. Therefore, that comic strip would not be recognized during a syntactic analysis.



Graph 4 – Comic strips presented in accordance with the Grammar

In the next step, students should specify software requirements, creating comic strips according to our guideline and based on our Grammar defined for comic strips. The results of our analysis if the students' works were in conformity with our Guidelines are presented on Graph 5. The results for each of the six questions identified on Graph 5 are described below.



Graph 5 - Comic strips presented in accordance with the Guidelines

- Question #1: What is the theme of the Comic Strip?
  66.67% of the students defined the theme of the story correctly; it is the knowledge domain that will be implemented.

- Question #2: Who are the characters?
  58.33% of the students defined the characters of the story correctly; they are the actors that will interact with the software.

- Question #3: Where is the story?
  66.67% of the students defined the environment of the story correctly; it is the scenario where the actors are inserted.

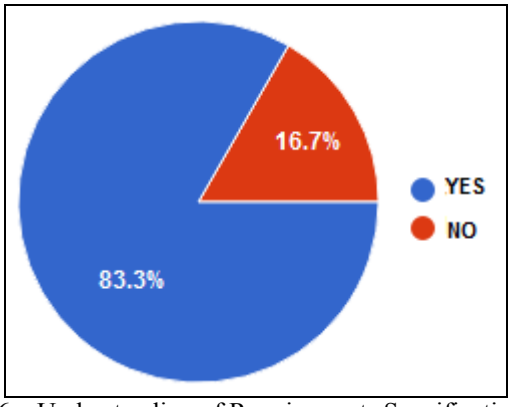
- Question #4: When is the story?
  38% of the students defined the time of the story correctly; it is when the story occurs.

- Question #5: Describe the process.
  100% of the students described the process correctly; it is the problem that needs to be solved.

- Question #6: Propose a solution for the problem.
  95.83% of the students proposed a solution for the problem correctly; they finished the story and the requirements specification.

After creating the comic strips, students answered two questions about the experience of specifying software requirements using comic strips. The results are presented in Graphs 6 and 7.

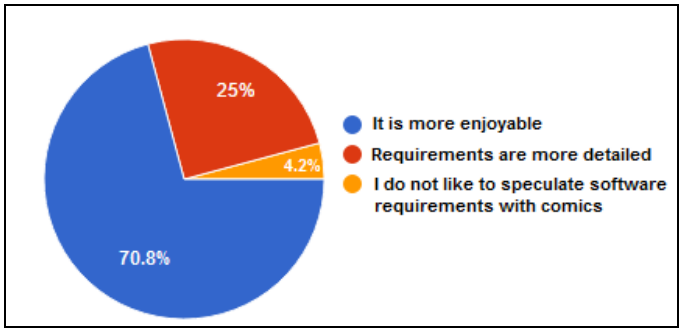
Graph 6 presents the answer to the first question: Did you understand better the concept of Requirements Specification using Comic Strips?



Graph 6 – Understanding of Requirements Specification using Comic Strips

It is worth mentioning that 83.3% of the students who participated in the experiment, which represents 20 students, stated that they understood better the specification of software requirements using comic strips. Only 16.7% of students, representing 4 students, stated that they did not understand it better.

The second question was what the students thought about the experience of specifying software requirements using comic strips, and the results are presented on Graph 7.



Graph 7 - Thoughts about the experience of specifying software requirements using comic strips

In Graph 7, we can see that 70.8% of students who participated in the experiment agreed that it was more enjoyable to specify software requirements using Comic Strips and 25% of students responded that this technique provides requirements that are more detailed. Only one student,

representing 4.2%, responded that he did not like to specify software requirements using comic strips.

VII. CONCLUSIONS

In this work, we proposed to introduce Comic Strips in the teaching of software requirements specification.

In our methodology, we defined a Grammar to describe the syntax of the language for comic strips, and we established a Guideline for its creation.

We conducted two experiments with Computer Science students, to verify how students would respond to these concepts. In the first experiment, we aimed to check if students can understand software requirements written in comic strips. In the second experiment, our main purpose was to find out if students could specify software requirements, by creating comic strips according to the guideline we had specified and based on the Grammar we had defined for comic strips.

Our first experiment was carried out with 77 students in the first year of Computer Science, so they did not have previous experience in software development. The results of this experiment were very positive; we concluded that students understood well the software requirements specification using Comic Strips.

The second experiment was carried out with 24 students in the fourth year of Computer Science. We explained them the Grammar for Comic Strips and the rules of Grammar. We also presented the guidelines for the creation of comic strips and explained them some examples of requirements, which were specified using comic strips. Then we asked students to write the specification of software requirements using Comic Strips, which should be in accordance with our Grammar and respecting the three acts, which we defined in the guidelines for creation of comic strips.

The results of this experiment were also very positive, as 95.8% of the students specified software requirements in accordance with our Grammar. This result leads us to state that it is possible to syntactically analyze the comic strips.

It is important to emphasize that the Guide for Comic Strips helped the students at the time of creation of their stories, to assist in the distribution of the details of the requirements.

Other significant results of this experiment were that most of the students reported that they understood better the specification of software requirements using comic strips and stated that it was more pleasant to specify software requirements using this technique.

The grammar and the guide we defined for the creation of comic strips demonstrated to be efficient in the teaching of software requirements specification and the results obtained with its use were very positive.

REFERENCES

[1] Sommerville, I. *Software Engineering*, 10th ed. Pearson, 2015.

[2] Pressman, R. *Software Engineering: A Practitioner's Approach*, 8th ed. McGraw-Hill Education, 2014.

[3] Motta, R. and Correia, W. "Design de histórias em quadrinhos digitais". *Sbgames.Org*, pp. 142–151, 2013.

[4] Beyer, D. and Hassan, A. E. "Animated Visualization of Software History using Evolution Storyboards". 13th Working Conference on Reverse Engineering, 2006.

[5] Ganesh, L. "The effect of comic strips as a supplementary material to teach computer networks". Proceedings of the IEEE 5th International Conference Technology Education T4E 2013, pp. 184–191, 2013.

[6] Bjarnason, E., Wnuk, K.and Regnell, B. "Requirements are Slipping Through the Gaps - A Case Study on Causes & Effects of Communication Gaps in Large-Scale Software Development". Proceedings of the 19th IEEE International Requirements Engineering Conference, pp. 37–46, 2011.

[7] D. Duarte, C. Farinha, M. Mira, and A. Rodrigues, "Collaborative Requirements Elicitation with Visualization Techniques," Proceedings of the IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2012.

[8] Moody, D. "The physics of notations: Toward a scientific basis for constructing visual notations in software engineering," *IEEE Transactions on Software Engineering*, vol. 35, no. 6, pp. 756–779, 2009.

[9] Zapata, S., Collazos, C. A., Giraldo, F. D. and Sevilla, G. "Distributed Elicitation of Software Requirements : an experimental case from Argentina and Colombia". In: 8th IEEE Computing Colombian Conference, *, pp. 1-7,* 2013.

[10] Boulila, N., Hoffmann, A. and Herrmann, A. "Using Storytelling to Record Requirements : Elements for an Effective Requirements Elicitation Approach". 4th International Workshop on Multimedia and Enjoyable Requirements Engineering, pp. 9–16, 2011.

[11] Takeda, M. M., et al. "A computational teaching approach through the use of a narrative technique and a comic strip". Proceedings of the IEEE 9th International Conference on Information Technology: New Generations, pp. 445–451, 2012.

[12] Babaian, C. S. "Comic Books , Graphic Novels , and a Novel Approach to Teaching Anatomy and Surgery". In: Proceedings of the IEEE Integrated STEM Education Conference, pp. 1–23, 2014.

[13] Barros, D. M. R., Begosso, L. R., Fabri, J. A., L'Erario, A. and Rosa, V. F. "The use of Comic Books in the Software requirements specification". In: Proceedings of the 11th Iberian Conference on Information Systems and Technologies, pp. 1–6, 2016.

[14] Louden, K. C. *Compiler Construction: Principles and Practice.Course Technology ed.,* 2004.