

# CLASSIFIQUI: a Game for Software Requirements Classification Learning

Francisco Elenilson Pessoa Monteiro<sup>1</sup>, Rainara Maia Carvalho<sup>1</sup>,  
Enyo José Tavares Gonçalves<sup>1</sup>

<sup>1</sup>Federal University of Ceará - Campus Quixadá  
Quixadá – CE – Brazil

**Abstract.** *Many problems in software development happen due to missing or insufficient handling of software requirements. The area that deals with how professionals must discover, document and validate requirements is Requirements Engineering (RE). RE is concerned with eliciting, analyzing, specifying, validating and managing user needs to be met by the software. In elicitation, professionals must discover and understand what are the customers and users needs. In this phase, a lot of information is passed on to the analysts and normally this information does not come in a succinct, complete and well-organized list. Serious games have been used to support several areas of computing knowledge, including RE. However, there is still a gap to support teaching requirements classification through play. Thus, this work presents the CLASSIFIQUI game, which was proposed to support the teaching of requirements classification. The game was evaluated through a study and proved to be suitable for teaching requirements classification.*

## 1. Introduction

Many problems in software development arise because of missing or poor attention to software requirements. In this way, Requirements Engineering (RE) becomes crucial for software development because it is responsible for discovering, analyzing and validating the requirements. [Wieggers and Beatty 2013].

RE comprises the entire process of discovering, analyzing and validating these requirements. RE has two main areas: requirements development, which involves elicitation, analysis, specification and validation; and requirements management, which encompasses the documentation maintenance steps and requirements management [Sommerville 2011, ISO 2018]. The main result obtained through RE is a set of documented and validated requirements [ISO 2018].

Thus, it is essential to support the teaching of RE in undergraduate computing courses, since professionals with a lack of knowledge about RE impact negatively software development [Navarro-Almanza et al. 2018].

The teaching of Software Engineering, and thus of RE, is evolving, and new approaches are emerging [SILVA J. 2018]. We can highlight the use of games for students to practice what they have learned theoretically, given that learning through games is more exciting and fun for students [Savi et al. 2011].

Classifying the information provided by the customer is an initial and essential step for creating requirements specifications [Sommerville 2011,

Wiegiers and Beatty 2013] and is the focus of this work. Requirements classification seems like a simple task, but this is not true. After elicitation, the requirements engineer must deal with a large amount of information obtained from customers and users. This information must all be classified, and many questions can arise in this process.

According to [Sakuda and Fortim 2018], digital games have gained significant prominence in several areas, such as education, training and simulation of professional situations. Therefore, developing a serious game based on the digital format becomes an interesting approach for teaching software requirements, especially for the Requirements Analysis phase, where the classification of information obtained in the elicitation into types of requirements is performed. This work proposes a game to help the teaching of requirements classification to support the training of requirements engineers. The game was developed as a mobile application and was evaluated in a study.

This paper is structured as follows: Section 2 presents concepts related to the classification of Requirements. Section 3 presents the related work. Section 4 presents the game Classifiqui. Game evaluation is presented in Section 5. Finally, the conclusion and future work are described in Section 6.

## 2. Background

The requirements describe what the system must do, its services, and the constraints on its operation. These requirements reflect customer needs for a system that serves a particular purpose. For example, in a sales control system, it is natural for customers to need to control a device, place an order, or find information [Sommerville 2011].

Several classifications of the types of requirements were proposed in the literature by different authors. Two of these widely used classifications are defined by [Sommerville 2011] and [Wiegiers and Beatty 2013].

According to [Sommerville 2011], software requirements are classified into two main categories: functional and non-functional requirements. Functional requirements (FR) are statements of services the system must provide, how the system must react to specific inputs, and how the system must behave in certain situations. Non-functional requirements (NFR) are restrictions on the services or functions offered by the system. They include *timing* restrictions, restrictions on the development process, and restrictions imposed by standards. Unlike individual system features or services, non-functional requirements often apply to the system as a whole.

We selected the classification proposed by [Wiegiers and Beatty 2013] because it has a greater variety of kinds of requirements, thus not overloading the classification of requirements in a type that might not be the best classification for it. Table 1 presents the types proposed by [Wiegiers and Beatty 2013] as well as their definitions.

Kind	Meaning
Business Requirement	A high-level organizational business objective.
Business Rule	A policy, guideline, standard or regulation that defines or restricts some aspect of the business.
Constraint	A restriction imposed on the options available to the developer for designing and building a product.
External Interface Requirement	Description of a connection between the software and the user, another system or hardware device.
<i>Feature</i>	One or more logic-related system features that add value to the user.
Functional Requirement	A description of behaviors that a system will exhibit under specific conditions.
Nonfunctional Requirement	A description of a property or characteristic that a system must exhibit or a constraint must be applied.
Quality Attribute	A type of non-functional requirement that describes a service or performance characteristic of a product.
System Requirement	A top-level requirement for a product that contains multiple subsystems.
User Requirement	A goal or task that specific classes of users should be able to perform.

**Table 1. Kinds of Requirements and its Meanings.**

### 3. Related Work

In this section, we introduce four games that support RE learning.

UbiRE [Lima et al. 2012] is a game developed as a fun way to teach Requirements Engineering (RE) in the context of ubiquitous systems. The game takes place in a 3D virtual environment represented by a house, where it is divided into 4 (four) phases, these being represented by the rooms of the house. The player should assemble the house system, where he must create the ubiquitous connections between the equipment in each room. These connections represent the game's difficulty levels. At each level, there is a minimum number of connections necessary to move on to the next.

[García et al. 2019] developed the game *Biyubi* to help teach software requirements elicitation techniques in undergraduate engineering courses. The game works as a real-world simulation where students interact with other game characters to learn elicitation techniques. The game occurs within a library environment, where requirements must be elicited to build the library's system, for which requirements elicitation activities are carried out.

*Requengin* [García et al. 2020] is a game based on 3D simulation where students can practice and learn the fundamentals of requirements engineering established by the ISO/IEC 29148. Using the real-world exploration perspective, students need to explore three simulated scenarios, they are the player's home, university classroom, and academic library.

ERQuiz [Sarinho et al. 2019] aims to teach the foundations of Requirements Engineering. This game has three different modes. The first one is called *Play by Time*,

where the player has 5 minutes to answer as many questions as he can about RE. Second, *Best of 5*, where the player will test players' knowledge of RE through five questions. Third, *Can't Miss*, where the player will have to answer a question, if he gets it right, the game continues giving him other questions, if he fails the game ends. The game has a ranking where, according to the player's score in each match, the game will analyze if the player's score can enter the ranking of the best players.

The games presented in this section aim to support the learning of Requirements, like Classifiqui. However, they do not support the requirements' classification activity.

#### **4. Classifiqui**

The name classifiqui was proposed as the combination of two words: **classification** and **requirements**. The game's target audience is undergraduate computing students, which offers Software Engineering and/or Software Requirements in its curriculum.

A set of requirements elicited by a system analyst arrives at the company, and it is necessary to classify them, for which requirements analysts and programmers are allocated to classify them. The classification process, if performed incorrectly, will require rework in the future. Therefore, players must keep the project within the established budget, in the context of the game, budget is related to the system's production time, that is, it has a deadline to be developed and players must maintain the project within that period.

Before using this game, players must acquire knowledge about the types of requirements defined by [Wieggers and Beatty 2013]. One of the players creates a room in the game to carry out the game so that the other participants can enter that room. Classifiqui is a card game (genre) with three kinds of cards: requirements, kind of requirements and bonuses. The game is an Android app whose code is available at Classifiqui<sup>1</sup>.

##### **4.1. Game Mechanics**

According to [Brathwaite and Schreiber 2009], mechanics are rules that act on players, game state and game views and describe all the ways to change the game state. The Players, Cards and Rules are presented below.

A maximum of seven players and a minimum of three players participate in each game match. Each player has at least one chance to use the Senior Requirements Analysts card once. According to [Calazans et al. 2017], the requirements' analyst is the bridge between customer needs and the development team. Thus, the analyst card has the power to reveal to the players the type of requirement for which it was requested. The player also have the chance to use the junior requirements analysts card twice. The junior requirements analysts have the same definition of senior requirements analysts, but with less experience. The player has cards for each type of requirement, being able to use them as many times as necessary.

The player who will start the classification is defined through the order of access to the room of the match in dispute, which means the first one to enter the room is the first to classify and so on.

One important thing about the game is that all project's requirements should be classified, since every information from requirements elicitation must be classified.

---

<sup>1</sup><https://github.com/eleMonteiro/classifiqui>

Therefore, the number of game matches corresponds to the number of requirements that needed to be classified.

The game displays a card from the *deck* of challenges for each player in his turn to play. The player can perform the classification of the requirement card after carefully reading it. The order in which these cards are displayed follows the order of access to the game room. If the player correctly classifies the requirement, he adds 1 (one) point to his score and has the right to classify a second requirement. If it is correctly classified, the player draws a card from the bonus *deck*, after the draw from the bonus card, the turn passes to the next player. Otherwise, the requirement passes to the next player.

The game displays to the players on their turn the type chosen by the previous player in case he misses the classification. Each player can use the requirement type cards as many times as they want to classify the requirements. Each player can use the junior requirements analysts card twice. It has the function of showing the user three possible types for the pulled requirement. Each player can use the Senior Requirement Analyst card only once, where it is possible to request to see what type of requirement. If players use any help cards, they do not get the right to classify another requirement in the round and therefore cannot draw a card from the bonus *deck*. Players only leave the game when all project requirements have been sorted. The game ends when all requirements are sorted.

## 4.2. Game Platform

This digital card game was developed using *React Native*, and the data storage uses *Firestore*. Thus, the game uses digital technologies and cards as the main interaction tool. In addition, the game was developed only in the mobile version. The developed game code can be found at Classifiqui<sup>1</sup>. The app can be downloaded from Classifiqui<sup>2</sup>.

## 4.3. Interaction with the Game

The game has four *decks*, three of which are based on those developed by [Beppe et al. 2018] and an additional one named help.

### 4.3.1. Challenges Deck

The challenges cards, which are part of the Classifiqui game, will be presented. The challenges represent software requirements to be classified. Therefore, these cards have the requirements of three systems selected in the industry: i) a solution for providing care and adopting pets; ii) an application to support and manage the document printing; and iii) an mobile app to support a guided visit to companies. The requirements of the first one were created for a Requirements discipline at the Federal University of Ceará. The two last solutions were developed by the Group of Computer Network, Software Engineering and Systems [Andrade et al. 2017]. Additionally, we considered the system for a coffee shop whose requirements were presented by [Wieggers and Beatty 2013]. Figure 1 shows requirements cards. The left represents a card of the printing support system, and the right is a requirement of the pet adoption system.

---

<sup>1</sup><https://github.com/eleMonteiro/classifiqui>

<sup>2</sup><https://www.dropbox.com/s/q44sdtqirriyhk/app-release.apk?dl=0>

**Figure 1. Card of the printing support system (lef) and pet adoption (right)**

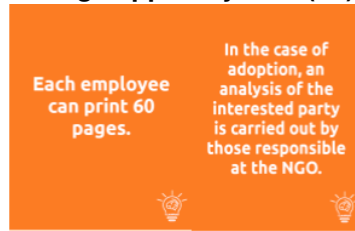
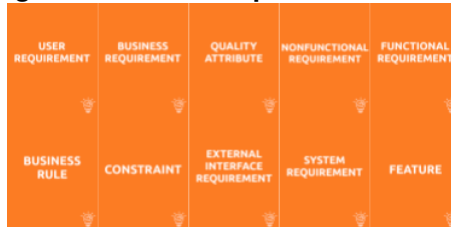


Figure 2 presents the kinds of requirements cards. These cards are used to classify the requirements and has the following options: (i) Business Requirement, (ii) Business Rule, (iii) Constraint, (iv) External Interface Requirement, (v) Feature. (vi) Functional Requirement, (vii) Non-functional Requirement, (viii) Quality Attribute, (ix) System Requirement and (x) User Requirement.

**Figure 2. Kind of Requirements Cards.**



### 4.3.2. Help Deck

The help cards are presented in this section. These cards help the players during the game. They can be used to get clues about the requirements to be classified. Figure 3 shows the two help cards used in the game. The Junior Requirements Analyst card reduces to three possibilities of choosing the type from the initial 10, thus making it easier for the user to classify the requirement correctly, the Senior Requirements Analyst card classifies the requirement for the player. All players initially gain the same amount of help cards in the game.

**Figure 3. Help Cards.**

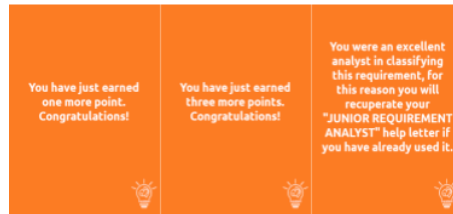


### 4.3.3. Bonus Deck

The bonus cards are used for players who correctly classify the requirements in each round of the game. Figure 4 shows two types of bonuses obtained by players after correct

classifications.

**Figure 4. Bonus Cards.**



#### **4.4. Score Rules**

The game does not contain negative scores, so the game's scoring system was carried out through the sum of points obtained by the participants for the correct classifications and the bonuses obtained. This score is displayed to players in *ranking* format, as this encourages other players to try to improve their ranking.

Players earn points for each correctly ranked requirement, and when the player correctly ranks two requirements in the same round, they get the chance to draw a card from the bonus deck to increase their score. Bonus cards contain scores from 1 (one) to 3 (three) points, however to make the game more dynamic, some bonus cards contain redemption for the junior requirements analyst and senior requirements analyst cards.

At the end of each match, the application calculates the score. It is based on the score obtained through the correct classification of the requirements plus the score of the bonus cards. Whoever gets the best result will be the champion player.

### **5. Classifiqui Evaluation**

This section presents the game evaluation. [Goal] The purpose of this study is to identify positive aspects and improvements to be implemented regarding the Classifiqui game. [Population and Sample] While experienced researchers can benefit from using the game, we believe that people with little experience in requirements are more prone to errors and difficulties to classify requirements. Therefore, we defined as the target population of this study undergraduate students in the area of computer science who were or had already taken the disciplines of Software Engineering and/or Software Requirements. We chose the students from the Software Engineering and Information Systems courses at Federal University of Ceará - Campus Quixadá as a sample. From 15 students invited, 7 accepted and participated. [Preparation] We created a questionnaire in *Google Forms*<sup>1</sup>, following the principles of [Kitchenham and Pfleeger 2002]. The questionnaire has 13 (thirteen) questions to assess the participants' experience while using the game, as well as questions related to the content provided in the game.

We presented the main questions of the evaluation form following: **Q1** - Do you have knowledge about any kind of existing requirements? Answer options: Yes/No. If you answered YES, describe your knowledge. **Q2** - It was easy to play Classifiqui. Answer options: Likert scale with 5 options, **Q3** - Classifiqui increased my knowledge about requirements classification. Answer options: Likert scale with 5 options. **Q4** - When

<sup>1</sup>The list of questions is available at <https://forms.gle/N2XFfB4dPUC7uHZW8>

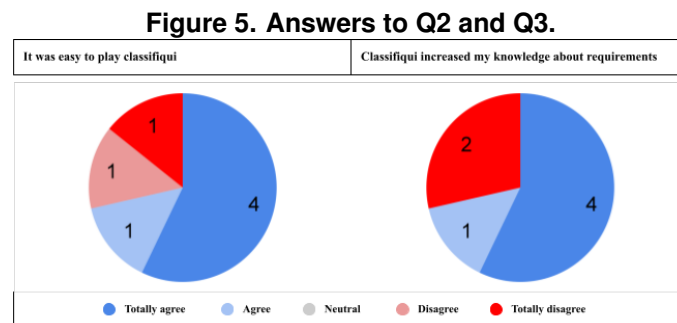
using the Classifiqui game, were there any problems/difficulties or disadvantages? Examples: crash, unreadable texts, among others. Answer options: Yes/No. If you answered YES, describe your opinion. **Q5** - What are your suggestions to improve the Classifiqui game?

[Execution] Data collection took place on March 30, 2021. The evaluation was planned to take place in the participants' environment, that is, on their own cell phones. Thus, the participants were instructed to install the game on their cell phones and use it right away. After using the game, participants were asked to answer a questionnaire.

The following section presents the main results.

### 5.1. Evaluation Results

Figure 5 shows results regarding questions 2 (game usability) and 3 (impact on knowledge about requirements). Most participants (5/7) agreed Classifiqui is easy to play and improved their knowledge about requirements due to the usage of the game.



The following are some of the improvements most cited by the participants: (i) It would be interesting to implement logout; (ii) It would be nice announcing on the screen that it is another player's turn, because at that moment it looks like the game has crashed; (iii) It could inform who got the question right in the round; (iv) It could show at the beginning a list of the rooms already created; (v) It could limit the sorting time; (vi) It could give feedback on who is the current player; (vii) It could place some informational buttons or instructions on some screens, especially on the bonus one; and (ix) Use more icons for a better user experience.

We also asked participants to answer their overall satisfaction level with the game. They characterized their experience on a scale from 1 to 5 as 1 being totally agree and 5 being totally disagreed. Most participants chose option 1 of the scale, indicating their experience as satisfactory.

Regarding the content, we asked the participants about their understanding of the requirements laid out in the game, if they were easy to understand and free of problems such as ambiguity. Most participants classified the requirements as clear and free of problems that made it difficult to understand and use the game.

We asked participants to describe the main difficulties encountered when using the game, below are some points described by them: (i) Not having understood how the game works, initially. But after understanding, it was easy. (ii) Knowing who the player was, there was a bit of confusion to know what to do on some screens.



Finally, these collected data provided evidence that the game can have a positive effect on teaching requirements classification.

## 5.2. Threats to Validity

This section discusses threats to the validity of the survey instrument. For [Kitchenham and Pfleeger 2002], four aspects must be considered: Face Validity, Content Validity, Conclusion Validity and Construction Validity.

[Face validity] It can be understood as a superficial review of the instrument items by inexperienced people. The questionnaire was initially presented to two people who had little knowledge of the subject being evaluated, with the aim of reviewing the structure, design and objectivity of the questionnaire questions. Participants achieved a high percentage of correct answers to the questions. There were some suggestions for improvement that were met. [Content validity] This is a subjective assessment of how suitable the instrument seems to a group of people with knowledge on the subject. We carried out a pilot test with three students. The goal was to test the participant's understanding of the research and ensure that it includes all the necessary information for the actual evaluation. The feedback received in this pilot test showed some improvement points, which were (i) the issue of viewing the cards, (ii) how the next player is chosen, (iii) how the cards would be better displayed on the screen.

[Validity of conclusion] It concerns the ability to reach the correct objective on the data collected, using statistical tests, and how reliable the measurements and these data are. Due to the low number of participants in this study, we could not make statistical inferences about the data, which posed a threat to the validity of this study. [Construction validity] It is the observation of how the research instrument behaves when it is in use. To try to mitigate threats of this type, we sought to elaborate objective questions, which were validated in the pilot test that was carried out. In the results of this study, we saw that the data converged towards a high percentage of success when using the game to answer the questions of our research instrument.

## 6. Conclusion e Future Work

This work focused on the development of a digital card game to support the teaching of requirements regarding their types, such as Functional Requirement, Non-Functional, Business Rule, Feature, among others.

The Classifiqui game, the main contribution of this work, allows the player to put into practice what was previously learned theoretically about the different types of requirements. The game was evaluated by users in an empirical study. We received positive feedback and some improvements to be applied.

Currently, Classifiqui only has requirements of three types and is available for Android systems. As future work, it is possible to develop requirements for all types described in this work and develop it for IOS systems. Also, we will work to improve the game's design based on the results identified in the evaluation.

## References

(2018). **ISO 29148**: ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering.

- Andrade, R. M. C., Carvalho, R. M., de Araújo, I. L., Oliveira, K. M., and Maia, M. E. F. (2017). What changes from ubiquitous computing to internet of things in interaction evaluation? In Streitz, N. and Markopoulos, P., editors, *Distributed, Ambient and Pervasive Interactions*, pages 3–21, Cham. Springer International Publishing.
- Beppe, T. A., de Araújo, Í. L., Aragão, B. S., de Sousa Santos, I., Ximenes, D., and Andrade, R. M. C. (2018). Greatest: a card game to motivate the software testing learning. In *XXXII Brazilian Symposium on Software Engineering*, pages 298–307.
- Brathwaite, B. and Schreiber (2009). *Challenges for Game Designers*. Course Technology.
- Calazans, A., Paldês, R., Braosi, E., Rezende, K., and Pereira, N. (2017). O perfil do analista de requisitos de software: uma comparação entre a academia e o mercado de trabalho brasileiro. In *Workshop em Engenharia de Requisitos*.
- García, I., Pacheco, C., León, A., and Calvo-Manzano, J. A. (2019). Experiences of using a game for improving learning in software requirements elicitation. *Computer Applications in Engineering Education*, 27(1):249–265.
- García, I., Pacheco, C., León, A., and Calvo-Manzano, J. A. (2020). A serious game for teaching the fundamentals of iso/iec/ieee 29148 systems and software engineering–lifecycle processes–requirements engineering at undergraduate level [S. l.]: *Computer Standards & Interfaces*, 67:103–377.
- Kitchenham, B. A. and Pfleeger, S. L. (2002). Principles of survey research part 2: Designing a survey. *SIGSOFT Softw. Eng. Notes*, 27(1):18–20.
- Lima, T., Campos, B., Santos, R., and Werner, C. (2012). Ubire: A game for teaching requirements in the context of ubiquitous systems. In *38a Conferencia Latinoamericana En Informatica*, pages 1–10.
- Navarro-Almanza, R., Juarez-Ramirez, R., and Licea, G. (2018). Towards supporting software engineering using deep learning: A case of software requirements classification. *5th International Conference in Software Engineering Research and Innovation*.
- Sakuda, L. and Fortim, I. (Brasília: Ministério da Cultura, 2018). *Ii censo da indústria brasileira de jogos digitais*.
- Sarinho, V. T., Gomes, V. O., and Sarinho, W. (2019). Erquiz: A multiplayer multiplatform instant messaging game for the competitive assessment of requirements engineering knowledge. *Proceedings of the XVIII SBGames*, pages 591–594.
- Savi, R., von Wangenheim, C. G., and Borgatto, A. F. (2011). A model for the evaluation of educational games for teaching software engineering. In *2011 25th Brazilian Symposium on Software Engineering*, pages 194–203.
- SILVA J., C. E. d. (2018). **Investigação do ensino de engenharia de requisitos na perspectiva da academia e da indústria:** um enfoque em documentação de requisitos. Dissertação de Mestrado em Sistemas e Computação.
- Sommerville, I. (2011). *Engenharia de Software*, volume 9. Pearson, São Paulo.
- Wieggers, K. and Beatty, J. (2013). *Software Requirements, 3rd Edition*. Microsoft Press.