Geolympus - Cloud Platform for Supporting Location-Based Applications: A Pervasive Game Experience

Juan Luis Berenguel Forte[1(✉)], Daniel Pérez Gázquez[1], Jeferson Arango-López[1,2], Francisco Luis Gutiérrez Vela[1], and Fernando Moreira[3,4]

[1] E.T.S. of Computer and Telecommunication Engineering, University of Granada, c/Periodista Daniel Saucedo Aranda, s/n, 18071 Granada, Spain
{juanlubf, danielpg, fgutierr}@ugr.es, jeferson@correo.ugr.es

[2] Faculty of Electronics Engineering and Telecommunications – FIET, University of Cauca, Cl. 5 # 4 – 70, Popayán, Cauca, Colombia

[3] REMIT, IJP, Universidade Portucalense, Rua Dr. António Bernardino Almeida, 541-619, 4200-072 Porto, Portugal
fmoreira@upt.pt

[4] IEETA, Univ Aveiro, Aveiro, Portugal

Abstract. The development of game experiences based on the player's location is becoming increasingly popular. This games are used to increase players' skills and knowledge about a particular topic in different contexts like education, health and tourism. When pervasive or narrative components are added, it becomes evident the need for a tool to manage the information in an appropriate and dynamic way. When we talk about pervasiveness, in addition to the elements of space and time, there is an additional component related to social interaction, which can be achieved through the exchange of information between various games to provide a better player experience. For this reason, in this article we present a platform enabled for the creation and edition of game experiences based on the player's location. Which has the ability to exchange information between projects, including narrative in the different games. A developed experience evidences the relevance of this platform in the education context.

Keywords: Geolocation · Pervasive game · Web platform · Cloud · Microservices

1 Introduction

Games based on players' location involve a component that is not found in console games: the locomotion of players through the outside world and, possibly, the realization of physical activities [1], taking the players into a highly pervasive environment. Geolocation games is one of the game genres in which pervasiveness is applied

Á. Rocha et al. (Eds.): WorldCIST'19 2019, AISC 932, pp. 256–266, 2019.
https://doi.org/10.1007/978-3-030-16187-3_25

14. Singer, L., Schneider, K.: It was a bit of a race: gamification of version control. In: 2nd International Workshop on Games and Software Engineering, Zurich, Switzerland, pp. 5–8. IEEE (2012)
15. La Toza, T., Towne, W., Van Der Hoek, A., Herbsleb, J.: Crowd development. In: 2013 6th International Workshop on the Cooperative and Human Aspects of Software Engineering (CHASE), San Francisco, USA, pp. 85–88. IEEE (2013)
16. Prause, C., Nonnen, J., Vinkovits, M.: A field experiment on gamification of code quality in agile development. In: Psychology of Programming Interest Group Annual Conference (2012)
17. Colomo-Palacios, R., Tovar-Caro, E., García-Crespo, A., Gómez-Berbís, J.: Identifying technical competences of IT Professionals: the case of software engineers Professional Advancements and Management Trends in the IT Sector. Int. J. Hum. Cap. Inf. Technol. Prof. (IJHCITP) **1**(1), 31–43 (2010)
18. Oprescu, F., Jones, C., Katsikitis, M.: I PLAY AT WORK—ten principles for transforming work processes through gamification. Front. Psychol. **5**(14) (2014)
19. Shepard, L.: The role of assessment in a learning culture. Educ. Res. **29**(7), 4–14 (2000)
20. Manrique-Losada, B., Gasca, G., Gómez, M.: Assessment proposal of teaching and learning strategies in software process improvement. Revista Facultad de Ingeniería **77**, 105–114 (2015)
21. Krutchen, P.: The 4 + 1 view model of architecture. IEEE Softw. **12**(6), 42–50 (1995)

As future work we identify: (1) apply and evaluate the proposed method for design gamified pedagogic instruments for others subjects of software engineering like software quality, design patterns, and effort estimation; (2) develop a digital tool to support the pedagogic instrument design and as a repository for the instruments, and; (3) develop a tool to guide the professor in the definition of rubrics for the gamified activities created.

Acknowledgement. This work has been partly founded by the Universidad de Medellín, the Universidad Politécnica de Madrid, and Procesix Inc., under the joint project "PSP/TSP Teaching and Learning Methodology as an initiative for quality and productivity levels improvement of software development teams".

References

1. Sommerville, I.: Software Engineering, 10th edn. Pearson, Boston (2015)
2. Akili, G.: Games and simulations: a new approach in education, 1st edn. IGI Global (2014)
3. Nah, F., Zeng, Q., Telaprolu, V., Ayyappa, A., Eschenbrenner, B: Gamification of education: a review of literature. In: International Conference of HCI in Business, pp. 401–409. Springer (2014)
4. Uyugari, F., Intriago, M., Jacome, E: Gamification proposal for a software engineering risk management course. Springer, Switzerland (2015)
5. Towhidnejad, M., Hilburn, T.: Integrating the Personal Software Process (PSP) across the undergraduate curriculum. In: 27th Conference Frontiers in Education, Pittsburgh, USA, pp. 162–168 (1997)
6. Chacón, P.: El Juego Didáctico como estrategia de enseñanza y aprendizaje ¿Cómo crearlo en el aula? Nueva aula abierta **16**, 32–40 (2008)
7. González, A.: Diseño de juegos y creatividad: un estudio en el aula universitaria. Opción **31** (4) (2015)
8. Hazeyama, A.: Collaborative software engineering learning environment associating artifacts management with communication support. In: 3rd International Conference on Advanced Applied Informatics, Kitakyushu, Japan, pp. 592–596. IEEE (2014)
9. Pieper, J.: Learning software engineering processes through playing games: suggestions for next generation of simulations and digital learning games. In: Proceedings of Workshop on Games and Software Engineering, Zurich, Switzerland, pp. 1–4. ACM (2012)
10. Dubois, D., Tamburrelli, G.: Understanding gamification mechanisms for software development. In: Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, Saint Petersburg, Russia. ACM (2013)
11. Barata, G., Gama, S., Jorge, J., Goncalves, D.: Engaging engineering students with gamification. In: Proceedings of the First International Conference on Gameful Design, Research, and Applications, Ontario, Canada, pp. 10–17. ACM (2013)
12. Berkling, K., Thomas, C.: Gamification of a Software Engineering course and a detailed analysis of the factors that lead to it's failure. In: 2013 International Conference on Interactive Collaborative Learning (ICL), Kazan, Russia, pp. 525–530. IEEE (2013)
13. De Melo, A., Hinz, M., Scheibel, G., Berkenbrock, C., Gasparini, I., Baldo, F.: Version control system gamification: a proposal to encourage the engagement of developers to collaborate in software projects. In: International Conference on Social Computing and Social Media, Crete, Greece, pp. 550–558. Springer (2014)

software design concept and the intention to apply PSP in the area of SE in which they work. Such an indicator is important, since one of the problems with PSP teaching is the lack of real implementation in companies.

Table 6. Frequency of competences assessment

Participant competence		Frequency (%)
What are the main concepts learned during the activity?	• Concept and importance of PSP • Importance of software design • Detailed lists • Prioritization lists • Disciplined Practices	50 20 10 10 10
What is the most important activity that the systems engineer must do in the software design based on PSP?	• Detailed and prioritized list • Acquisition of skills • Importance of design • Checklist generation • Design activities planning	65 15 10 5 5

6 Conclusions

This paper presents a proposal of a pedagogic instrument design method based on experiential work as learning engine and gamification as a transversal motivating element. This method includes five components that cover the whole process of pedagogic instrument design, from preparation and design, going through the pilotage of application of the instrument to test his mechanics and goals, until reach the schedule and assessment of the instrument once it has been applied to the target audience.

The study case consists in designing a pedagogic instrument with the proposed method, for teaching software design under PSP. The results obtained from the survey support the conclusion that the didactic technique applied to the instrument is appropriate. In fact, the 100% of participants say they would apply the concept of PSP in the area of SE which they currently perform, showing that if other alternatives are sought to present the subjects, it is possible to obtain more receptivity of the public. In relation with the level of learning achieved by the participants, they recognize the importance of PSP (50%), to a lesser extent the importance of software design (10%) and the use and prioritization of checklists (10%). This means that at the level of competencies of the participants, we achieved to emphasize the importance of software design and the use of PSP framework in this area of knowledge.

Regarding to the competencies-based assessment, in this paper we present a rubric template as an instrument for the facilitator in the application of the game CAR DESIGN PSP. Such template includes the Bloom's taxonomy learning domains, activity categories corresponding with the main tasks expected in students and a set of rubrics for orientate the assessment of application level for each activity. Such template is an important contribution for gamified activities, since it is not only important the design process, but also the evaluation process of the competencies to be promoted.

Table 4. (*continued*)

Level	Description	
6: Evaluation	E	Recognize all the missing features required in the design. All the judgments founded by the car inspector related to the external criteria are accepted to improve the design
	G	Recognize some of the missing features required in the design. At least the judgments founded by the car inspector related to the external criteria are accepted to improve and complete the design
	F	The missing features required in the design are not recognized for the team The judgments founded by the car inspector to improve the design, are not accepted

5 Study Case Discussion

The assessment was applied to 20 people (professionals from a software company at Medellín (Colombia) and systems engineering students at the Universidad of Medellín) during the 2nd semester of 2015. In this section we present the results obtained from the application of the instrument for the Sect. 2 'instruments and teaching techniques' and Sect. 3 'levels of student satisfaction', from component E. The survey contains 20 variables; the most representative ones according to each evaluated feature are presented in Table 5. The assessment scale is Very Poor (VP), Poor (P), Fair (F), Good (G) and Very Good (VG). According to the results the positive ratings are related to the features of the didactic technique used—gamified instrument; in most cases the highest percentage corresponds to good and very good insights on the evaluated feature.

Table 5. Frequency of teaching techniques

Teaching techniques (Features)	Frequency (%)				
	V P	P	F	G	VG
Instructions presentation	0	0	10	10	90
Time for development of the activity	0	0	5	30	65
Clarity of instructions	0	0	0	20	80
Teaching materials quality	0	0	0	0	100

Regarding the assessment of the student satisfaction level during the gamified instrument application we found that features as the enjoyment level and creative thinking stimulation are above 85% with ratings as very good or excellent; while the level of closeness to reality is below 60% with ratings as excellent and very good.

Finally, as a way for measuring the student competencies, the variables assessed are shown in Table 6. About the concepts learned, PSP gets a rate of 50%, this is one of the core concepts of the instrument. However, students identify a lesser extent (10%) the importance of software design and disciplined practices for software development. Another aspect to highlight is that 100% of participants manifest understand the

The application level is defined as a scale of achieved learning, in terms of: Excellent (10 points), Good (8 points), or Fair (6 points). The application level for each learning level from the template, is in Table 4:

Table 4. Application level for each learning level of assessment template

Level		Description
1: Knowledge	E	All-important major and minor elements for design software are identified and appropriately prioritized. All relevant information is obtained and exact information sources are consulted. Design recommendations are well supported by the information
	G	All major elements for design software are identified but one or two minor ones are missing or priorities are not recognized. Sufficient information is obtained and most sources are valid. Design recommendations are mostly supported by the information
	F	Many major elements for design software are not identified. Insufficient information is obtained and/or sources lack validity. Design recommendations are not supported by information collected
2: Comprehension	E	Among the alternatives analyzed they have been considered: prioritized criteria, improvement cycles, and identification of bugs. Three or more alternatives are considered. Each alternative is appropriately and correctly analyzed for technical feasibility
	G	Among the alternatives analyzed they have been considered: improvement cycles and identification of bugs. At least three alternatives are considered. Appropriate analyses are selected but analyses include some minor procedural errors
	F	Among the alternatives analyzed they have been considered only prioritized criteria Only one alternative is considered. Inappropriate analyses are selected and/or major procedural and conceptual errors are made
3–4: Application/Analysis	E	Car design has consistent aspects by categories. All conditions and rules are considered correctly
	G	Some mistakes are evident in the definition of criteria by categories. Some conditions and rules are considered correctly
	F	The criteria by all categories are not defined. The conditions and rules are not considered
5: Synthesis	E	Responsibilities have been delegated fairly, and each member contributes in a valuable way to the design. The work of all team members demonstrates the implementation of a design and implementation strategy of the solution The car design was correctly finished, at time and with resources constrains established
	G	Some minor inequities in the delegation of responsibilities. Some members contribute more heavily than others but all members meet their responsibilities The work of the half team members demonstrates the implementation of a design and implementation strategy of the solution. The car design was at least 80% finished, at time and with resources constrains established
	F	Major inequities in delegation of responsibilities. Team has obvious freeloaders who fail to meet their responsibilities or members who dominate and prevent others from contributing. The teamwork no demonstrate applying a design and implementation strategy of the solution. The car design was at least 50% finished

(*continued*)

D – Scheduling the Final Instrument. The pedagogic instrument was adjusted based on improvements and suggestions incorporated in the pilot. In this session was possible to evaluate the game using the assessment proposal mentioned in the component E.

E – Assessment. We developed an assessment of the proposed method, considering the three sections of the reference assessment proposal: (1) competencies of participants; (2) instruments and teaching techniques; and (3) levels of student satisfaction regarding the teaching process. Such assessment was performed using the survey as a support tool for evaluation. We design a detailed instrument for assessing the competences of participants, as a complement to the Sect. 1 from the reference assessment proposal.

Such a detailed assessment instrument is based on a rubric template. The template was designed to support the competence-based assessment, due to other assessment sections were more developed—instruments and teaching techniques, and satisfaction levels. The template is shown in Table 3, and the application level detail are as follows:

Table 3. Rubric template for supporting the competence-based assessment

Learning levels			Rubric features		Application level
#	Level	Activity category	Related abilities/skills	Weight	Excellent (E) Good (G) Fair (F)
1	**Knowledge**	**Become acquainted**	• Recognize the role of design software in the software engineering process **[10%]**	**30%**	(E)/(G)/(F)
		Prioritizing information	• Identify the needed information about: Errors in design quality; PSP for design software; and 4 + 1 View Architecture **[20%]**		(E)/(G)/(F)
2	**Comprehension**	**Understanding and inferring**	• Understand the meaning of using PSP for software design **[5%]** • Interpret the basic aspect for car design **[5%]**	**15%**	(E)/(G)/(F)
		Exploring alternatives	• Plan and analyze alternatives of a solution to given problem **[5%]**		(E)/(G)/(F)
3-4	**Application/Analysis**	**Structure analysis**	• Interpret elements, principles, and structure for car design **[10%]**	**15%**	(E)/(G)/(F)
		Matching	• Identify internal relationships and components from car **[5%]**		(E)/(G)/(F)
5	**Synthesis**	**Fulfillment of duties**	• Assign, compliance, and fulfill of responsibilities of design **[5%]**	**30%**	(E)/(G)/(F)
		Implementation strategy	• Apply a strategy for design and implement the solution **[10%]**		(E)/(G)/(F)
		Completion	• Complete the design given time and resource constraints **[15%]**		(E)/(G)/(F)
6	**Evaluation**	**Reflection**	• Recognize the missing features required in the design **[5%]**	**10%**	(E)/(G)/(F)
		Judgment compliance	• Accept judgments relating to external criteria to improve and complete the design **[5%]**		(E)/(G)/(F)

- **Game materials:** Materials for game are: (1) checklist for the Car inspector to register the mechanical, design, and technology elements for the car, (2) colored paper, (3) puzzle, (4) template of the 4 + 1 architectural view model [21], and (5) chronometer.
- **Game roles:** The roles for CAR DESIGN PSP are shown in Table 2.
- **Game steps**: Next, the step-by-step for developing this game are the following:

1. Participants conform teams of five persons where a participant assumes the role of car inspector and the other four will be experts from the automotive sector.
2. It has four quadrants corresponding to categories to generate checklists for car design: design, technical specifications, safety, technology, and comfort.
3. The inspector tells the team when start to fill the quadrant (distributed by experts) identifying items and assigning them a priority for generating checklists.
4. The inspector assesses the expert performance in team, and if he/she has a satisfactory performance and delivers puzzle pieces to assemble a car.
5. The winning team is the one that make the most detailed specifications of each category of car design and assemble the puzzle in the shortest time possible.

C – Pilotage. We developed a pilot, conforming a team of five students of different levels of a software engineering academic program of the Universidad de Medellín (Colombia). In this pilot, we establish improvement actions and recommendations incorporated in the final game version.

Table 2. Roles of CAR DESIGN PSP

Role	Responsibilities
Car inspector	1. Check if the team has all the resources necessary to execute the game activities 2. Check if the 4 + 1 architectural view model is consistent with the car elements 3. Prepare report of the game activities 4. Measure time for each game activity 5. Reward the team with the puzzle pieces, when is necessary 6. Check the template of the 4 + 1 architectural view model in each phase
Expert	1. Work in team 2. Participate in all game activities 3. Create a list of elements necessary for the car design 4. Assemble the puzzle
Facilitator	1. Help to teams in achieving the game objectives 2. Support every team to do their best in the different activities 3. Promote collaboration and try to achieve synergy 4. Indicate mission, challenge, instructions and rules of the game

Table 1. Components of PID

Component				
A—Preparation	B—Design	C—Pilotage	D—Scheduling	E—Assessment
Define goals to achieve with the instrument, based on the analysis of: (1) competencies to be developed, (2) learning goals, (3) profile of the population, (4) particular interests and age of the population	Outline elements of gamification to include in the instrument. *i.e.* be reward, status, achievement, and competition. Also the mechanics of instrument is defined, *i.e.* the rules and processes	Test instrument with a different audience to the target population (friends, family, and colleagues). Then adjusts of the game mechanics (rules, materials, or time for each activity) are executed	Provide spaces, resources, and materials required for the application of the final instrument	Identify participants perception about the instrument and facilitator performance, by using the assessment proposal presented in [20]

4 Pilot Study Case

According to the PID method, we design a pedagogic instrument for teaching *software design* under PSP, called *CAR DESIGN PSP*. Such instrument uses an analogy between 'car design' and 'software design', considering that before construct a car or a software product is necessary elaborate the structural models of a final product. Such instrument can be applied in a session of 1.5 to 2 h. Also, in CAR DESIGN PSP we use a checklist to evaluate the product quality, corresponding the PSP premise: "measure before improve". The results of the PID application are as follows:

A – Preparation. The development results of the steps included in this component are:

- **Learning Goals:** Identifying software bugs in early stages caused by software design models of poor quality; emphasizing the importance of software design in the software engineering process, and encouraging the implementation of good practices described in the PSP framework for software design.
- **Competencies to be developed by the student:** Understand the importance of using PSP in software design; improve how to design software implementing PSP good practices; and implement software design practices focused on PSP.

B – Gamified Instrument Design. The instrument comprises:

- **Rules:** Comprise the rules set required to achieve the game goal: (1) Teamwork, not interfere with the work of other teams; (2) Respect time assigned to create the car elements list; (3) Respect the turn of expert participants when they describe car elements; (4) Accept and follow the instructions given by the Car Inspector; (5) Car Inspector must not exhibit the checklist to experts; (6) Car Inspector must not indicate car design elements to participants; and (7) Team must use all puzzle pieces.

3 PID Method

In this section, we present the PID (Pedagogic Instrument Design) method to design pedagogic instruments for SE teaching. The goal of PID is guide the creation of pedagogic instruments, based on: (1) experience as key factor to learning, and (2) gamification in an educational environment, as a strategy to stimulate the classroom work and increase the participants motivation. The method is designed as a sequential path where a trainer can obtain a pedagogic instrument to guide the teaching of a specific subject. The elements of the method are presented in Fig. 1 and are as follows:

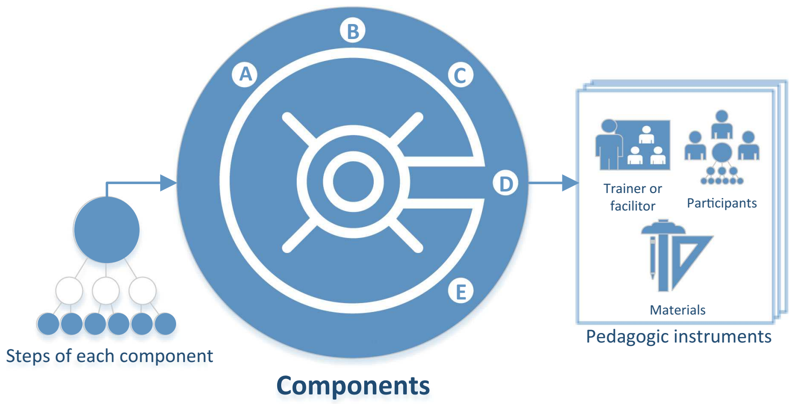

Fig. 1. Elements of the PID method

- **Components.** Comprises the following components: (A) Preparation, (B) Design, (C) Pilotage, (D) Scheduling, and (E) Assessment.
- **Steps.** Each component contains a step-by-step sequence to obtain a pedagogic instrument designed under the gamification strategy.
- **Pedagogic instrument.** Generated artifact because of the steps of each component. Each generated instrument comprises: *Participants* (groups of students or professionals in training; *Trainer or facilitator* (professors, trainers, or facilitators of an activity conducted with an instrument), *Materials* (set of necessary resources for the application of the gamified pedagogic instrument.

The method was designed considering the following pedagogical principles: (1) Planning [17]; (2) Environment gamification [18]; and (3) Experimentation [19]. The description of each component is shown in Table 1.

applied in software development phases; (2) integrate such approach to the development process, and (3) evaluate the proposed activities. The preliminary results show an important increase in the quality of software artifacts produced by the students in the gamified approach regarding to the artifacts produced without it. In the same sense, Barata *et al*. [11] highlight the motivational power for incorporating game principles in non-games processes, like education. They compare a software engineering gamified course with his previous non-gamified version through different performance measures.

Berkling and Thomas [12] propose gamifying a software engineering course to promote independent learning in students. Such proposal is based on that games are good motivators and involve the participants in an environment where they want to reach their goals and obtain recognition among their classmates.

In the context of configuration management, de Melo *et al.* [13] apply gamification to recognize software developer collaboration and commitment. They use a tool for extracting information from a control version system where developers execute code commits. Such a tool shows a ranking of more active developers using metrics. The goal is determine if gamification usage encourages collaboration and commitment in team members. Moreover, Singer and Schneider [14] use gamification of version control for encouraging students to make frequently commits with a social software.

Concerning to software development process, La Toza *et al.* [15] propose crowd development as an option for organizing software development process into micro-tasks. Micro-tasks are short (few minutes), modular, and self-descriptive. Such proposal could motivate students to join and contribute to an open source real project developing new skills, sharing knowledge, and participating in collaborative work.

Finally, Prause *et al.* [16] describe an experiment of gamification of code quality in agile development. The experiment consists on form teams of ten experimented students in programming and one instructor doing agile development. The teams work for four weeks in a share space. The indicators used for measure software internal quality are: (1) understandability of source code, (2) reputation score of each developer because his uploaded files and, (3) number of bugs injected and removed for each developer.

In summary, in the previous approaches exist an interest for incorporating active learning strategies in the teaching and learning of software engineering process, looking for the development of technical and social competencies in the students. In this context, gamification is an option to promote motivation and engagement. However, its necessary formalize the gamification teaching experiences with elements adapted to new generations, spaces adopted for reflecting future improvement and guarantying participants wellbeing, among others. Additionally, all such approaches incorporate gamification for software engineering teaching but is also necessary include in these experiences the definition and assessment of competencies expected in students as a guide for teachers. Accordingly, the goal of this paper is present the PID method to use gamification as a teaching method in a software engineering course including the assessment of competencies to develop in students.

Related approaches using gamification in software engineering teaching are focused on developing competencies in software engineering in several topics [4]. They indicate that the gamification compared to other teaching techniques, help to understand and learn concepts. Recent gamification research has suggested that game element-mapping to learning content may indeed facilitate processes good to software engineering [3, 4].

Both, academic and professional qualifications of software industry should promote and reflect the required competencies. For this reason, it is important to explore new ways of teaching and learning that support the development of management competencies, continuous improvement, and high performance, as are described in Personal Software Process (PSP) framework. These competencies should be aligned with the needs and requirements of the software industry about the professionals training [5].

In this paper, we propose the PID (pedagogic instrument design) method. The created instrument following PID comprises a teaching strategy, techniques and materials required for teaching a particular subject. The aim of the PID is facilitating the trainer work from a pedagogical point of view. PID is based on gamification as a strategy to design playful scenarios in the classroom, in order to encourage direct interaction of the participants (student or professional in training). The validation of PID is based on a pilot pedagogic instrument for teaching software design. The overall purpose is the concepts appropriation and promotion the creativity development by participants; the specific purposes are oriented to strengthening teamwork and cooperation [6, 7].

This paper is organized as follows. Section 2 describes background and related work. Section 3 introduces PID, the method to design pedagogic instruments. Section 4 describes the experimental study cased used to evaluate PID. Section 5 then combines the results and discussion about the pilot study. Finally, Sect. 6 presents our conclusions.

2 Background and Related Work

In this section, some approaches of application of gamification in software engineering teaching, and the reported benefits in terms of motivation and students performance are summarized.

Hazeyama [8] presents a learning environment for collaborative software development associating artifacts management with communication support. Such management is supported by a tool providing functions like file-based artefacts management, planning and progress report management, meeting minute management and announcement from teaching staff the student progress in the proposed activities. Meanwhile, Pieper [9] proposes the usage of simulation and digital learning games as a teaching strategy for a software engineering learning environment due to their potential to extend the learning experiences beyond lectures and class projects.

Dubois and Tamburrelli [10] promote the gamification usage for engaging, training, and monitoring students involved in the software products development from inception to maintenance phases. They propose a strategy based in three complementary activities: (1) analyze gamification approaches and identify the most appropriated to be

Using Gamification in Software Engineering Teaching: Study Case for Software Design

Gloria P. Gasca-Hurtado, Maria C. Gómez-Álvarez(✉), and Bell Manrique-Losada

Universidad de Medellín, Cra.87 No. 30-65, Medellín, Colombia
{gpgasca,mcgomez,bmanrique}@udem.edu.co

Abstract. Software engineering discipline needs promoting and responding to the demands of the software industry and their challenges, centered on the diversity and short delivery times in the projects. Looking to align with such demand, software engineering teaching has evolved incorporating newel strategies for increasing student motivation in the learning process. Gamification is one of them strategies centered on games principles, as the interactivity, ludic, and enhance engagement. This strategy, compared to other teaching techniques, improve the processes of conceptual understanding and learning. In this paper we propose a method, based on gamification, to design pedagogic instruments, comprising a strategy, techniques, and materials for teaching a specific subject. The goal of method is facilitate the gameful activities design in the classroom and then increase of motivation, cooperation and teamwork in participants, in the learning process of conceptual and practical subjects. The method was validated with the topic of software design in a course of a software engineering of the Universidad de Medellín (Colombia). In this pilot we establish improvement actions and recommendations incorporated in the final game version.

Keywords: Active learning strategies · Engineering teaching and learning · Pedagogic instrument · PSP

1 Introduction

Today software industry demands the development of high quality complex software. Such demand requires software engineers with excellent competencies, able to choose the suitable tools and processes to accomplish dynamic requirements. Actually, key challenges of software engineers are increasing diversity and the need to shortened delivery times while guaranteeing trustworthy quality [1]. Therefore, software engineering teaching should consider such diversity. Accordingly, research in this field incorporates active learning strategies, like gamification, for increasing student motivation in the learning process. Gamification is being used in the learning context because it's a strategy focused on the interactive and highly engaging character of games. This strategy motivates learners to "take responsibility for their own learning, which leads to intrinsic motivation" [2], and "enhance engagement and improve learning outcomes by means of integrated learning environments" [3].

Á. Rocha et al. (Eds.): WorldCIST'19 2019, AISC 932, pp. 244–255, 2019.
https://doi.org/10.1007/978-3-030-16187-3_24

3. Vuorikari, R., Punie, Y., Carretero, S., Van den Brande, L.: DigComp 2.0: The Digital Competence Framework for Citizens: Update Phase 1: The Conceptual Reference Model. Publications Office of the European Union, Luxembourg (2016)
4. Carretero, S., Vuorikari, R., Punie, Y.: DigComp 2.1: The Digital Competence Framework for Citizens With Eight Proficiency Levels And Examples of Use. Publications Office of the European Union, Luxembourg (2017)
5. Siiman, L., Mäeots, M., Pedaste, M., Simons, R-J., Leijen, Ä., Rannikmäe, M., Võsu, K., Timm, M.: An instrument for measuring students' perceived digital competence according to the DIGCOMP framework. In: Zaphiris, P., Ioannou, A. (eds.) Proceedings of the International Conference on Learning and Collaboration Technologies LCT 2016, pp. 233–244. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39483-1_2
6. Pérez-Escoda, A., Castro-Zubizarreta, A., Fandos-Igado, M.: Digital skills in the Z generation: key questions for a curricular introduction in primary school. Comunicar **23**(49), 71–79 (2016). https://doi.org/10.3916/C49-2016-07
7. Siddiq, F., Hatlevik, O.E., Olsen, E., Throndsen, R.V., Scherer, R.: Taking a future perspective by learning from the past—a systematic review of assessment instruments that aim to measure primary and secondary school students' ICT literacy. Educ. Res. Rev. **19**, 58–84 (2016). https://doi.org/10.1016/j.edurev.2016.05.002
8. Hatlevik, O.E., Ottestad, G., Throndsen, I.: Predictors of digital competence in 7th grade: a multilevel analysis. J. Comput. Assist. Learn. **31**(3), 220–231 (2015). https://doi.org/10.1111/jcal.12065
9. Van Deursen, A.J.A.M., van Diepen, S.: Information and strategic Internet skills of secondary students: a performance test. Comput. Educ. **63**, 218–226 (2013). https://doi.org/10.1016/j.compedu.2012.12.007
10. Calvani, A., Fini, A., Ranieri, M., Picci, P.: Are young generations in secondary school digitally competent? A study on Italian teenagers. Comput. Educ. **58**(2), 797–807 (2012). https://doi.org/10.1016/j.compedu.2011.10.004
11. Claro, M., Preiss, D., San Martín, E., Jara, I., Hinostroza, J.E., Valenzuela, S., Cortes, F., Nussbaum, M.: Assessment of 21st century ICT skills in Chile: test design and results from high school level students. Comput. Educ. **59**(3), 1042–1053 (2012). https://doi.org/10.1016/j.compedu.2012.04.004
12. Livingstone, S.: EU kids online. In: Hobbs, R. (ed.) The International Encyclopedia of Media Literacy. Wiley-Blackwell, Oxford (2017)
13. Hatlevik, O.E., Guðmundsdóttir, G.B., Loi, M.: Digital diversity among upper secondary students: a multilevel analysis of the relationship between cultural capital, self-efficacy, strategic use of information and digital competence. Comput. Educ. **81**, 345–353 (2015). https://doi.org/10.1016/j.compedu.2014.10.019
14. Ramos, J.L., Carvalho, J.M.: Tablets no ensino e na aprendizagem. A sala de aula Gulbenkian: entender o presente, preparar o futuro. Fundação Calouste Gulbenkian, Lisboa (2017)
15. Aesaert, K., van Nijlen, D., Vanderlinde, R., van Braak, J.: Direct measures of digital information processing and communication skills in primary education: using item response theory for the development and validation of an ICT competence scale. Comput. Educ. **76**, 168–181 (2014). https://doi.org/10.1016/j.compedu.2014.03.013
16. Karaseva, A.: Pedagogy of connection: Teacher's experiences of promoting students' digital literacy. In: Erstad, O., et al. (eds.) Learning Across Contexts in the Knowledge Society, pp. 225–242. Sense Publishers, Rotterdam (2017)

the tablet had an impact on the development of their digital competence, both in the areas of "Information and data literacy" and "Communication and collaboration". On the contrary, students labelled as non-frequent users disagree as to that impact.

Although results may seem, somehow predictable, they partially contradict the ones found by Pérez-Escoda, Zubizarreta and Fandos-Igado [6], who concluded that students do not develop digital competence by simply using media and technology. However, they are in line with the assumption made by the same authors, and others [9, 10], regarding the need to address digital competence in schools in a way that digital technology is used with a view to increase acquisition in a gradual and progressive manner. In this respect, teachers also need to be digitally competent in order to infuse digital technologies in their practices and facilitate their learners' digital competence. Nevertheless, their ability to do so is often questioned and the need to get more insights on how they can achieve it is claimed [13]. Results are also in line with those found by previous studies [14], conducted within a similar context, that suggest students who use the tablet more often learn more and reach higher academic achievements.

At this point, some limitations of the study need to be considered as well as ways to overcome them in the future. First, the fact that it used an indirect measurement instrument. Several authors [7, 15] point at the weaknesses of such measurements, arguing that students' self-reports capture individual beliefs, self-confidence and self-efficacy, which are not always a good or faithful representation of their actual performance level. Considering the inclusion of direct measures, such as performance-based tests, should be taken into account in future studies. Second, the fact that other variables did not come into play during the analysis, which could have helped to deepen the findings achieved. These could include: (i) the type of activities students performed in school and at home and knowing whether teachers prioritized digital competence during lessons and school activities or not. Research points at the role teachers play in promoting students' digital competences [16]; (ii) the socio-economic background of the students. Different studies refer family background, cultural capital or academic achievements and aspirations as predictors of digital competence [8, 10, 11, 13]. These are directions that future research in the field should approach. Despite the limitations, this article contributes to the existing literature in several respects, such as the unveiling of students' perceptions regarding the impact of tablet use on the development of their digital competence, which is an under-researched topic, or the opportunity to further research on emerging aspects that may contribute for positive impacts.

Acknowledgements. This work is supported by the Portuguese Foundation for Science and Technology (FCT) under grant number SFRH/BPD/100367/2014 and project UID/CED/00194/2019.

References

1. Ferrari, A.: Digital Competence in Practice: An Analysis of Frameworks. Publications Office of the European Union, Luxembourg (2012)
2. Ferrari, A.: DIGCOMP: A Framework for Developing and Understanding Digital Competence in Europe. Publications Office of the European Union, Luxembourg (2013)