

Gamification Design for a Software Testing I Undergraduate Course Using Level-Up Framework

Kláudio Henrique Mascarenhas Medeiros^{1,2}, Roberta de Souza Coelho¹,
Charles Andryê Galvão Madeira³

¹Programa de Pós-Graduação em Sistemas e Computação (PPgSC) - Universidade Federal do Rio Grande do Norte (UFRN) - Natal, RN - Brasil

²Centro Universitário Unifacisa - Campina Grande, PB - Brasil

³Instituto Metrópole Digital - Universidade Federal do Rio Grande do Norte (UFRN) - Natal, RN - Brasil

klaudiohenrique.medeiros@gmail.com, roberta@dimap.ufrn.br

charles@imd.ufrn.br

Abstract. Introduction: *Software Testing is an essential part of Software Engineering; however, it faces educational challenges such as insufficient practical training and student demotivation. Objective:* *This study proposes a gamified design for an undergraduate Software Testing I course. Methodology or Steps:* *This design is guided by the Level-Up framework for creating gamified educational experiences, which integrates Design Thinking steps with the Octalysis gamification framework. Expected Results:* *This gamified approach is expected to contribute to Software Testing education by improving students' performance and intrinsic motivation, promoting a more engaging, collaborative, and meaningful learning experience.*

Keywords *Gamification, Education, Software testing, Software testing education.*

1. Introduction

In modern society, people rely on software systems to solve problems, meet demands, and interact with one another. Therefore, it is essential that such systems function properly. In this context, Software Testing aims to identify defects and ensure that software behaves as intended [Myers et al. 2004, Sommerville 2011].

Despite its relevance, Software Testing remains insufficiently addressed in Information Technology (IT) curricula [Astigarraga et al. 2010, de Jesus et al. 2020], as dedicated courses are often absent or offered only in intermediate or final semesters [Scatalon et al. 2019, Silvis-Cividjian 2021]. Additionally, testing activities frequently suffer from low student motivation [Blanco et al. 2023]. As a result, these issues contribute to the formation of unmotivated professionals [Straubinger e Fraser 2023] and to a shortage of qualified practitioners [Silvis-Cividjian 2021].

To address student demotivation, game-based strategies have been adopted in both academia and industry. One such strategy is gamification, which consists of applying game elements to non-game contexts [Deterding et al. 2011] to promote positive behaviors, such as participation, collaboration, and creativity. In Software Testing,

gamification has been used to foster engagement in activities such as unit, functional, and mutation testing [de Jesus et al. 2018, Rojas e Fraser 2016, Fulcini et al. 2023].

This study proposes a gamified approach for an undergraduate Software Testing I course offered at the Federal University of Rio Grande do Norte (UFRN). The approach is based on the Level-Up framework for designing gamified educational experiences [de Souza Brito e Madeira 2024], which combines Design Thinking principles [Brown 2020] with gamification strategies derived from the Octalysis framework [Chou 2019]. The main contribution of this work is a gamified design based on stakeholders' requirements (e.g., students and instructors) and guided by a consolidated gamification framework. This design focuses on game elements beyond points and leaderboards, such as narrative and collaboration.

The remainder of this paper is organized as follows. Section 2 reviews related work, Section 3 describes the design of the gamified approach based on the Level-Up stages, and Section 4 concludes the paper and discusses future work.

2. Related Work

Blanco et al. [Blanco et al. 2023] developed a gamified solution for an undergraduate Software Verification and Validation course. The gamification design includes elements such as points, leaderboards, and narrative, and was developed based on input from gamification designers and experts [Trinidad et al. 2021] and did not investigate participants' intrinsic motivation. Clarke et al. [Clarke et al. 2021] conducted a study that combined different learning and engagement strategies in Software Testing classes, including gamification. The gamification design used in this study relies on game elements like points, teams and leaderboards. Jesus et al. [de Jesus et al. 2020] used a gamified approach with elements like points, badges, avatars and leaderboards, to teach functional testing concepts. In this study, students were not previously consulted about their needs and expectations, as expected by frameworks such as design thinking and did not consider the relatedness factor with motivation questionnaires.

This work contrasts with the aforementioned works, mainly because it uses a framework for designing gamified experiences for education that focuses on understanding stakeholders and their expectations before proposing a solution (Level-Up), as well as implementing narrative, teams and methods for evaluating the relatedness factor in a gamified dynamic [Ryan et al. 1983], which contributes to a better understanding of the student's relationship with partners in a competitive-cooperative activity, investigating how teamwork can contribute with software testing tasks.

3. Gamification Design

The gamified approach presented in this work was designed using the Level-Up framework [de Souza Brito e Madeira 2024], which supports the development of gamified educational experiences, comprising five steps, which are described in this study.

Regarding **Step 1 (Scenario research)**, we conducted a semistructured interview with the course instructor, as well as a presentation on gamification for students, during which we discussed and collected information about students' backgrounds and expectations regarding a gamified experience. This step was conducted in a previous

study [Nascimento et al. 2024]. Therefore, the gamification design is guided by the following needs: (i) opportunities to engage in projects and practical activities; (ii) access to support from instructors or peers; and (iii) exposure to real-world scenarios. The design incorporates elements such as peer collaboration, clear objectives, a sense of challenge, narrative and thematic components, progressive difficulty, and appropriate rewards.

In **Step 2 (Defining gamification objectives)**, the primary goal is to foster motivation and encourage student engagement in Software Testing activities. In the context of the Software Testing I course, the focus includes topics such as structural testing criteria, and defect-based testing. Additionally, student activity-related aspects are considered, such as adherence of test cases to the session log standard, use of functional testing criteria, criticality of identified defects, consistency in task completion, compliance with deadlines, participation in discussions, and collaboration.

For **Step 3 (Selecting game elements)**, the gamification design was guided by motivational drivers of the Octalysis framework, known as core drives. The proposed design integrates different core drives and game elements to support student motivation and engagement. The selection of game elements was mainly guided by needs identified in previous steps and Octalysis's game techniques that incorporates the framework's core drives [Chou 2019]. In a traditional learning approach (Figure 1a), students are guided by obligations such as course syllabi and exams, have limited autonomy, and are subject to continuous pressure, such as test performance and task deadlines. In contrast, our gamified approach is guided by the following core drives and game elements (Figure 1b):

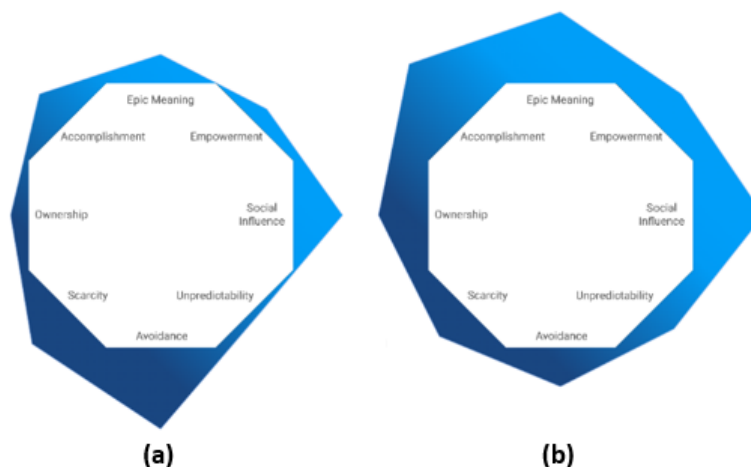


Figure 1. Core drive comparison between traditional (a) and gamified (b).

- **Core Drive #1 (Epic Meaning and Calling)** refers to the perception that an individual is doing something greater than themselves or has been chosen for a meaningful mission. **Selected game elements:** (i) **Narrative:** The experience follows a storyline in which students must defend a planet or organization from enemy attacks (e.g., hackers); (ii) **Elitism:** Students are organized into teams with shared goals, requiring collective effort and fostering a sense of responsibility.
- **Core Drive #2 (Development and Accomplishment)** relates to progress, skill and overcoming challenges. **Selected game elements:** (i) **Status Points:** A scoring system based on mission completion; (ii) **Progress Bar:** Students can

visualize progress through a bar that increases as missions are completed; (iii)

Leaderboards: Students are ranked based on accumulated points during the gamified dynamic. Only the top 50% of students are displayed at any given time. Team leaderboards rank groups based on points earned by their members.

- **Core Drive #3 (Empowerment of Creativity and Feedback)** relates to engagement in creative activities and receiving feedback on created outputs. **Selected game elements:** (i) **Milestone Unlock:** New challenges and content are unlocked as the course progresses and students advance in the dynamic.
- **Core Drive #4 (Ownership and Possession)** refers to the feeling of ownership over something and the desire to improve it through action. **Selected game elements:** (i) **Collection Sets:** Certain elements, such as badges or missions, are organized into collections following specific criteria (e.g., 90% attendance in Units I–III; completion of 5, 10, or 20 tasks; participation in 1, 5, or 10 forum discussions); (ii) **Protector Quest:** Students assume the role of protectors defending entities (e.g., a planet or organization) from threats such as bugs and system failures, while developing Software Testing skills and concepts.
- **Core Drive #5 (Social Influence and Relatedness)** relates to social interactions such as mentoring, social acceptance, teamwork, and competition. **Selected game elements:** (i) **Group Quests:** Team-based challenges requiring collaboration to achieve shared objectives. These may involve tasks where each team member has a specific role, or collective goals based on scoring (e.g., reaching 2000 points per team); (ii) **Conformity Anchors:** The class average performance is displayed for selected tasks or overall scores to encourage improvement.

This gamified approach minimizes the use of urgency-based or pressure-driven strategies. Therefore, Core Drives #6 (Scarcity and Impatience), #7 (Unpredictability and Curiosity), and #8 (Loss and Avoidance) are not directly addressed in this design.

The **Step 4 (Prototyping the gamified approach)** involves the implementation of gamification's design. At this stage, each student has an individual profile displaying information such as points, leaderboards, and missions. In the prototype's first version, profiles are implemented using Google Spreadsheets. Each document is accessible only to the student, instructors, and teaching assistants (TAs). Only instructors and TAs are allowed to update game-related information. More details about the spreadsheets are available at: <https://github.com/klaudiohmmedeiros/gamificationdesigntest20261>. An example of a mission set is shown in Figure 2:

- **Classroom attendance missions:** Goals that involve a percentage of attendance in classes (e.g. 90% of attendance). They can be divided into assessment units;
- **Grading missions:** Goals that involve achieving a minimum score on tests (e.g. 85 or more on a test with maximum grade of 100, A or B grades);
- **Exposition missions:** Goals that involve the student presenting something or participating in discussions (e.g. participation in forums, in-class presentations);
- **Team missions:** Goals that involve tasks or scoring thresholds for the student's team (e.g. obtaining a target score, sub-missions for each team member);
- **Specific missions:** Goals that involve performing some specific action involving Software Testing in a given task (e.g. code coverage, boundary value analysis).

Finally, **Step 5 (Evaluating results)** consists of evaluating the solution based on results obtained from prototype's application, investigating if the objectives have been

Elitism		Checklist	Status points	Description	Missions	Reward
Name (or nickname)	the_placeholder					
Student's Team	Bug Busters					
Points	100	☒	Guaranteed Presence I	The student had only one unexcused absence in Unit I		60
Progress		☐	Guaranteed Presence II	The student had only one unexcused absence in Unit II		60
Average Points	195		Grading missions			
Level	★	☐	Master of Testing I	The student got a grade of 8.6 or higher in Unit I		100
Next level	50	☐	Master of Testing II	The student got a grade of 8.6 or higher in Unit II		100
			Exposition missions			
		☒	The Arguer I	The student commented in 1 online discussion		40
			The Arguer II	The student commented in 5 online discussions		60
Student leaderboard	Student	☐	Center of the Spotlight	The student conducted an expository presentation of some activity		40
300	SKILL		Team missions			
290	ironmind	☐	Quality Alliance Level 1	The student's team scored over 1000 points		40
280	Unknown_student7	☐	Quality Alliance Level 2	The student's team scored over 1800 points		60
		☐		The student's team scored over 2400 points		100
Team leaderboard	Team					
580	Master of Testers	☐	Master of Exercises I	The student completed 3 activities satisfactorily and submitted them on time.		40
350	No Bug Shall Pass	☐	Master of Exercises II	The student completed 8 activities satisfactorily and submitted them on time.		80
240	Bug Busters		Specific missions			
		☐	Learning with Other	The student completed the activity "Bugs Research Activity".		20
				The student found 6+ bugs in the "Palindrome" exercise.		20

Figure 2. Example of a mission set for a Software Testing course

achieved and the influence of each game element. For this step, students will complete testing exercises during a semester, and their activity and performance will be analyzed. This step also includes the application of interviews and questionnaires¹, such as the SRQ-L (Self-Regulation Questionnaire - Learning) and IMI (Intrinsic Motivation Inventory).

4. Conclusions and Future Work

This work presents a gamified approach for a Software Testing I course. This approach integrates Design Thinking and Octalysis to design a motivational learning experience aimed at increasing student engagement in software testing activities. The design emphasizes elements that foster drives like epic meaning, development and relatedness.

This gamification approach has some challenges and limitations. Since the gamification is managed through spreadsheets, instructors and TAs must create and maintain these spreadsheets, define missions, and continuously update the records to ensure that the information remains accurate and up to date. From a design perspective, instructors with little or no experience in gamification may face difficulties in selecting game elements or implementing the gamified dynamics. This process could benefit from a tool that supports instructors in the design and management of gamification.

As future work, we plan to apply this gamified design in a Software Testing I course over a semester to observe student behavior, collect their perceptions, and investigate its impact on the teaching and learning process. Additionally, future studies may compare this approach with traditional teaching methods, as well as refine its design and implementation through a dedicated tool to support the gamified experience.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Generative artificial intelligence (ChatGPT, GPT-5) was used for grammatical corrections and to improve the overall quality and readability of the manuscript.

¹The SRQ-L is available at <https://selfdeterminationtheory.org/self-regulation-questionnaires/>, while the IMI is available at <https://selfdeterminationtheory.org/intrinsic-motivation-inventory/>

References

- Astigarraga, T., Dow, E. M., Lara, C., Prewitt, R., e Ward, M. R. (2010). The emerging role of software testing in curricula. In *2010 IEEE Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*, pages 1–26. IEEE.
- Blanco, R., Trinidad, M., Suárez-Cabal, M. J., Calderón, A., Ruiz, M., e Tuyá, J. (2023). Can gamification help in software testing education? findings from an empirical study. *Journal of Systems and Software*, 200:111647.
- Brown, T. (2020). *Design Thinking: uma metodologia poderosa para decretar o fim das velhas ideias*. Alta Books.
- Chou, Y.-k. (2019). *Actionable gamification: Beyond points, badges, and leaderboards*. Packt Publishing Ltd.
- Clarke, P. J., Davis, D. L., Buckley, I. A., Potvin, G., Thirunarayanan, M., e Jones, E. L. (2021). Combining learning and engagement strategies in a software testing learning environment. *ACM Transactions on Computing Education (TOCE)*, 22(2):1–25.
- de Jesus, G. M., Ferrari, F. C., de Paula Porto, D., e Fabbri, S. C. P. F. (2018). Gamification in software testing: A characterization study. In *Proceedings of the III Brazilian Symposium on Systematic and Automated Software Testing*, pages 39–48.
- de Jesus, G. M., Ferrari, F. C., Paschoal, L. N., de Souza, S. d. R. S., de Paula Porto, D., e Durelli, V. H. S. (2020). Is it worth using gamification on software testing education? an extended experience report in the context of undergraduate students. *Journal of Software Engineering Research and Development*, 8:6–1.
- de Souza Brito, A. L. e Madeira, C. A. G. (2024). Level-up: um processo gamificado para a sala de aula. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 694–707. SBC.
- Deterding, S., Dixon, D., Khaled, R., e Nacke, L. (2011). From game design elements to gamefulness: defining "gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15.
- Fulcini, T., Coppola, R., Ardito, L., e Torchiano, M. (2023). A review on tools, mechanics, benefits, and challenges of gamified software testing. *ACM Computing Surveys*, 55(14s):1–37.
- Myers, G. J., Badgett, T., Thomas, T. M., e Sandler, C. (2004). *The art of software testing*, volume 2. Wiley Online Library.
- Nascimento, E. H., Coelho, R., Madeira, C., Medeiros, K., Silva, L., e Neto, C. O. (2024). Definição e avaliação de uma abordagem gamificada para o ensino de teste de software. In *Simpósio Brasileiro de Engenharia de Software (SBES)*, pages 455–465. SBC.
- Rojas, J. M. e Fraser, G. (2016). Code defenders: a mutation testing game. In *2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 162–167. IEEE.
- Ryan, R. M., Mims, V., e Koestner, R. (1983). Relation of reward contingency and interpersonal context to intrinsic motivation: A review and test using cognitive evaluation theory. *Journal of Personality and Social Psychology*, 45(4):736–750.

- Scatalon, L. P., Carver, J. C., Garcia, R. E., e Barbosa, E. F. (2019). Software testing in introductory programming courses: A systematic mapping study. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 421–427.
- Silvis-Cividjian, N. (2021). Awesome bug manifesto: Teaching an engaging and inspiring course on software testing (position paper). In *2021 Third International Workshop on Software Engineering Education for the Next Generation (SEENG)*, pages 16–20. IEEE.
- Sommerville, I. (2011). Software engineering 9th edition. *ISBN-10*, 137035152:18.
- Straubinger, P. e Fraser, G. (2023). A survey on what developers think about testing. In *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*, pages 80–90. IEEE.
- Trinidad, M., Calderón, A., e Ruiz, M. (2021). Gorace: A multi-context and narrative-based gamification suite to overcome gamification technological challenges. *Ieee Access*, 9:65882–65905.