Session 2: *IT Education and Practice*

Professional paper
UDC: 371.3:004.42

Game-Based Learning of Software Testing

Stefan Šošić[1], Olga Ristić[1*], Marjan Milošević[1]
[1] University of Kragujevac, Faculty of Technical Sciences Čačak, Serbia
[*] olga.ristic@ftn.kg.ac.rs

Abstract: *The importance of software testing is often overlooked leading to consequences that can affect a company's reputation. Software testing is representing a critical aspect of the software development process but considered often as tedious process when using traditional education methods. It is a challenge how to teach software testing in an effective way that can successfully motivate students. In this paper, we have carried out analysis of traditional software testing education method and software testing education with support of games. We pointed out advantages and disadvantages of using games in education of software testing.*

Keywords: *software; testing; games; traditional education;*

1. INTRODUCTION

Software testing is a critical phase in software product development [1]. It is also one of the skills that a good programmer possesses, but software testing is often considered repetitious action. It is an open challenge how to teach software testing in an effective way that can successfully motivate students. The traditional way of learning requires students to resort to standard methods of learning from provided literature and materials [2]. Every human action causes the production of the hormone dopamine, which serves as a significant part of the brain's reward system. Learning from the provided literature is a tedious process for students.

In contrast, playing games causes the student to produce very high levels of dopamine. As a result, students remain focused and spend significantly more time playing games than learning. A new generation of students has grown up in the digital world, using computers and smart devices every day, playing games and browsing the Internet. Students spend a lot of time playing games and use their problem-solving skills at a high level of motivation. In fact, the problem that the game gives them keeps their attention and concentration. Philippe Bootz's diagram clarifies the unique quality of digital texts which applies also to texts in games [3].

Given the difference between traditional and learning supported with games, more and more games that support learning have emerged in recent years. Those games have succeeded in improving student motivation and increased students desire to learn and make learning more enjoyable. Combining games and learning software testing provides easier education of software testing techniques and methods [4]. With minimal effort, students gather knowledge in the field of software testing that they can use in other subjects on faculty, in the implementation of their own projects or at work.

2. LEARNING SOFTWARE TESTING THROUGH GAMES

Educational games for software testing can be online web browser games, desktop offline games, multiplayer or single player. The target of testing games is to support material for subjects containing content related to software testing. In this way, students can improve their software testing skills even outside the classroom. Therefore, software testing games can be considered a additional material for learning of software testing. Software testing is a large area and encompasses a large number of methods that can be used in software testing [5].

Game used for software testing education should cover at least few of software testing methods shown in Table 1 [6, 7]. If software is not systematically and thoroughly tested, then software quality suffers. If students want to become experts in the field of software testing, they must know all the methods of software testing. The more software testing methods the game covers, students' knowledge will be more comprehensive.

Table 1. *Description of software testing methods*

Software testing method	Description
Unit testing	Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring that a piece of software at the code level are functional and operate as they were designed to.
Integration testing	After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. Then the components are tested as group using integration testing to ensure that whole components of an application behave as expected.
System testing	System testing is a black box testing method used to evaluate the completed and integrated system to ensure it meets specified requirements. The functionality of the software is tested in detail. It is typically performed by a separate testing team than the development team before the product is forward into production.
Acceptance testing	Acceptance testing is the last phase of functional testing and is used to assess whether the final piece of software is ready for delivery. It involves ensuring that the product follows all of the original business criteria and that it meets the end user's needs. This requires the software be tested internally and externally.
Performance testing	Performance testing is used to determine how an application will behave under various conditions. The goal is to test its responsiveness and stability in real user situations. Performance testing is a non-functional testing technique and can be broken down into 4 types: Stress testing, Load testing, Spike testing, Endurance testing.
Security testing	Security testing is a non-functional software testing technique used to determine if the information and data in a system is protected. With the increasing of cyber-attacks and cloud-based testing platforms, there is a growing concern and requisite for the security of data being used and stored in software.
Usability testing	Usability testing is a testing technique that measures an application's ease-of-use from the end-user perspective and is often performed during the system or acceptance testing stages. The main goal is to determine whether or not the visible design and refined of an application meet the intended workflow for different processes, as for example logging into an application. Usability testing is a easy way for teams to overview separate functions, or the whole system, is intuitive to use.
Compatibility testing	Compatibility testing is used to measure how an application or piece of software will operate in different environments. It is used to check that your product is compatible with multiple operating systems, browsers, platforms, or resolution configurations. The goal is to ensure that software's functionality is consistently supported across any expected environment end users will be using.

3. BUILDING AN EDUCATIONAL GAME FOR THE SOFTWARE TESTING EDUCATION

Educational games are tools to support the teaching of educational contents, attractive and dynamic manner. However, before an educational game is used, it is necessary to evaluate if it can support the teaching of the proposed contents [8]. In order to develop quality software testing education game, development process should go through the phases given on the Fig. 1.

Each phase contains its processes. Only by performing all the processes from all phases provides as a result educational game for testing software with high quality. That further means that the student acquires the necessary knowledge in the field of software testing by playing the game [9]. Representation of processes is contained in next phases:

1. Literature review and systematic mapping (Fig. 2),
2. Implementation of the educational software testing game (Fig. 3),
3. Evaluation of the education testing software game (Fig. 4).

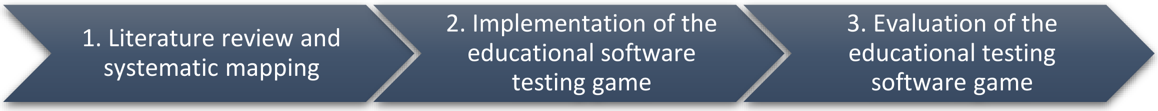
1. Literature review and systematic mapping
2. Implementation of the educational software testing game
3. Evaluation of the educational testing software game

Figure 1. *Development process phases of educational game*

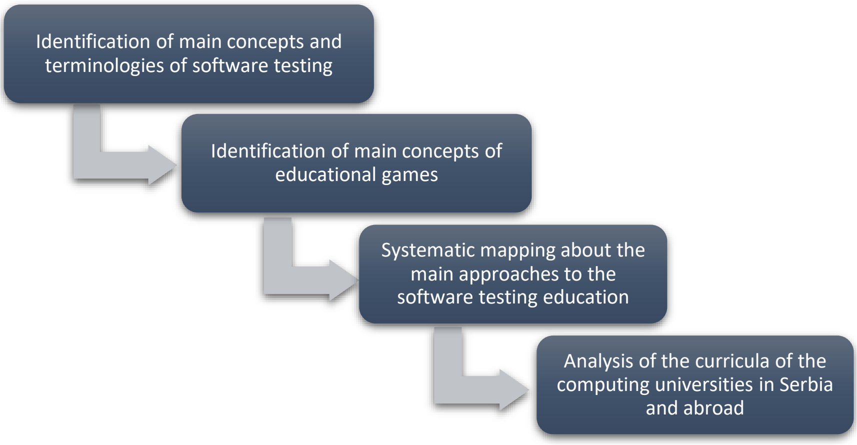
Identification of main concepts and terminologies of software testing
Identification of main concepts of educational games
Systematic mapping about the main approaches to the software testing education
Analysis of the curricula of the computing universities in Serbia and abroad

Figure 2. *Phase of literature review and systematic mapping*

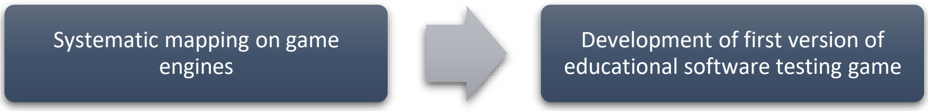
Systematic mapping on game engines
Development of first version of educational software testing game

Figure 3. *Implementation phase of the educational software testing game*

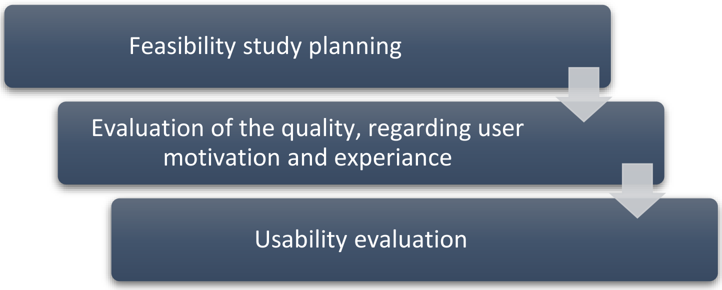
Feasibility study planning
Evaluation of the quality, regarding user motivation and experiance
Usability evaluation

Figure 4. *Evaluation phase of the education software testing game*

After defining methodology used for development of the educational software testing game, it moves on to structuring the game [10]. Software testing methods covered by education game have been presented in Table 1. and there was said that game should cover at least several software testing methods. To accomplish that, multiple game levels are introduced. Each game level is representing one software testing method. Finishing one level of game means that student have learned one software testing method. At each level, there are different challenges regarding the addressed contents [11]. At the beginning of each phase there should be a description informing the students what they must do to succeed in the considered phase (Fig. 5). Another functionality should be related to the theoretical content of software testing. For each new content considered in the game, there should be a module in which the user can access and read about the new content related to software testing.

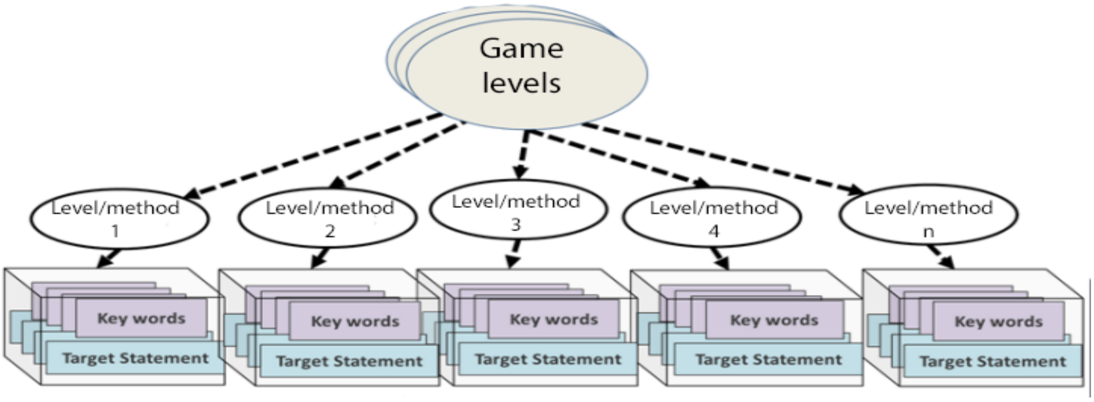

Figure 5. *Education model of each game level*

4. ANALYSIS OF EDUCATION SOFTWARE TESTING GAMES

Based on research in the field educational games in software testing [12, 13], we split effectiveness of those games in three parts:

1. Motivation,
2. User experience and
3. Learning.

Starting with the *motivation* of students, we highlight the key factors when it comes to motivation and these are: Satisfaction, Confidence, Relevance and Attention. To ensure satisfaction, students must understand that they will have the opportunity to use the things learned from the game in practice. Confidence is achieved by making the students feel they are really learning while playing the game. If the material is in line with the interests of students and related to other knowledge they have, the relevance is provided. The most important thing that distinguishes this way of learning is maintaining attention. In order to keep the student's attention, the game must have a variety of content, good design and problems that will interest students to solve them.

The next part of game effectiveness is *user experience*. The key factors in user experience are: Competence, Fun, Challenge and Immersion. If student have positive feelings about the efficiency during the game and was able to play the game trough his skills, then competence is ensured. To assess the fun of the game, we need to ensure that the student wants to play the game again, wants to recommend it to other colleagues, and wants the game to have more levels to accomplish. Game should be made to be challenging, in the way that evolves at an appropriate pace and does not become monotonous, offers new obstacles, situations or variations of activities. Students who play educational software testing game often fell more in the game environment than in the real world, forgetting what was around them, forgetting their day to day tasks, because they are staying focused on the game.

The last and most important factor of effectiveness of the educational software testing game is learning. The knowledge gained in the game should be usable in real life. When it comes to games, a person produces more dopamine hormones, and thus students spend more time learning, in other words, playing an educational game. The challenge ahead is to produce suitable games that exercise the testing methods described in Table 1. of this paper.

These examples would allowing a fine progress of the learner without leading to embarrassment (too challenging) or frustration (too easy). An overarching narrative may help to engage students. Once a software testing curriculum is implemented using these games, a further challenge lies in evaluating the effects on learners. Only if we meet all the parts of the effectiveness we have listed, learning in this way will be more effective compared to traditional learning. It will be easier for students to master the material of subjects that contain content related to software testing [14].

5. EXAMPLES OF THE EDUCATIONAL GAMES APPLICATION IN LEARNING SOFTWARE TESTING

In some countries, the application of educational games in learning software testing has been used for a last dacade. One of the first scientific papers that were created and show the development of games for learning software testing are presented in paper [15]. The games that were initially used mainly included functional testing technique, while structural testing and defect-based testing were less covered. Among the more well-known educational games used to learn functional techniques in software testing are: JoVeTest, U-TES, iTest LEarning and iLearnTest [6].

The other educational game is Bug Hide-and-Seek for reinforces testing principles by demanding students to develop correct solutions or solutions that consciously contain bugs [16]. While finding the correct or buggy solutions, students write corresponding tests. They should identify whether their solution contains bugs or not. The main goal of the game is to hide a smart bug that will mislead other students' tests and despite the hidden bug. The second goal is to recognize correct or incorrect software behavior.

6. CONCLUSION

Although software testing is a critical aspect of the software development process, there is a shortage of professionals in this field. Students usually do not understand the importance of software testing, so they often lack the motivation to learn this skill. To make learning software testing easier and give a clearer picture of the importance, many educational software testing games have been developed. As proven, students spend more time playing games than learning. Allowing students to play games and simultaneously learn software testing can be a winning combination. With educational software testing games, students' motivation increases, as well as the time spent on acquiring knowledge in the field of software testing. In the future work, we plan to survey students and provide an educational software testing game that they will play, give their opinion on this way of learning as well as grading of the achieved knowledge in the field of software testing.

ACKNOWLEDGMENT

This study was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia, and these results are parts of the Grant No. 451-03-68/2020-14/200132 with University of Kragujevac - Faculty of Technical Sciences Čačak.

REFERENCES

[1] Myers, G. J., Sandler, C. & Badgett, T. (2011). *The art of software testing.* New Jersey, USA: John Wiley & Sons.

[2] Paul A. Howard-Jones, Tim Jay. (2016). Reward, learning and games. *Current Opinion in Behavioral Sciences,* 10, 65-72, doi.org/10.1016/j.cobeha.2016.04.015.

[3] Clegg, B., Rojas, J. M., Fraser, G. (2017). Teaching Software Testing Concepts Using a Mutation Testing Game. *ACM/IEEE International Conference on Software Engineering (ICSE)(SEET). IEEE Press*, 33-36.

[4] Fraser, G. and Arcuri, A. (2013). Whole test suite generation. *IEEE Transactions on Software Engineering*, 39(2), 276–291.

[5] Wong, W., Bertolino, A., Debroy, V., Mathur, A., Offutt, J., and Vouk, M. (2011). Teaching software testing: Experiences, lessons learned and the path forward. *XXIV Conference on Software Engineering Education and Training (CSEE&T).* Honolulu, USA: IEEE, 530-534.

[6] Valle, P., Toda, A., Barbosa, E., Maldonado, J. (2017). Educational Games: A Contribution to Software Testing Education. *IEEE Frontiers in education Conference (FIE)*, 1-8.

[7] Thiry, M., Zoucas, A. and da Silva, A. C. (2011). Empirical study upon software testing learning with support from educational game. *XXIII International Conference on Software Engineering and Knowledge Engineering (SEKE).* Miami Beach, USA: IEEE, 481–484.

[8] Ribeiro, T. P. B. and Paiva, A. C. R. (2015). Educational game for learning software testing. *X Iberian Conference on Information Systems and Technologies (CISTI)*. Aveiro, Portugal: IEEE.

[9] iLearnTest: Educational game for learning software testing. *In X Iberian Conference on Information Systems and Technologies (CISTI). IEEE, Aveiro, Portugal, 1–6.*

[10] S. Jia and C. Yang, "Teaching software testing based on cdio," World Transactions on Engineering and Technology Education, vol. 11, no. 4, 2013.

[11] Karla, M., Paul, M. K., Tom, L., Julieta, N., & Luis, N. (2011). Designing and Evaluating Emotional Game-based Learning Environments. *Serious Games and Edutainment Applications*, 245-271.

[12] Pablo, M. G., Pilar, S. T., Iván, M. O., José, L. S., & Baltasar, F. M. (2007). Adaptive units of learning and educational videogames. *Journal of Interactive Media in Education*, 1-15.

[13] Westera, W. (2019). Why and How Serious Games can Become Far More Effective: Accommodating Productive Learning Experiences, Learner Motivation and the Monitoring of Learning Gains. *Journal of Educational Technology & Society.* 22(1). 59-69.

[14] Simpson, E. S. (2005). Evolution in the Classroom: What Teachers Need to Know about the Video Game Generation. *TechTrends: Linking Research & Practice to Improve Learning*, 49, 17-22.

[15] Levy, L. & Novak, J. (2009). Game Development Essentials: Game QA & Testing. *Cengage Learning, 58-70.*

[16] Pfau, J., Smeddinck, J. D. & Malaka, R. (2017). Automated Game Testing with ICARUS: Intelligent Completion of Adventure Diddles via Unsupervised Solving. *CHI PLAY Extended Abstracts,* 153-164.