

Can gamification help in software testing education? Findings from an empirical study.

Raquel Blanco, Manuel Trinidad, María José Suárez-Cabal, Alejandro Calderón Sánchez, Mercedes Ruiz, Javier Tuya

DOI: <https://doi.org/10.1016/j.jss.2023.111647>

Software testing is an essential knowledge area required by industry for software engineers. However, software engineering students often consider testing less appealing than designing or coding. Consequently, it is difficult to engage students to create effective tests. To encourage students, we explored the use of gamification and investigated whether this technique can help to improve the engagement and performance of software testing students. We conducted a controlled experiment to compare the engagement and performance of two groups of students that took an undergraduate software testing course in different academic years. The experimental group is formed by 135 students from the gamified course whereas the control group is formed by 100 students from the non-gamified course. The data collected were statistically analyzed to answer the research questions of this study. The results show that the students that participated in the gamification experience were more engaged and achieved a better performance. As an additional finding, the analysis of the results reveals that a key aspect to succeed is the gamification experience design. It is important to distribute the motivating stimulus provided by the gamification throughout the whole experience to engage students until the end. Given these results, we plan to readjust the gamification experience design to increase student engagement in the last stage of the experience, as well as to conduct a longitudinal study to evaluate the effects of gamification.

Design, implementation, and validation of a benchmark generator for combinatorial interaction testing tools.

Andrea Bombarda, Angelo Gargantini

DOI: <https://doi.org/10.1016/j.jss.2023.111920>

Combinatorial testing is a widely adopted technique for efficiently detecting faults in software. The quality of combinatorial test generators plays a crucial role in achieving effective test coverage. Evaluating combinatorial test generators remains a challenging task that requires diverse and representative benchmarks. Having such benchmarks might help developers to test their tools, and improve their performance.

For this reason, in this paper, we present BenCIGen, a highly configurable generator of benchmarks to be used by combinatorial test generators, empowering users to customize the type of benchmarks generated, including constraints and parameters, as well as their complexity. An initial version of such a tool has been used during the CT-Competition, held yearly during the International Workshop on Combinatorial Testing. This paper describes the requirements, the design, the implementation, and the validation of BenCIGen. Tests for the validation of BenCIGen are derived from its requirements by using a combinatorial interaction approach. Moreover, we demonstrate the tool's ability to generate benchmarks that reflect the characteristics of real software systems.

BenCIGen not only facilitates the evaluation of existing generators but also serves as a valuable resource for researchers and practitioners seeking to enhance the quality and effectiveness of combinatorial testing methodologies.