

## **ANALYSIS OF WAYS TO OPTIMIZE THE MUTATION TESTING SYSTEM FOR SOFTWARE**

*bachelor Taras Kriukov, Vasyl Stefanyk Precarpathian National University, Ivano-Frankivsk, Ukraine*

Mutation testing is a method for assessing the effectiveness of tests in detecting errors in software code, which involves making minor changes, called mutations, to simulate programming errors [1]. These changes may include modifications to logical operators or numerical values. Initially, mutations are introduced to a certain fragment of the code being tested, after which test scenarios are run for each change. If the test suite does not detect any changes in the code and the test is successfully passed, the mutant is considered «alive», indicating gaps in the tests. The final stage involves analyzing the results of testing and calculating the mutation score, which measures the percentage of "killed" mutants relative to the total number.

However, due to high computational demands and time costs, the widespread implementation of mutation testing in software development is quite limited. In this regard, current research is focused on analyzing and developing new methods and systems that can effectively identify and remove "mutants", which are variations of the program that do not change its external behavior, such as Trivial Compiler Equivalence (TCE) [2], Random Mutants Selection (RMS) [3], Selection of Random Operators (ROS) [3], and Selective Mutation (SM) [3].

The report discusses the analysis and comparison of the above-mentioned methods for optimizing the mutation testing system of software implemented in the Ruby programming language.

The application of mutation testing within the scope of quantum computing [4] will be the subject of our future work.

**References:** 1. A. A. Serban, Z. Micskei, "Application of mutation testing in safety-critical embedded systems: a case study" in *Acta Polytechnica Hungarica*, vol. 21, no. 8, pp. 87-106, doi: 10.12700/APH.21.8.2024.8.5 2. M. Kintis, M. Papadakis, Y. Jia, N. Malevris, Y. Le Traon and M. Harman, "Detecting trivial mutant equivalences via compiler optimisations," in *IEEE Transactions on Software Engineering*, vol. 44, no. 4, pp. 308-333, 1 April 2018, doi: 10.1109/TSE.2017.2684805 3. G. Guizzo, F. Sarro, J. Krinke, and S. R. Vergilio, "Sentinel: a hyper-heuristic for the generation of mutant reduction strategies," in *IEEE Transactions on Software Engineering*, vol. 48, no. 3, pp. 803-818, 1 March 2022, doi: 10.1109/TSE.2020.300249. 4. V. Tkachuk and M. Kozlenko, "Improved quantum genetic algorithm on multilevel quantum systems for 0-1 knapsack problem," in *Advances in Artificial Systems for Logistics Engineering*, in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 135, Kyiv, Ukraine, Feb. 20-22, 2022, pp. 51-70, doi: 10.1007/978-3-031-04809-8\_5.