Michał Mnich

Dissertation abstract

The dissertation concerns selected issues of software engineering. The aim of the thesis is to present new methods of optimizing mutation testing or using them in popular software development methodologies. Mutation testing is considered to be one of the most effective methods for testing software code. The disadvantage of this approach is high time and memory consumption. This dissertation addresses the issues of optimizing the mutation process and its application in software engineering processes. The thesis proposes a number of optimization mechanisms to reduce the duration of the code mutation process.

The first mechanism deals with reducing the number of mutants based on the analysis of code changes between different software versions. The second uses a Bayesian approach to optimize the probability of mutant generation from a specific group of mutation operators so as to reduce the number of mutants without significantly reducing the efficiency of the mutation analysis process. The third model deals with the generation of multiple mutants in a single compilation. Theoretical results are presented here, as well as experiments verifying whether the efficiency of the mutational testing process improves when the model is used. The paper also introduces a version of the Test-Driven Development methodology enriched with a step containing mutation testing. It was experimentally confirmed that the use of such an enriched methodology contributes to increasing the code quality.

The model of a self-adaptive, distributed, scalable system was used to perform mutation testing. This system has been implemented by the author of the dissertation as a computational cluster with an implemented method for optimizing the mutation and testing process. The dissertation also includes a description of the architecture of this system.

The dissertation undertakes to justify the following research theses:

Thesis 1. The efficiency of mutation testing does not decrease significantly if mutants are generated only for new or modified code in a new software version.

Thesis 2. Changing the probability distribution of mutation operator draws, done using a Bayesian approach, can be used to significantly reduce the number of mutants with an acceptable small loss in process efficiency.

Thesis 3. The enrichment of the Test-Driven Development approach with mutation testing significantly increases the quality characteristics of the code and increases the independence of testing by reducing the bias of the test code author.

Thesis 4. The approach of introducing multiple mutations into a single compilation has its limitations and not every mutation operator can coexist with all others.

Thesis 5. The approach of introducing multiple mutations into a single compilation for languages where modification of intermediate code is not possible can significantly speed up the mutation testing process.

.....