

WHAT

- Apply evolutionary computation and other meta-heuristic search techniques to generate effective test cases for complex software systems automatically
- Develop a general Evolutionary Testing Framework (ETF) for the automation of test case generation on basis of search techniques with an extensible architecture open for the integration with other automated software testing tools.
- Develop translator, monitoring and evaluation techniques for different test objectives. Provide these techniques as plugins to the Evolutionary Testing Framework. Test objectives that will be considered:
 - Functional (specification-based) testing
 - Functional (structural) testing
 - Temporal behaviour testing
 - Safety testing
 - Robustness testing
 - Error-driven testing (in order to find and reproduce sporadic errors)
- Develop Automated Testing Environments (ATE) for several case studies of different application areas on the basis of the Evolutionary Testing Framework, the translators designed, and the monitoring and evaluation techniques developed:
 - Automating the execution and evaluation of the generated test cases.
 - Employed on a wide spectrum of real world pilot complex software systems defined by the industrial partners (Daimler - automotive, Motorola - telecommunications, SMEs)
- Describe a general procedure how and when evolutionary testing could be applied to various testing problems and objectives. Describe software and system engineering life cycle processes (requirements capture, design, coding, testing, maintenance) to incorporate and adapt the use of the developed techniques and tools.
- Develop a concept how to deal with reproducibility issues of the test. Since each test run is unique through the application of meta-heuristic search techniques, a model for the reproducibility of the tests is necessary in order to reduce the test evaluation effort.