Automation of software avionics verification in accordance with DO-178C standard

Andrei Khasanov, Shishkin Vadim, Larin Kirill

Introduction

Verification is one of the important software life cycle processes, which provides the monitoring process of procedures implementation for DO - 178C [2]. It must detect and record errors that may have been made during design or confirm that there are no possible errors. But verification is very time-consuming. At each stage of verification, the package of documents is developed which confirms the passage of verification and the errors absence (or presence). Usually, the verification process goes step by step from top to bottom, starting from the system requirements related to software, and ending with low-level software requirements (detailed modules description and their interaction). The verifier describes these stages in details in the software verification plan, which is developed separately for each project.

The verification is an integral process. A feature of integral processes is their simultaneous execution with the software design processes throughout the entire software life cycle.
The most time-consuming verification processes are testing and development of supporting documentation.

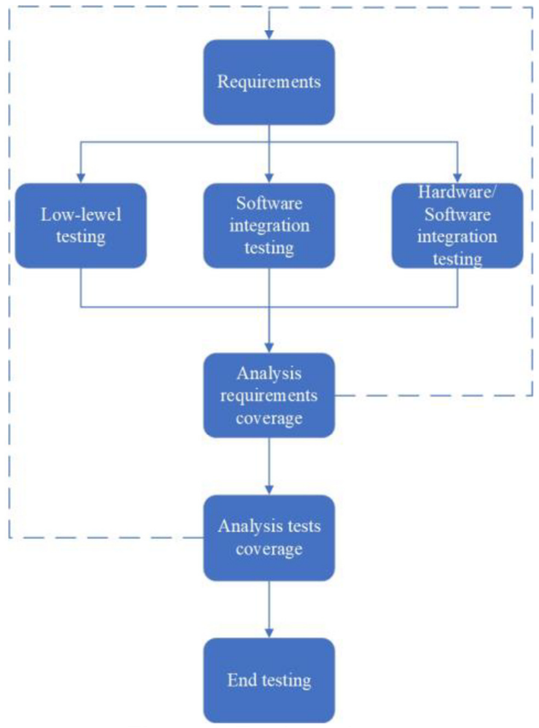

Figure 1. Testing scheme.

Testing Process

Hardware and software integration testing is performed to verify correct software operation on this hardware (in the target computer's environment). At present, due to the rapid growth of software volumes and its complexity, there is a need to develop methods and software platforms for generating test data and accompanying documentation. To resolve this problem, a software package UTSTgen, which generates integration tests based on the requirements and documentation for these tests, has been developed. The test sets generator operation scheme is presented in Figure 2.
To realize this scheme (Figure 2 (a)), special bench equipment is required, which makes it possible to issue input values in the avionics communication line and read the output values. Such testing is limited by the physical capabilities of the hardware and, depending on the complexity of the system, may not allow some input values to be set. Use the test driver (Figure 2 (b)) allows to set the value of any system parameter, while blocking its real value. This feature is useful, for example, when checking hardware checkout functions. To simulate failure conditions, it is enough to change the value of the parameter in the driver, which stores the state of the hardware.

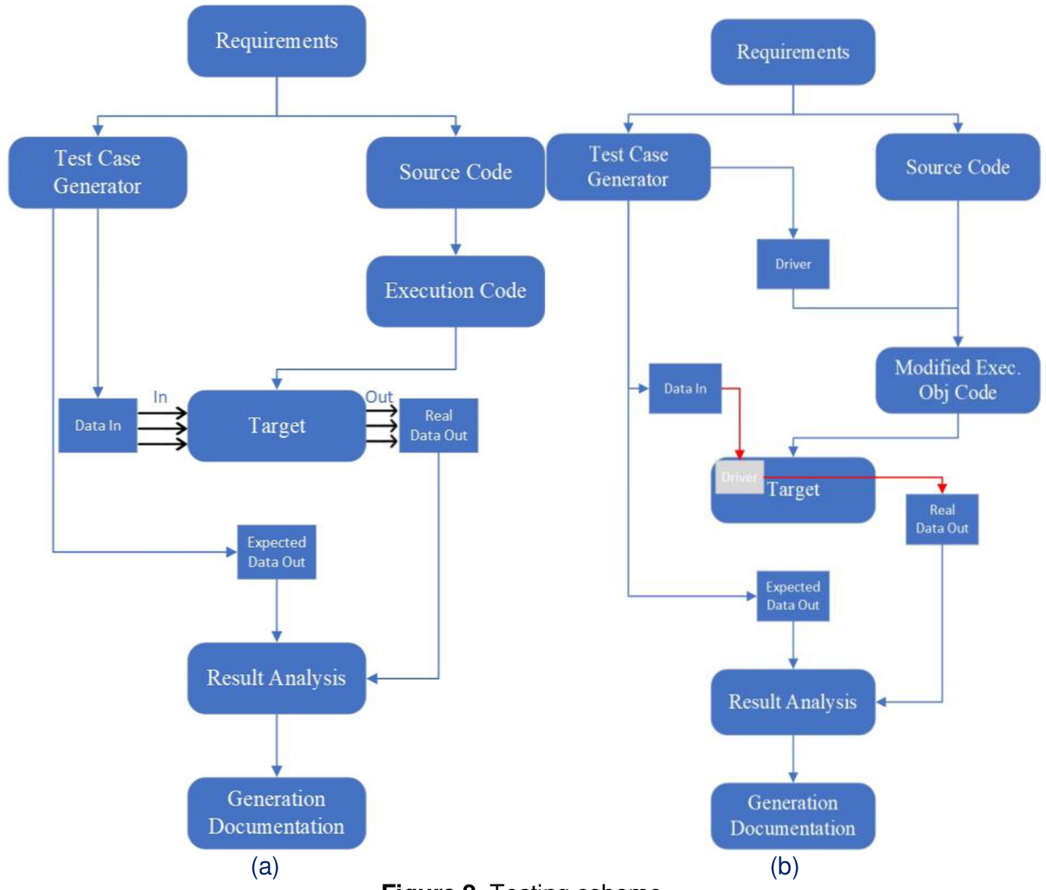

Figure 2. Testing scheme.

Conclusion

Developed UTSTgen software platform makes it possible to significantly speed up and simplify the process of developing test cases, test procedures and documentation for them.
During the measurement of time, the time for formalization of requirements and their validation was not taken into account. The average test case generation time is ~25ms. The validation requirement for a discrete signal is 1ms, for a analog signal it is 3ms. This indicator indicates the high performance of the system. Formalization of requirements is a separate task and is solved depending on the requirements format.
Exclusion from the stage of preparation of supporting documentation of the human factor can improve the quality of testing and the verification process as a whole.

Ulyanovsk state technical university

Severnii Venetz, 32, Ulyanovsk, Russia (www.ulstu.ru)