Automated Software Testing Process

Assist. Paul POCATILU
Economic Informatics Department, Academy of Economic Studies Bucharest

Software testing is an activity whose costs are very high. Manual testing involves a lot of effort, measured in person per month. Using automated testing, with specific tools, this effort can be dramatically reduced and the costs related with testing can decrease.

Keywords: *software testing, manual testing, automated testing.*

1 The testing process

Software testing is a very important process in software development cycle. Through testing, the quality of final software application can be improved. But software testing is a very expensive process and it can consume a lot of resources as money, time and people.

The automation of software testing consists of a series of processes, activities and tools brought together in order to execute the software under test and to record the result of the tests.

The testing process has the following activities:

- test planning
- test design
- test implementation
- test execution
- test evaluation

A general testing process is depicted in figure 1.

Each activity has specific deliverables that are used from a phase to another. At the end, bug reports and other documentation will result. These documents are used by the development team to identify the cause of faults and to correct them.

After the test plan has been elaborated, based on specific inputs (budget, resources, timeline), the next step is to analyze the requirements and to define the objectives of testing for the test team.

The design phase is focused mainly on the definition and design of test procedures. At this time a decision will be made about what should be tested manually and what will be tested automatically.

Test cases and test procedures are the result of test implementation phase. Test scripts are written in specific programming languages like Visual Basic, Java or C++. In this phase, some test scripts can be reused from the previous tests.

Test execution has as an input the test plan and the test procedures. After the execution of the tests, the results are evaluated using an oracle. An oracle is a specialist that could decide if the result is correct or not.

2. Tools used in automated software testing

Tools for automated software testing are various, and they can be used in different areas of testing. At this moment there are many tools to assist software testing: capture/playback tools, tools for automated execution of tests, coverage analyzers, test case generators, logical and complexity analyzers, code instrumentation tools, defect tracking tools and test management tools.

Capture/playback tools are used to record testing sessions in script files, allowing to playback them later. It can be made multiple tests and the results can be compared. Those tools are useful in regression testing.

Tools for automated execution of tests are similar to capture/playback tools, but the test cases are specified by the user in script files.

Coverage analyzers are used to evaluate how much of the structure of the tested code has been covered by the test cases. These tools are useful for identifying the code sequences that were not covered by the tests.

Test case generators are based on information such as requirements, data models, and object models used to generate significant test cases. The main advantage of using these tools is that the testing redundancy is eliminated by finding the test cases that assure the best cover for the code.

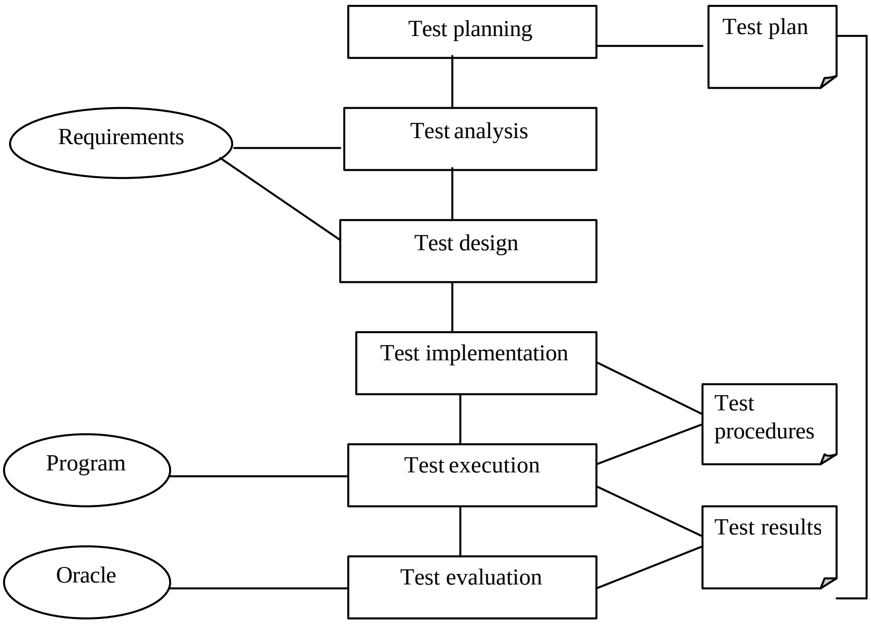

Figure 1 – The automated testing process

Test data generators are used to fill in files and databases with data for testing. Data are chosen randomly or based on some specific conditions. These tools are usually used for large volume of data necessary in operational testing and load testing.

Logical and complexity analyzers are used to quantify the complexity of the code. Many of these tools can show the graphical structure of the code. These tools are useful to find the necessary test cases to execute some points of the code in complex modules.

Tools for code instrumentation are used for analysis of source code of the program under test and inserting calls to specific functions in order to gather information about program while it runs.

Defect tracking tools are used to manage the information regarding the detected errors, their status and to centralize that information in order to provide information regarding the trend of the faults. Some steps of the software development cycle can be improved based on the observed trends.

Tools for test management are used to assist in planning and organizing of all elements involved in testing such as: script files, test cases, test reports and tests results.

The main producers of software testing tools are *Rational* (Rational TestFactory, Rational TestManager, Rational Robot, and Rational ClearQuest,) and *Mercury Interactive* (WinRunner, LoadRunner, TestDirector).

3. People specialization

In order to use automated software testing, people need to be trained. The responsibilities of automated test engineer are [DUST99]:

- to develop test procedures and test cases based on requirements
- to execute manually the test procedures
- to test procedure walkthroughs
- to conduct tests
- to prepare reports regarding test progress and regression
- to use the test standards.

The minimum required skills for an automated test engineer are:

- experience in software testing
- experience in designing test suites
- familiar with GUI design and GUI design standards
- familiar with the business area of the software that is developed
- experience in programming.

People without knowledge of automated tools can also be part of testing teams, where they will be given specific tasks in the testing process.

4. Costs implied by the automated software testing

The costs related of automated testing include:

- cost of software tools
- cost of hardware
- cost of salaries for automated test engineers
- cost of training people
- percents from the pay off for equipments and software tools used in testing process
- maintenance expenses for equipments and tools
- expenses for analysis the behavior of the product based on the test data generated or on the results of the previous tests.

Compared with manual testing, the cost of automated testing is higher, especially at the beginning of the automation process. Software testing tools and hardware equipment (if required) are very expensive. The return of investment will be positive after a period of time of using the automated testing.

There is no cost when running of automated tests compared with the manual testing, which implies cost for each time a test is run, depending on the size of the testing procedure.

The total cost of testing is defined as the sum between the cost of manual tests and the cost of automated tests:

$$C_T = C_M + C_A + a \ ,$$

where α are defines other expenses.

5. Conclusions

Software testing cannot be automated completely. Some tests still have to be done manually. There are specific test where automated tools are of no use.

The automation of testing process can decrease the testing associated costs by reducing the time spent on creating and running the test cases. This decrease of testing costs will appear after a period of time, after the use of the automation tools.

References

[DUST99] Dustin, Elfriede, Rashka, Jeff, Paul, John – *Automated Software Testing*, Addison Wesley, 1999

[EFTI02] Eftimescu, Despina, Ilioiu, Alexandru – *Automated testing*, Net Report, April 2002, pp. 22-26 (in Romanian)

[KORE90] Korel, Bogdan, *Automated Software Test Data Generation*, IEEE Transactions on Software Engineering, vol. 16, No. 8, August 1990, pp 870-879.

[IVAN99] Ivan, Ion, Pocatilu, Paul *Object Oriented Software Testing*, INFOREC Publishing House, Bucharest, 1999 (in Romanian).

[POCA02] Pocatilu, Paul, *Software Testing Costs*, in Informatica Economica vol. VI, nr. 1, 2002, pp. 90-93 (in Romanian)

[POCA02a] Pocatilu, Paul, *Testing electronic commerce applications*, in Proceedings of ISIS'2002 Symposium, Iasi, October, 24-26, 2002 (in Romanian)

[ROBI00] Robinson, Harry, Intelligent Test Automation, Software Testing & Quality Engineering, September/October 2002, pp 24-32