Rapid Testing Methods in Latvia

Girts Mamontovs, Larissa Zaitseva

Abstract: *The paper studies the processes comprising rapid testing. The classification of the testing methods is suggested, including both classical and recently developed methods. To identify rapid testing methods usage in Latvia a survey has been made among Latvia leading IT companies employees. Results of this work are described.*

Key words: *Testing, Rapid Testing, Testing Methods*

INTRODUCTION

Nowadays computer software is widely used in all aspects of our life – it's hard to imagine any peace of hardware (starting from cell phone and completing with cars) without software operated it. Software development speed has risen dramatically to meet all those software requirements. The same for software testing – each day there is more software to test. So new approaches are needed to increase a speed and a quality of a testing process. There are couple of new methods that can be used in testing. However classical testing theory is still in place and being used for the most software projects.

There are several definitions of software testing. Most commonly used in the Latvia states that software testing is a process of analyzing and operating software for the purpose to find bugs. So, it can be divided into two basic parts (Figure 1) – analyzing and operating [4].

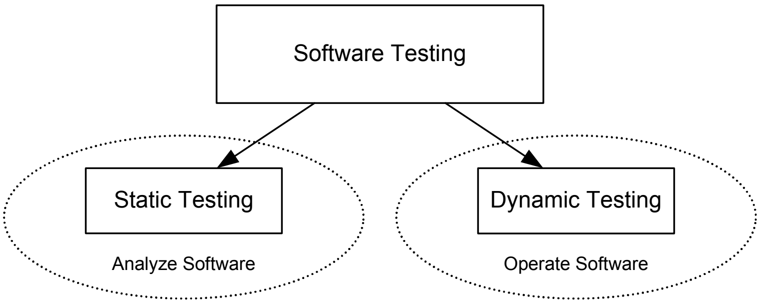

Figure 1. Software Testing

Testing process related to the software analysis is called a Static Testing. This type of testing refers to testing something that's not running – examining and reviewing it. Static testing is mostly used in the early stages of the software development project such as requirements specification and design.

Testing process related to the software execution is called Dynamic Testing. This is what you would normally think as of testing – running and using the software. Dynamic testing is mostly used during development and testing stages of the software development project.

At present there are lot of different tools created to support rapid software development. One of ways to reduce testing schedules and improve current testing process quality is called Rapid Testing.

BASICS OF THE RAPID TESTING

The term "rapid testing" is similar to term "rapid development" introduced by Steve McConnell [2]. As he has pointed out, rapid development means different thing for different people. To some people its user involvement in a development process, to others special tools usage (e.g. CASE) or very tight schedules. Rather than identify rapid development with a specific tool or method, McConnell [2] makes the following definition: Rapid development is a generic term that means the same thing as "speedy development" or "shorter schedules." It means developing software faster than you do now. A "rapid development project," then, is any project that needs to emphasize development speed [2]. In a similar vein, *rapid testing* means testing software faster than you do now, while maintaining or improving your standards of quality [1].

Rapid testing is supported by four major components:

- People.
- Integrated test process.
- Static testing.
- Dynamic testing.

If any of these components is weak, the effectiveness of testing will be greatly impaired [1].

Testing without good test personnel is impossible. Even there is a very good testing process in place and all used methods are adequate effectiveness of the testing will be very low. So the first step of testing is to provide a good training for all stuff on implemented process and testing methods.

Integrated test process is another important thing each software development company must have. It allows joining all the software development activities (e.g. design, implementation, testing) into one complete process where each phase is well planned and fits into big picture.

Static testing provides formal verification process at the early stages of the software development project so that potential bugs can be eliminated at the very beginning of the work. It involves inspections, walkthroughs, and peer reviews of designs, code, and other work products, as well as static analysis to uncover defects in syntax, data structure, and other code components [1].

Dynamic testing shows it real power when used during implementation phase (unit testing) and followed up in testing phase of the software development project. Generally speaking, dynamic testing consists of running a program and comparing its actual behavior to what is expected [1].

RAPID TESTING METHODS

There are many testing methods available. Mostly all of them are used to identify some specific defects in the requirements/design documentation or code. Usually each testing method is designed to identify some specific type defects so it's important to know what testing methods currently are being used in the software development projects.

The classification of testing methods is shown in Figure 2. Nowadays two testing methodologies are available: static testing and dynamic testing.

Static testing is a very important part of the testing process because it's allows to identify defects at the early stages of the software testing so there are only few testing methods available. Most commonly used static testing methods are:

1. Inspections.
2. Walkthroughs.
3. Peer reviews.
4. Audits.

Inspections and walkthroughs have lot in common. Both involve a team of people reading or visually inspecting requirements, design documentation or code. The objective of such a meeting is to find errors and suggest alternatives if any. Usually inspection is a set of procedures and error-detection techniques for group code reading. Walkthrough, like the inspection, is a set of procedures and error-detection techniques however there are slight differences. Rather than simply reading the program or using the error checklists, the participants play computer. The person designated as the tester comes to the meeting armed with a small set or paper test cases [3].

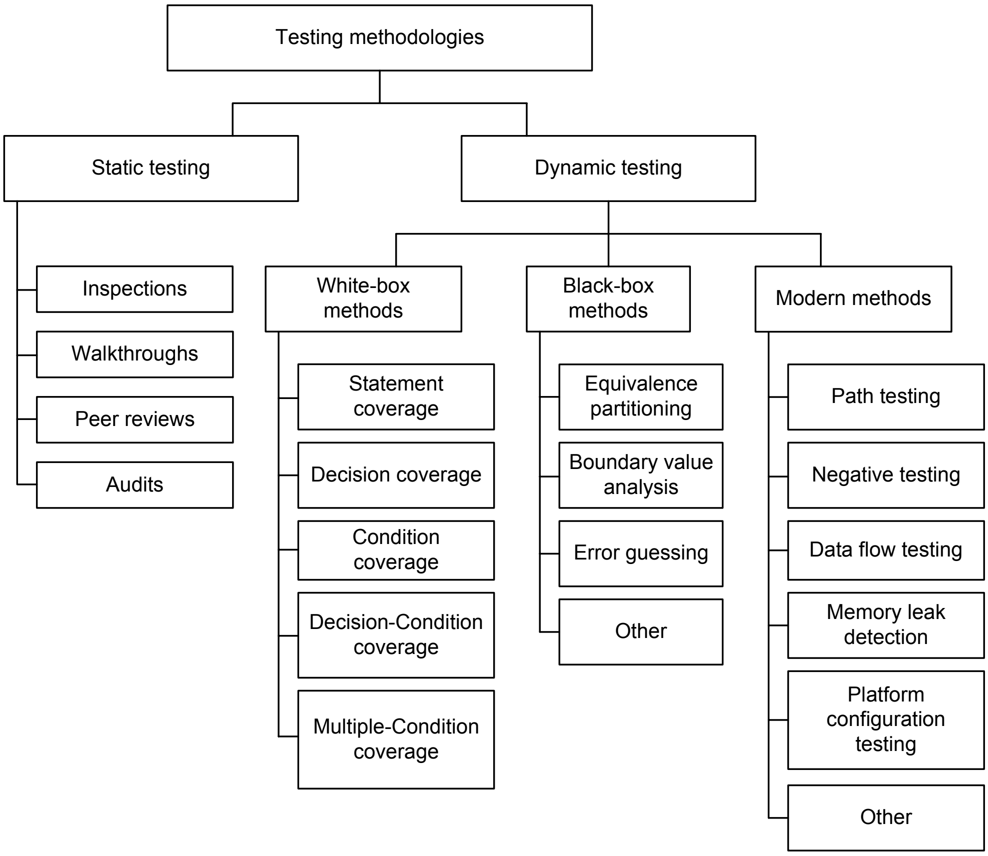

Figure 2. Testing methods classification

Peer review is a technique of evaluating anonymous programs in terms of their overall quality, maintainability, extensibility usability and clarity [3].

Audit is similar to inspection however it's being performed by external group who reviews requirements, design documentation and established software development process as well.

Dynamic testing is another important part of the testing process. It's being used starting from implementation phase (unit testing) and followed in the following testing stages. Most commonly used dynamic testing methods are:

1) White box techniques (statement coverage, decision coverage, condition coverage, decision-condition coverage, multiple-condition coverage).
2) Black box techniques (equivalence partitioning, boundary-value analysis, error guessing).
3) Modern techniques.

White box testing is used in software testing to check that the outputs of a program, given certain inputs, conform to the structural specification of the program. Testing is done with knowledge of the code used to execute certain functionality. For this reason, a programmer is usually required to perform white box tests.

Black box and modern techniques are used in software testing to check that the outputs of a program, given certain inputs, conform to the functional specification of the program. The term black box indicates that the internal implementation of the program being executed is not examined by the tester.

Some of the well-known black box testing methods are:

- Equivalence partitioning – set of test cases that tests the same thing or reveals the same bug.
- Boundary value analysis – situations at the edge of the planned operational limits of the software.

- Error guessing – enumerate a list of possible error or error-prone situations and then write a test cases based on this list.

Less popular are modern testing methods. These methods usually allow identifying some specific type of defects so it's important to include them in testing. Some of the modern test methods are:

- Path testing – all paths in the code are identified and prioritized, only most important are verified. Path usually is started at the condition entry point and completed on condition end point or next condition entry.
- Negative testing – software validation in the broken environment, e.g. printing to the offline printer.
- Data flow testing – initial entry data set is used to create exit data set. When any changes to code are introduced test is performed once more using the same entry data set and new exit data set is compared to the previous one.
- Memory leak detection – used for memory leak detection is the software.
- Platform configuration testing – validates software using different hardware sets.

Good testing process is supported by white box testing methods, followed up by black box testing methods and modern methods.

RAPID TESTING IN LATVIA

To identify rapid testing trends in Latvia a survey has been made among Latvia leading IT companies. Totally 53 people returned complete questionnaires – project managers (19%), developers (25%), testers (41%). 15% of the respondents hold office that is not primary related to the software development projects (quality auditors etc).

As shown in Figure 3 around 46% of respondents are informed about rapid testing basic concepts and it's benefits.

The most strange thing is that even half of respondents are informed about rapid testing and only 14% are claiming that they are really trying to use it in their everyday work (Figure 4).

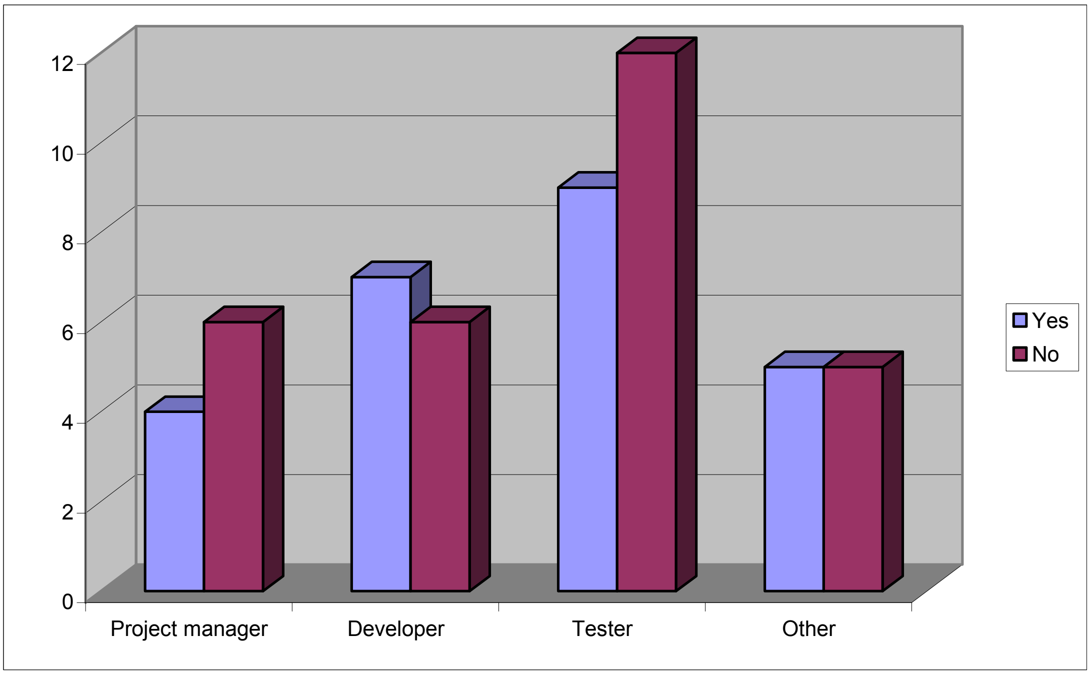

Figure 3. Knowledge about rapid testing concepts

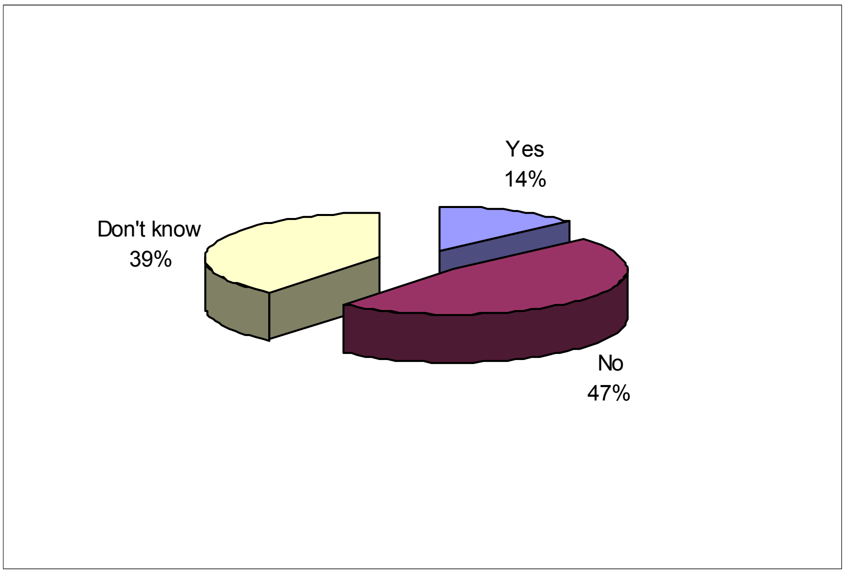

Figure 4. Rapid testing usage in Latvia leading IT companies

Figure 5 shows that most popular static testing technique in Latvia is inspections (used in 77% software development projects) followed by the walkthroughs (71%). 15% has been audited and unfortunately approximately 5% does not what static testing techniques they are using if any. The good news is that only one responded noticed that in his project there were no static testing used.

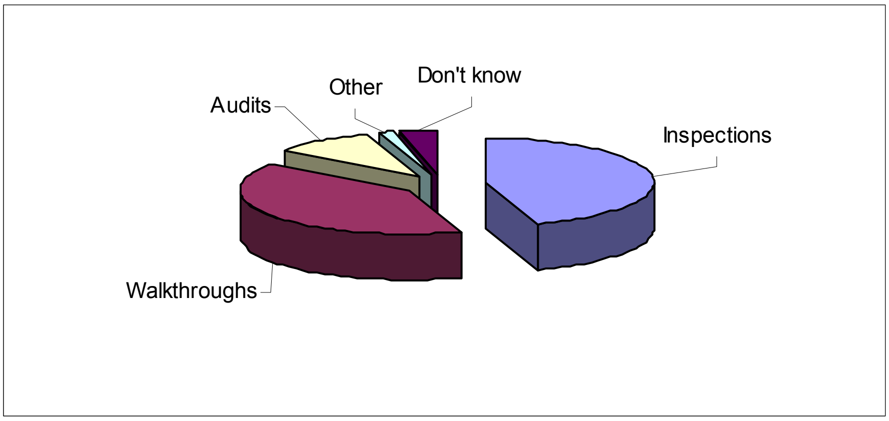

Figure 5. Static testing techniques in Latvia

Figure 6 shows that negative testing and data flow testing are most preferred in Latvia leading IT companies - 55% of respondents are using both of these testing methods. The next favorite is path testing followed by boundary-value analysis method (43%) and memory leak detection (41%).

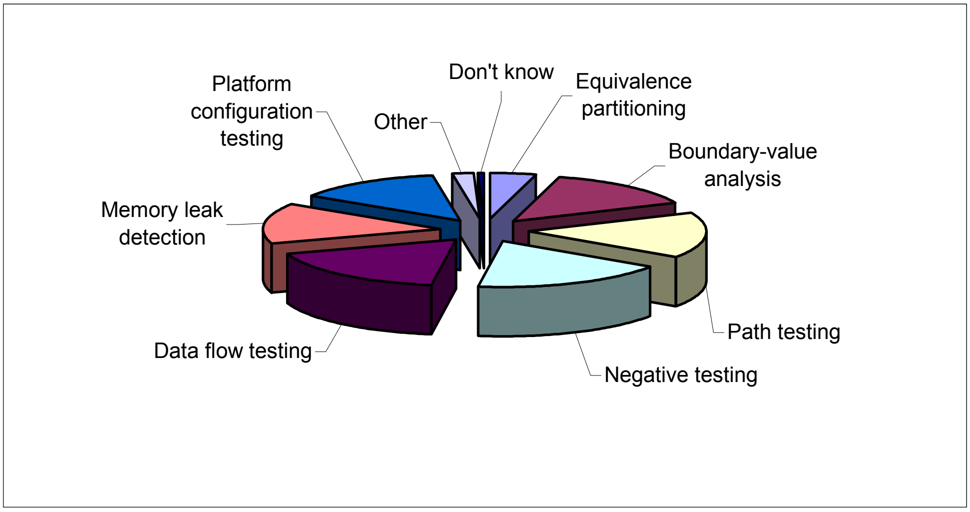

Figure 6. Dynamic testing techniques in Latvia

CONCLUSIONS AND FUTURE WORK

Rapid testing can be successfully applied to every software development project. Well-trained people, integrated development process and proper use of testing methods – these are key elements to achieve better schedules and lower development costs.

Unfortunately only 14% of respondents are using rapid testing ideas in their every day work. The good news is that mostly all of them are using static and dynamic testing techniques during all development process. More work should be done to improve existing development process to merge design, implementation and testing activities into one huge software development process.

The classification of the available software must be explored to suggest most applicable testing methods for each of the software type.

REFERENCES

[1] Culbertson, I., C.Brown, G. Cobb. Rapid Testing. - New York : Prentice Hall PTR, 2002. - 399 p.

[2] McConnel, S. Rapid Development. - Microsoft Press, 1996. - 647 p.

[3] Myers, G. J. The Art of Software Testing. 2nd edition. - John Willey & Sons, Inc., 2004. - 234 p.

[4] Zaiceva, L. Programmatūras izstrādes tehnoloģija. - Riga : RTU, 2002. - 244 lpp.

ABOUT THE AUTHORS

Girts Mamontovs, Mr. sc. ing., Software Engineering Department, Riga Technical University, Phone: +371 7089571, E-mail: girtsm@cs.rtu.lv

Prof. Larissa Zaitseva, Dr. sc. ing., Software Engineering Department, Riga Technical University, Phone: +371 7089522, E-mail: Larisa.Zaiceva@cs.rtu.lv