

Testing Lifecycle: Don't be a fool, use a proper tool.

Zdenek Grössl and Lucie Riedlova

Abstract. Show historical evolution of testing and evolution of testers. Description how Testing evolved from “random clicking” through application to mature lifecycle with aspects of testing loop: Management, Analysis, Design, Execution and (to close the loop) Management again. During the lifecycle of testing there is a great benefit if testers use some “Testing supportive tools”. MS Excel could be used, but... We present our experiences with testing tools during the whole testing lifecycle. We describe the needs, usage and benefits of several tools during the lifecycle of testing. At the end of each phase there is a summary of properties a tool should have. Main message: If you want to use testing tools, make sure you know what you want to get from this tool. And if you need any help, NESS can give you various hints and tips.

Keywords: lifecycle, management, analysis, design, execution, tools, tool, support, reporting, test set, coverage, requirements

1 Evolution of testing

Question: “Why do we test software?” We believe that we all know the answer: To ensure that product that we give to our customer will fulfill his or her needs.

SW developers perform their testing, but only at the level of debugging the code, additional testing is performed by them very rarely. Someone called “Tester” is needed to click through the application before we hand it over to the customer or before we release it to their production environment.

1.1 Test execution

We hope we can agree on the fact that in the beginning there was Test execution. Testers found bugs, reported to SW developers and they fixed the bugs. Test execution standalone has many disadvantages: We cannot tell how much there is to be tested; we don't know when this testing phase ends; it is usually uncoordinated; we can have troubles with regular

repeating of the same tests and we don't know where we are according to the testing phase. If a bug is found we cannot exactly tell what is affected by this bug.

1.2 Test analysis and design: background of execution

Execution itself is not enough. Some preparation is needed before execution begins. In this moment disciplines as Test Analysis and Test Design appear and are involved.

Test analysis tells us WHAT we should test. We can cover Requirements (RQs) by Test Cases (TCs) and trace them. In case a bug is found we can exactly tell which RQs are affected and we can easily repeat execution of TCs. We evaluate TCs, but we have no feedback about the analysis and testing itself.



Fig. 1. From Test analysis and design to Test execution

When we need to repeat huge amounts of tests, when we need to automate tests, for test data management and test script management we are talking about Test Design. Test design gives us answers to question of HOW should we test. Test design helps us with automated tests, will help us with management of test scripts, their versioning, and keep them actual. Test design can help us manage our test data; we will use Test design techniques when we need to perform the same test many times with different entry data. Very often we need to anonymous test data loaded e.g. from Production environment of client.

1.3 Test management: Coordinates testing activities

We currently have three roles in a game, which may cause uncoordinated activities, gaps, idle times, duplication of effort etc. We need someone to

guide us through the project. There is a need of Test management as a discipline, which is connection between testing team and the project management. Test manager is someone who is responsible for the whole testing process.

Test management contributes with an overview, coordination and prioritization of activities; with risk management; reporting to management and communication of necessary info. Test management enables iterative approach, evaluates testing as whole process and gives feedback which should be used within next "testing round" – next iteration.

Hence the loop is closed. From now on we can talk about Testing Lifecycle.

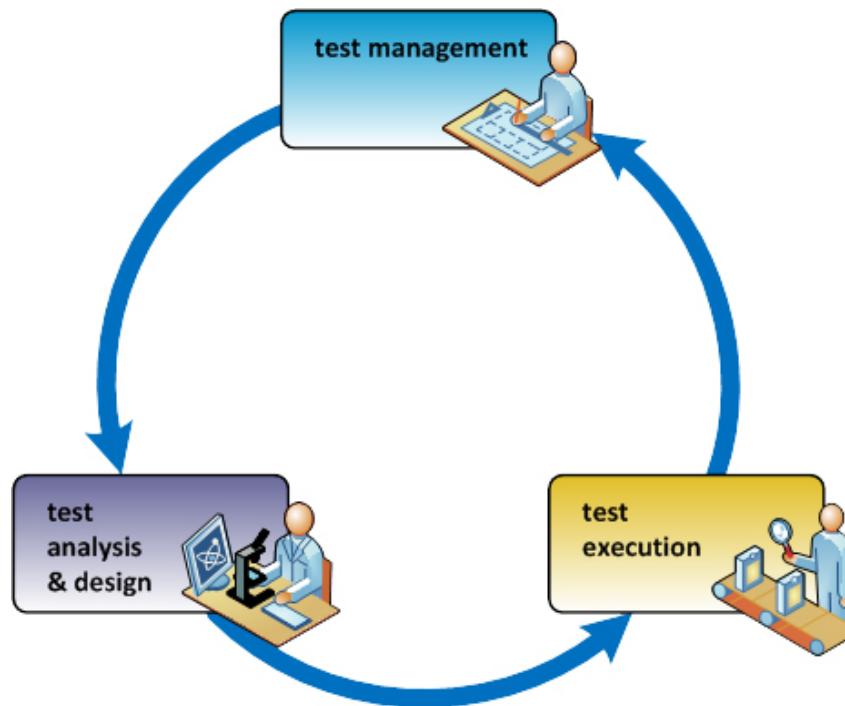


Fig. 2. Testing lifecycle

2 Testing tools

All four roles mentioned in the testing lifecycle can now start working. But please consider the possibility of making their effort more efficient. They can use Testing tools. How can these tools help us? Which one is the best? What should we focus on when we have to choose one? We will go through the whole lifecycle and we will show you how testing tools helped us to test better. And by the words “to test” we mean performing all activities, which will successfully move us to the next testing phase, until the target product fulfills the Acceptance criteria, defined in advance by project management and customer.

2.1 Testing tools from the Test management point of view

Project manager informs us about upcoming project where our testing skills will be needed. We get at least the rough info which should help us to decide our approach to testing. Shall we use automated tests or manual? If there is some third-party involved, we should be informed now to take this into our future plans. It is possible that by this time we already know the delivery schedule and milestones. It is useful to start planning human resources that will be necessary for the project.

Several tools we use for Test management:

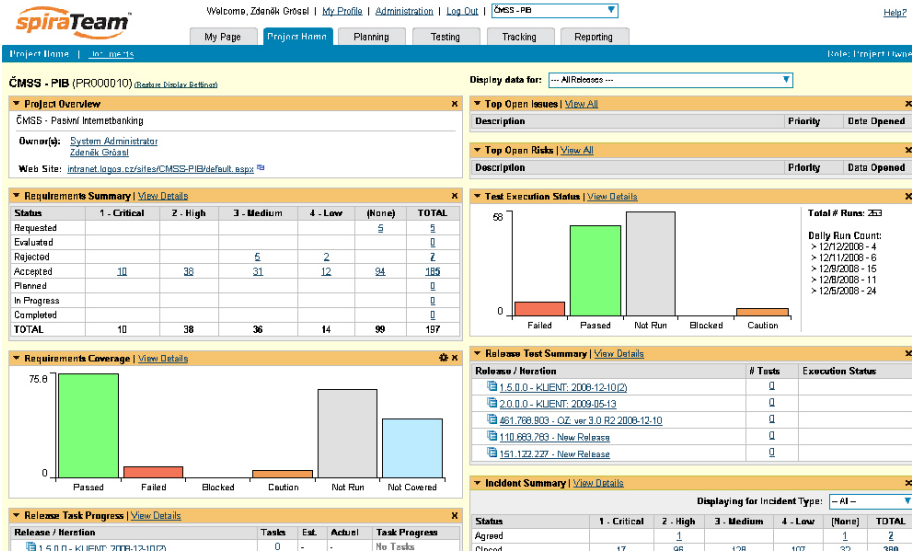


Fig. 3. Spira Team – Project Home page (cut)

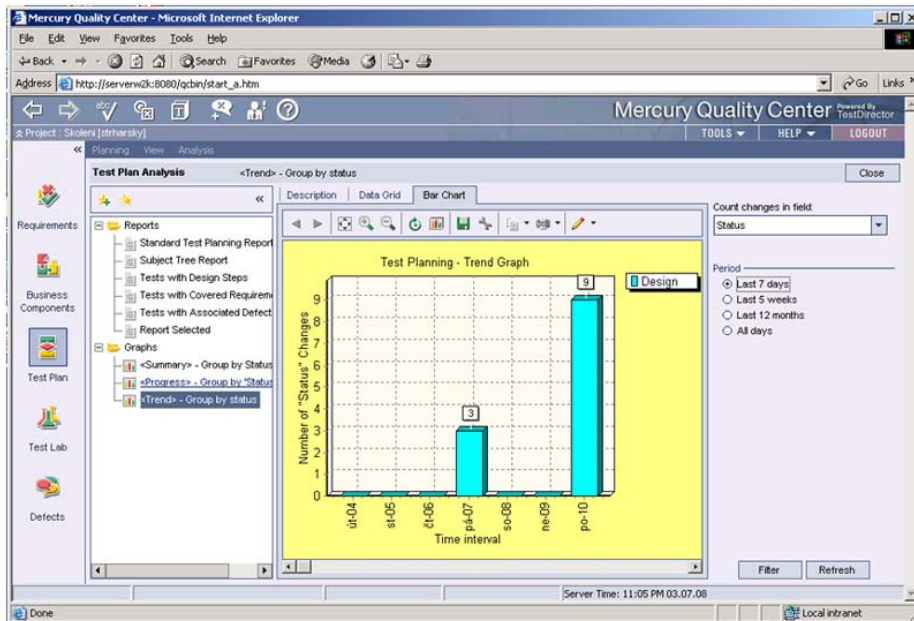


Fig. 4. Mercury QC – Test Plan Analysis

2.1.1 Properties of useful tool

You can find many test management tools with various functions. We dedicated our presentation to those properties important for us in daily use. The tool should cover needs of a Test manager. These needs may vary from project to project and should cooperate with needs of the rest of the team. Try to figure these needs!

2.2 Testing tools from the Test analysis and Design point of view

2.2.1 Test analysis

Analysts in cooperation with clients specify the task and collect requirements. At this point it is necessary for the test analyst to perform the following: Revise collected RQs; sort them to categories (at least into the functional and the non-functional) and select appropriate tool for RQ management. Further, analysts should evaluate the risks – set the priorities and severities of the tests and the main activity: which levels and types of test are necessary to perform to ensure quality.

Analysts prefer using MS Excel at meetings with customers – this tool is simple and is widely used by customers. However, we face several inconveniences. One of them was poor support of requirements catalogue versioning. Customer may have other version than the analyst. This is why it is important to find appropriate system for online sharing of document so that all sides know what version is valid. If a customer insists on MS Excel, you can use SharePoint, for example. The access by using web browser is available not only to team members but also to other participants. We used SharePoint as a repository and we stored documents in standard format. The need to have current version of the document was fulfilled. However, it was only this very need.

If – as test analysts – we have to relate TC and RQ, we usually have received from analysts list of requirements, use cases list, and if we are lucky complete analysis. Tools can help us e.g. to relate these artifacts.

Examples of tools we use for test analysis:

Spira Team, Test Director, MS Excel...

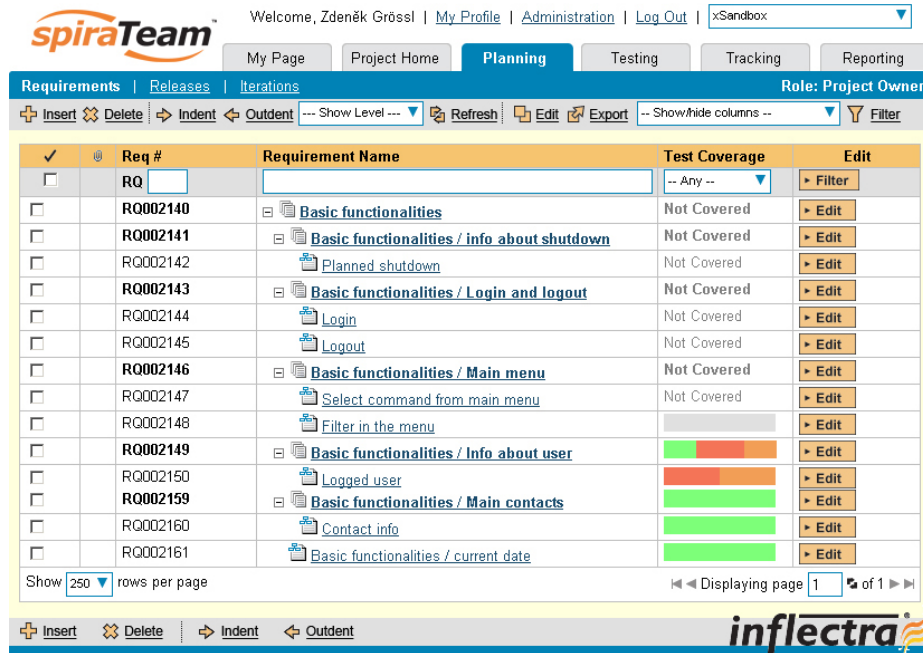


Fig. 5. SPIRA TEAM – RQ coverage overview

2.2.2 Test design and test data preparation

When we know WHAT we want to test we need to know HOW to test it. We need to know how to use tests, test scripts, testing data. Also, in case of frequent regression tests we might have certain problems. Here, we describe several cases when we cannot work without testing tools:

Examples of tools we use for Test Design:

Spira Team together with Selenium; Soap UI; Visual Studio 2005 Team edition for Testers; Test Complete; Load Runner...

2.2.3 Properties of useful tool

You can find many testing tools with various analytical or designing functions. We dedicated our presentation to those properties important for

us in daily use. The tool should cover needs of a Test analyst & designer. These needs may vary from project to project and should cooperate with needs of the rest of the team. Try to figure these needs

2.3 Test execution

Test execution is probably the most important part of the lifecycle as bugs are found during this phase. If test execution should be coordinated and efficient, it is useful to use testing tools. Here are our experiences with several tools.

Examples of tools we use for Test Design:

Spira Team, JIRA, Mercury QC, Selenium, SOAP UI, Load Runner, Test Complete, Excel...

The screenshot displays the JIRA issue tracking tool with a NESS customized design. The interface is divided into several sections:

- Navigation Bar:** Includes links for HOME, BROWSE PROJECT, FIND ISSUES, CREATE NEW ISSUE, and ADMINISTRATION. A search bar is also present.
- Issue Details Sidebar:**
 - Key:** GEUFODEV-774
 - Type:** Install request
 - Status:** Closed
 - Resolution:** Fixed
 - Priority:** Major
 - Assignee:** Šveida Libor
 - Reporter:** Šveida Libor
 - Votes:** 0
 - Watchers:** 0
 - Available Workflow Actions:** Reopened
 - Operations:** Assign this issue (to me), Attach file to this issue, Attach screenshot to this issue, Clone this issue, Comment on this issue, Create sub-task, Delete this issue, Edit this issue, Link this issue to another issue, Move this issue, Voting (cannot vote on resolved issues), Watching (not watching this issue, watch it to be notified of changes)
- Main Content Area:**
 - Issue Title:** GE - UFO Development: Engine R16 na UAT
 - Created:** Thursday 15:22 | **Updated:** Thursday 16:29
 - Component:** Engine
 - Security Level:** External Bugs (bugs reported by external testers)
 - UFO Environment:** UAT
 - Link to install package:** u:\Space\LogosSpace\Engine\3.4.16\Core\Bin\engine-3.4.16.2.ear
 - Navigation:** Return to search, Issue 1 of 19 issue(s), Previous, Next
 - Actions:** All, Comments, Work Log, Change History, Sort Order
 - Comments:** There are no comments yet on this issue.
- Footer:** Powered by Atlassian JIRA™ the Professional Issue Tracker. (Enterprise Edition, Version: 3.6.5-#161) - Bug/feature request - Contact Administrators

Fig. 6. JIRA (with NESS customized design) as bug tracking tool

2.3.1 Properties of useful tool

You can find many testing tools with various functions. We dedicated our presentation to those properties important for us in daily use. The tool should cover needs of test executive site. These needs may vary from project to project and should cooperate with needs of the rest of the team. Try to figure these needs

3 Overall summary

The perfect testing tool does not exist. Why? Because the meaning of "PERFECT" cannot be measured. Every tool suits something and in case you have very complex tool which takes care about everything: Is this tool still easy for you to maintain? Or do you spend too much extra time on maintenance, administration etc?

If you want to use tool as "perfect" as possible, there are several things to be aware of. See our presentation where we mention some of key features (apart of the lifecycle).

4 How do we test at NESS Czech?

We perform all kinds of tests, static and dynamic, we design tests for any phase of project and we execute them as well. Details depend on current project.

4.1 Which tools do we use?

We use any tool that suits the needs. We – Lucie and Zdenek – like the SPIRA TEAM (and several our customers too). When we do testing at client's side and they insist on MS Excel, we try to convince them that there are alternatives too. List of other tools could be very long, and as partners of HP, IBM and others companies we use their tools as well. Mercury QC, JIRA (Atlassian), SPIRA TEAM, TestComplete, Load Runner, SOAP UI... and many others. What are our most powerful tools: **Open minds!** That is why our clients want us to work for them.

4.1.1 SPIRA TEAM (by Inflectra Corporation)

As we mentioned SPIRA TEAM several times, let us tell you few words about this tool.

Spira team brings complete QA solution in one package. It is server-based application with SQL database and it is approached via web interface. SPIRA has very user-friendly environment, which was appreciated by our customers, who were not primary IT-oriented. Bonus of this tool was in the reporting. You can choose between all kinds of reports, output format, included info and some additional filters. With one of last versions came Document repository, which is quite useful feature. You can connect SPIRA with other tools using various plug-ins. All this can be found on Inflectra web pages at <http://www.inflectra.com>.

What we appreciated with this tool was very intuitive environment. We got positive feedback from our clients, even from non-IT users. After one year we have over 30 projects and over 60 external users of SPIRA. We call this successfully established tool.

4.1.2 JIRA (by Atlassian)

We are partner of Atlassian and we use JIRA as bug tracking tool. Our instance of JIRA is used by more than 2000 users on more than 50 projects as system for bug tracking, project management, help-desk etc. JIRA supports agile SW development and is used by more than 10.000 companies around the world.

If you face any problems you can log it within seconds into JIRA; as manager you can have all activities on all your projects very well organized, as JIRA is designed with both business and technical users in mind. If you want all your internal tools to follow company design rules, you can customize JIRA as well. And administration of JIRA is very easy as well.