



VI - Test tools



VI- Test tools

00 - Agenda

Chapter VI – Test tools

- VII/01 Types of test tools
- VII/02 Effective use of test tools
- VII/03 Introduction of test tools in an organization
- VII/04 Summary

VI- Test tools

01 – Types of test tools

General remarks

- Test tools may be used to support test activities
 - Test execution support is referred to as test automation
 - Test tools may also support other test activities.
- Test tools are named after the type of support they provide
 - Tools are available for each level of the testing process
- In analogy to
CASE – Tools (Computer Aided Software Engineering),
all testing tools are sometimes referred to as
CAST-Tools (Computer Aided Software Testing)

VI- Test tools

01 – Types of test tools

Classifications of test tools /1

- Tools used for special tasks vs. test tool suites
 - Single tools support one particular test task or activity.
 - Tool suites cover several tasks and integrated several single tools
- Intrusive test tools vs. test tools that do not alter the test object
 - Intrusive tools may interfere in the execution of the test object and may cause it to differ from the true environment
 - Debuggers introduce breakpoints and alter the interrupt handling
 - Test drivers provide objects artificial (input) data
 - Coverage is determined by counters that are added to the code
 - This is not always desired
 - During performance testing, the test object must work as close to the real environment as possible
 - During system testing , test object must be embedded in a real time environment



VI- Test tools

01 – Types of test tools

Classifications of test tools /2

- Test tools for particular implementations
 - E.g. depending on the type of application like web applications
 - E.g. for particular development platform like java
 - E.g. for non-functional tests like security tests
- In house development tool
 - E.g. excel spreadsheets
 - E.g. SQL script
 - E.g. database for handling data
 - E.g. specific test result comparisons tools

VI- Test tools

01 – Types of test tools

Test management and test planning tools /1

► Test management tools

- Collecting, categorizing and administration of test cases
- Evolution / set of metrics describing the test case
- Time and resource planning, budget planning
- Creating test progress report, evaluating tests, documenting tests
- Interfacing to test execution tools, defect tracking tools and requirements management tools
- (release management / configuration management)

► Requirement management tools

- Gathering requirements on the system
- Prioritizing requirements
- Referencing requirements to test cases
- Consistency checks / evaluations (e.g. of degrees of coverage)

VI- Test tools

01 – Types of test tools

Test management and test planning tools /2

- Incident management tools (defect management tools)
 - Recording and tracking defects
 - Prioritizing, categorizing and sorting defects
 - Evaluations, i. e. metrics showing the degree of progress of testing
 - Workflow for the life cycle of a defect: changes of status, responsibility
- Configuration management tool
 - Tracking the different versions of components:
requirements met by a particular version, operating environment, compiler in use, etc.
 - Source code and object code administration
 - Reference to test management / requirement management / change management



VI- Test tools

01 – Types of test tools

Tools for static tests: reviews

- Tools for reviews
 - Supporting the review process (workflow)
 - Documenting review results
 - Evaluating review result s
 - Providing check lists for reviews
 - Supporting online reviews
 - Providing traceability between documents and source codes

VI- Test tools

01 – Types of test tools

Tools for static tests: static analysis

- Tools for static analysis
 - Compliance with coding styles / conventions
 - Analysis of code structure
 - control flow analysis, unreachable (dead) code, metrics for complexity of code (e.g. cyclamate number)
 - data flow anomalies
 - Link check of HTML or XML Code
- Model based test suite analyzing tools
 - Analysis of data models / consistency check
 - Analysis of specifications documents / object design models / state diagrams
- Prerequisites
 - The specifications are provided as a formal language documents
 - Close integration into the software development process, thus mostly seen as a development tool



VI- Test tools

01 – Types of test tools

Tools for test specification /1

- Test data generators create data to be used for tests
- Tools derive data from formal descriptions or from the structure definition
 - will not replace human effort because they lack creativity, intuition and knowledge of the test object
 - data generated automatically will often need to be reworked manually
- Tools are classified depending on the source of the data
 - Data base design
 - Source code
 - Interface specification
 - Object specifications



VI- Test tools

01 – Types of test tools

Tools for test specification /2

- Database related test data generators
 - They generate data from database or from flat files they derive data from the recognition of structures and contents
- Code based test data generators
 - They generate test data from the source code
 - They are not able to provide expected result values
 - Similar to white-box methods, they can only generate test data on the basis of the provided code
 - They cannot identify functionality that is missing

VI- Test tools

01 – Types of test tools

Tools for test specification /3

- Interface related test data generators
 - Generate data according to the interface parameters
 - Derive equivalence classes and boundary values directly for the defined parameter ranges
 - Cannot provide expected result values but may well be used for testing robustness
- Specification based test data generators
 - Generate test data directly from the specification documents
 - Specifications documents must use a strict formal notation
 - Documents produced with the aid of CASE tools may provide a good base for these tools



VI- Test tools

01 – Types of test tools

Tools for test execution

- For all test levels, tools may be introduced to support test execution
- Test execution tools may cover the following:
 - Delivering data
 - Receiving data or writing logs of output behavior
 - Documenting test execution
- Example of test execution tools:
 - **Debugger**
 - **Comparators**
 - **Test drivers**
 - **Simulator/ frameworks**
 - **Test robots**



VI- Test tools

01 – Types of test tools

Tools for test execution – Debugger

- Tool for finding errors in a program
- The sequence of program execution can be interrupted
- Single statements and conditions can be checked
- Variables can be defined individually and referenced

Comparator / comparison tools

- They compare expected and actual result based on files or databases of different formats
- Relevant data to be compared are selected using filter functionalities



VI- Test tools

01 – Types of test tools

Tools for test execution – Drivers and stubs

➤ Test drivers

- enable access to the test object, when interfaces have not been implemented
- regulate data input, data output and log the test progress
- record actual results
- May be either **standard products** or programs developed for the **particular** test object
- Often provide their own system environment

➤ Stubs

- Simulate functionality of an evoked **component**



VI- Test tools

01 – Types of test tools

Tools for test execution – Simulators / framework

- They are a replica of the **productive environment** (or a part thereof) and are needed, when **security considerations** prevent the use of the target productive environment
- The representation of the productive environment should be **as close as possible**
- They are mostly used at the level of **systems test** and **integration tests** – also possible used on component tests
- Functions simulating (part of) the system are also called **framework** or **test harness**
- If they focus on a component test they may be called **unit test framework**



VI- Test tools

01 – Types of test tools

Tools for test execution – Test robots

- May address external interfaces of the test object directly
- May accept and/or supply data, the test progress runs automatically
- Often provide a functions to **compare** actual with expected result
- Often **capture / replay** tools are used as test robots, they record test execution steps via the user interface and save as a **script file**
- Allow for **automatic repetition** of the test sequence, using the recorded script
- Well suited for **regression testing** and exploratory testing



VI- Test tools

01 – Types of test tools

Tools for test analysis and test object analysis /1

- ▶ They support or automate test analysis tasks
- ▶ They are named according to their use
 - ▶ Tools for dynamic analysis
 - ▶ Tools for the analysis of coverage
 - ▶ Tools for static analysis

VI- Test tools

01 – Types of test tools

Tools for test analysis and test object analysis /2

- Analysis of coverage (white box testing)
 - Counters are implemented that will record every access
 - After completing tests, the counters will be used to evaluate coverage (e.g. statement, branch coverage)
- Dynamic analysis (test object = component)
 - May support dynamic tests
 - Control and log the internal state of the test object, e.g. memory use
- Test monitors (test object = system)
 - Test monitor continually analyze, verify and document the use or system resources
 - They “observe” the behavior of the test object in the system environment and detect problematic situation



VI- Test tools

01 – Types of test tools

Tools for non-functional tests

► Load and performance test tools

- Monitoring the **real time** behavior of the test object in different situations
- Tools generate and execute a parameter driven **repetition** of test cases
- Deployment in complex environments requires **expert know-how** to ensure that the results are close to true conditions, e.g. network interaction





VI- Test tools

02 – Effective use of test tools

Using test tools

- **The use of test tools causes cost and efforts**
 - **providing** the appropriate tool
 - developing the required **tool skills**
 - **installing** the tool in the system environment
 - possible **adjusting** the tool or setting parameters
 - ensuring system operations **administration** efforts
 - **changeover** time preparing different tests
 - time and effort of **operating** the tool
- **The advantages of using a tool must outweigh these costs**
 - a **cost/ benefits analysis** for a tool deployment must be done in advance
 - in some cases, the total benefit will only show for tool use in more than one / in all projects

VI- Test tools

02 – Effective use of test tools

Advantage of test tools in general

- Reducing total **human** resources needed
 - allocating activities to the test tools
 - supporting and speeding up manual tasks
- Increasing the **quality** of **test execution**
 - **Iterating** identical activities
 - **Automatic evaluations** provide objective measurements
- Higher potential for test controlling
 - Managing data with test tools enables a **diversity of evaluation**
 - In this way, providing better information base to the management for **decision making**



VI- Test tools

02 – Effective use of test tools

Risks of test tools in general

- Quality **deviations** of the tool deployed
 - **Functionality** if the test tool does not meet expectations
 - **Usability** of the test tool does not meet expectations
 - **Other quality requirements** are not met
- Wrong **estimation** of benefits and not costs
 - **Benefit** was overestimated
 - **Costs** of purchase, introduction or operation were underestimated
- Wrong **deployment** of the tool
 - e. g. missing knowledge of the test methods behind
 - e.g. omitted the task of reflecting about different procedures / test processes and their suitability for the particular project
 - “A fool with a tool is still a fool”



VI- Test tools

02 – Effective use of test tools

Benefits and risks of test automation tools /1

- Test automation tools / test robots
 - The mere **recording** of test sequences usually not enough, test script must written / altered / changed / **reworked**
 - Developers skills / knowledge on scripting are always needed when deploying test robots
 - The **expected results** of the test have to be delivered for automated evaluations and comparison, otherwise potential might be wasted

VI- Test tools

02 – Effective use of test tools

Benefits and risks of test automation tools /2

➤ Data-driven approach

- **Script run program functions** of the test object. The script looks for **data** on an **external file** / spreadsheet / database
- Tester wishing to execute **new** or changed **test cases**, do not need to write new scripts but rather adapt the **external file**
- Changes in data or on the GUI might alter the reaction of the test object, **processing problems** might occur

➤ Keyword- driven approach*

- Script are modularized down **to atomic user interactions** with the test object. Extremely flexible test sequences can be created without editing the scripts
- **Test data** and invoked **functions** are saved externally. A **control script** evaluates these and invokes the particular functions with their data
- problem: the necessary external data will grow fast in **complexity**

*also known as “action word approach”



VI- Test tools

02 – Effective use of test tools

Benefits and risks of performance test tools

- ▶ Performance test tools are used mostly on applications, which are **distributed** and which communicate via **networks**
- ▶ In most cases, the **test environment cannot** be completely **isolated** and is subject to the influence of factors that are not known in detail at the time of preparing and executing test
- ▶ The complexity of the environment may make it **impossible** to **repeat identical tests** (result are hardly comparable)
- ▶ In many cases, detailed **expert knowledge** is needed to analyze the tool output correctly and to draw the right conclusions

VI- Test tools

02 – Effective use of test tools

Benefits and risks of other test tools /examples

- Static analysis tools
 - They examine the **source code** to check compliance with conventions, e.g. **programming rules**
 - Its often necessary to **prepare** the code for static analysis
 - A problem often encountered: a relatively **large amount** of indications, it is difficult to identify their **relevance**
- Test management tools
 - Information must be kept openly accessible
 - A **spreadsheet** is the tool most commonly used by the test manager for evaluations and reports
 - The reports and evaluations should adapt to the organization, not the other way around!

VI- Test tools

03 – Introduction of test tools

Introducing a new tool in organizations:

a demanding process that needs to be controlled/managed

➤ **Steps towards tool introduction**

➤ **Requirements definition:**

Demands on the tool are to be clearly define, weighted and linked to measurable criteria

➤ **Market research:**

List all possible candidates with their key properties

➤ **Tool demonstration:**

Invite the vendors for a comprehensive demonstration

➤ **Evaluation:**

Examine tool on short list. Test compliance with requested functionality. Assess further quality criteria incl. old copies, vendors support, etc

➤ **Review result and final selection:**

Review evaluation result and make final decision

Support introduction by coaching and training for tool usage. Ideally, set up pilot project to introduce the tool



VI- Test tools

03 – Introduction of test tools

Advantages of a pilot project for tool introduction

- Getting to know the tool in detail with **strong and weak points**
- **Interfacing** to **other tools** in use, adapting process and workflows
- Defining reports according to the standards of the organizations
- Assess if tool meets the **expected benefits**
- Estimate whether the cost of deployment is within scope
- No roll-out without **piloting**: otherwise expect acceptance problems



VI- Test tools

03 – Introduction of test tools

Success factors of software deployment

- **Step-by-step** introduction and rollout in the **complete organization**, not only in one project
- Make tool usage **mandatory** for the respective workflow / process
- **User guidelines** are necessary for tool deployment
- User must have access to adequate **training**, quick support must be available for user
- **Experience** gained from tool deployment should be made available for **all user**
- The **actual use** of the tool should be followed up, so that any necessary interventions can be made to **improve its acceptance**

VI- Test tools

03 – Introduction of test tools

Summary

- There is a broad range of **test tools** available, covering many **different tasks**
 - Test management tools
 - Test planning tools
 - Test specification tools
 - Test execution tools
 - Tools for test object analysis
 - Tools supporting non-function test
- Tool deployment should be carried out based on a **cost-benefit analysis**
- The **introduction** of a new test tool must be **prepared carefully** in order to be **successful**
- A step-by-step rollout starting with a **pilot project** is recommended.



Thank You