Chapter 7: Test Tools and Automation

Objectives

- At the end of this chapter the student be able to know about the following test tools and automation.
  - To define the over view of test tools and automation
  - To know about Testing Tool Acquisition
  - To define Testing Tool Introduction and Deployment
  - To Classification of Testing Tools

Introduction to Test Tools and Automation

- The purpose of using tools for testing is to get as many as possible of the noncreative, repetitive, and boring parts of the test activities automated.
- The purpose is also to exploit the possibility of tools for storing and arranging large amounts of data.
- There are a huge number of testing tools on the market, and it is growing fast. Example: jtest , JUnit , GUIdancer Jcover etc. See more testing tools , have a look in Appendix 9A!
- Every testing tool automates some testing activities to a certain degree.
- No single tool automates everything completely.

Introduction to Test Tools and Automation Cont.

- Test automation is not an easy task.
- A company can be more or less ready for test automation.
- It requires a certain level of maturity to be able to use tools efficiently.
- It is important to select tools with great care so that they don’t end up as “shelfware.”

Testing Tool Acquisition

- In a professional organization it is important to treat the investment in (testing) tools as the serious decision it is.
- Tools are usually expensive, and even if they are not expensive to buy, they are expensive to implement and maintain in the organization.
- Acquisition and introduction of a tool in a company requires organizational considerations.
  - It is not something you just rush in and do (like fools!);
  - conscious decisions about what to do and how to do it in the company must be made before the work can commence.
- The acquisition should include the following activities:
  - Tool selection preparation
  - Tool evaluation
  - Selection of the winner

Testing Tool Acquisition Cont.

- Tool or No Tool?
- Tool Selection Team
- Testing Tool Strategy
- Preparation of a Business Case
- Identification of Tool Requirements
- Buy, Open-Source, or Do-It-Yourself
- Preparation of a Shortlist of Candidates
- Detailed Evaluation
- Performance of Competitive Trials

Testing Tool Acquisition Cont.

❖ Tool or No Tool?

- The first thing we must do when the idea of automation occurs is find out what it is we are trying to achieve with the tool.
- Introduction of a testing tool or testing automation is not necessarily the answer to all problems.
- Only work that is well-specified is appropriate for automation.
- Work that requires creativity is not a candidate for automation either.
- We cannot get a computer to be creative and think outside of the box.
- Automation may help solve problems caused by:
  - ➢ Work that is to be repeated many times
  - ➢ Work that it is slower to do manually
  - ➢ Work that it is safer to do with a tool

Testing Tool Acquisition Cont.

- Once the problem is described and well understood, we can consider how to solve it.
- There may be a number of alternative solutions, including the acquisition of a tool.
- Maybe it does seem like a tool is the best solution, and in that case we can go on with the selection preparation.

❖ Tool Selection Team

- The next step is to establish a team to perform the evaluation and selection of the tool.
- The team must be composed of a team leader and representatives for all potential users of the test tool.

Testing Tool Acquisition Cont.

- Including
  - developers,
  - professional testers,
  - responsible for tools,
  - responsible for process, and
  - future product users.

Testing Tool Strategy

- The point to make clear is how a new tool will fit into the overall goals for the company .e.g.
  - with regard to general process improvement or
  - the achievement of a certain level of capability or
  - a specific certification.
- This may have an impact on the type of tool to choose.

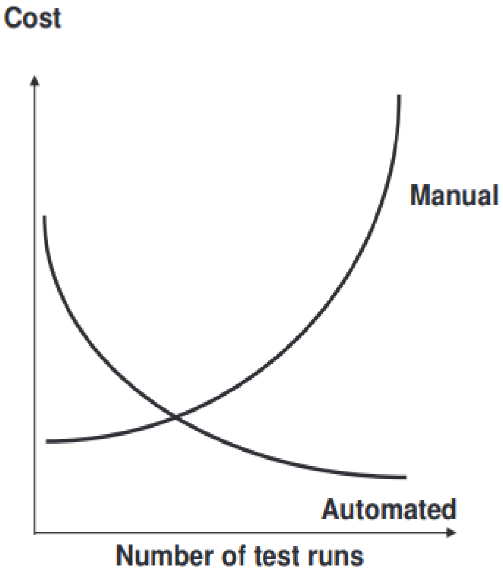
Testing Tool Acquisition Cont.

❖ Preparation of a Business Case

- In a business case we compare the cost of a solution with the benefits the solution is going to bring us.
- The cost of selecting, implementing, and maintaining a tool is usually significant.
- It includes expenses for:
  - ➢ Selection
  - ➢ Acquisition
  - ➢ (list price minus possible discounts, open source, or own development)
  - ➢ Licenses
  - ➢ Tailoring
  - ➢ Implementation
  - ➢ Training
  - ➢ Tool usage
  - ➢ Maintenance of automated testware
  - ➢ Tool maintenance

Testing Tool Acquisition Cont.

- Some of these expenses are measured in money , time being spent by employees must be considered in the calculation.
- On the other side of the business case equation, we have the benefits.
- For test execution tools the cost/benefit depends very heavily on how often the automated tests will be executed.



- Tests that are only executed a few times during the entire lifetime of the product are usually not worth spending automation resources on.
- On the other hand, it may be well worth automating the tests that are executed many times, for example, tests used for extensive regression testing of high-risk areas.
- It is of course possible to have a mixture of manual and automated tests.

Testing Tool Acquisition Cont.

Identification of Tool Requirements

- It is part of the selection preparation to identity the requirement applicable for the tool to be implemented.
- First of all the tool needs to have some *functionality*.
  - We must define what we require the tool to do.
  - The integration with other tools is another important part of the functionality we need to specify.
- Connected to the functional requirements we have the nonfunctional requirements.
  - These should at least include aspects of performance, usability, availability, and maintainability for the tool.

Testing Tool Acquisition Cont.

- The *environmental requirements* or constraints are requirements forced on us from the environment around our organization.
  - specific platform that we need to use.
- The last thing to consider is *project requirements* or *constraints*. Namely
  - resources, time, and money.

Buy, Open-Source, or Do-It-Yourself

- There are advantages as well as disadvantages, to buying a standard tool, getting an open-source tool, or developing one’s own tool.
- If a company decides to develop its own tool, this must be undertaken like any other development project (i.e., at least as seriously as a project with an external customer).

Testing Tool Acquisition Cont.

❖ Preparation of a Shortlist of Candidates

- Many sources for information about testing tools exist, for example, articles, suppliers' Web pages, other companies, exhibitions, and research reports.
- It can be useful to supplement the evaluation with a look at the supplier.
- The supplier is "the family-in-law" that we will have to live with for a longtime, so investigate for example:
  - ➢ The supplier's employees—Do they match ours?
  - ➢ The supplier's own use of the tool
  - ➢ The supplier's financial status
  - ➢ The supplier's focus—Is testing tools a niche?
  - ➢ The supplier's acquaintances
  - ➢ The supplier's reputation
  - ➢ The supplier's support facilities

Testing Tool Acquisition Cont.

❖ Detailed Evaluation

- After this first selection a stricter and stricter evaluation is made until only two candidates are left in the field.
- It is important that the evaluation group agrees on
  - ➢ how the evaluation is to be made and
  - ➢ precisely what is significant in the selection.
- An evaluation method includes:
  - ➢ Description of the scale for the evaluation of fulfillment of the requirements, for example
    - ✓ Fully, Almost, Partly, Not
    - ✓ From 0 to 100%
  - ➢ Description of the selection criteria, based on the fulfillment evaluation, for example
  - ➢ All priority 1 requirements fulfilled at least 80% and at least 50% of the priority 2 requirements fulfilled

Testing Tool Acquisition Cont.

❖ Performance of Competitive Trials

- The two finalists should undergo a detailed evaluation that should include at least one demonstration and preferably a trial period, so that the tools may be tried out under as realistic circumstances as possible.
  - ➢ functional and nonfunctional requirement can be tested.
- It may be important to investigate if a tool can handle the volumes that the company or the project may have to handle.
  - ➢ Volume may also be a question of a large number of users and/or a large number of platforms possibly distributed over large distances.

Testing Tool Introduction and Deployment

- The introduction or implementation of a tool in an organization is an organizational change project.
- Management commitment is essential for the implementation to be a success.
- An implementation process should include the following activities:
  - Make necessary adjustments
  - Perform a pilot project
  - Assess the pilot project
  - Produce a rollout strategy
  - Make the rollout happen
  - Follow up on the rollout

Testing Tool Introduction and Deployment Cont.

- The necessary resources, both in terms of people, time, money, and training must be provided and sustained until the usage of the new tool is an engraved part of everyday working life.
- The roles that must be in place to make the tool implementation a success are:
  - The sponsor
  - The target group
  - The champions
  - The change agents
- The introduction of a tool requires *a tool custodian*.
  - This is a technical person who is responsible for the setup and maintenance of the tool.
  - He or she provides internal help and support with technical issues and can be responsible for contact to the supplier of the tool for second-level support.

Testing Tool Introduction and Deployment Cont.

❖ Testing Tool Piloting

- A pilot project should always be performed for the tool before we commit to implementing it across all projects.
- There are a number of reasons for performing a small-scale pilot project.
  - ➢ to verify the business case and ensure that the benefits of the usage of the testing tool can really be achieved.
  - ➢ to get some experience in the usage of the testing tool.
  - ➢ to identify further adjustment need to the processes and to the tool, as appropriate.

Testing Tool Introduction and Deployment Cont.

- They also require interfaces with other tools and other processes, for example, configuration management of testware.
- Finally a pilot can help us refine the estimate for the actual costs and benefits for the implementation.

- A pilot should take between three and six months and be followed closely.

Testing Tool Rollout

- based on a successful evaluation of the pilot project.
- Rollout normally requires a great involvement of all the people carrying roles in the test tool implementation, not least the users of the testing tool, the target group.
- A rollout strategy that suits the nature of the organization must be defined.

Testing Tool Introduction and Deployment Cont.

- A “big-bang” rollout, where everybody starts using the tool at a given point in time, works in some organizations.
- We must be prepared to
  - Support the users
  - Support the users
  - Support the users
  - Support the users
- until the usage of the testing tool is a completely integrated part of the work.

❖ Testing Tool Deployment

- A testing tool is a part of the test environment for our tests.
- The tools we use should be kept under proper configuration management like the rest of the test environment and other testware.

Testing Tool Introduction and Deployment Cont.

- It is important to be able to register with which version of a tool specific tests have been prepared and/or executed.

Testing Tool Categories

Testing Tool Classification

- Many tools for the support of software development are available, and it is growing every day.
  - It is therefore impossible to list specific tools.
- The purpose of this section is
  - to present different types of testing tools and
  - the advantages and disadvantages of testing tools .
- There are different classification schemes, for example according to:
  - The test activity they support
  - The test level the tools primarily support
  - The types of failures or defects they can find
  - The test approach or test technique they support
  - The purpose they have
  - The domain to which they are applied
  - Who the primary users of the tools are

Testing Tool Categories Cont.

- Tool support exists for the following primary users:
  - ➢ All testers
    - ✓ Test management tools, including configuration management tools
  - ➢ Test analysts and technical test analysts
    - ✓ Test design tools
    - ✓ Test data generation tools
    - ✓ Test oracles
    - ✓ Simulation and emulation tools
    - ✓ Test execution tools
    - ✓ Keyword-driven automation tools
    - ✓ Comparison tools
    - ✓ Fault-seeding and fault-injection tools
    - ✓ Web tools

Testing Tool Categories Cont.

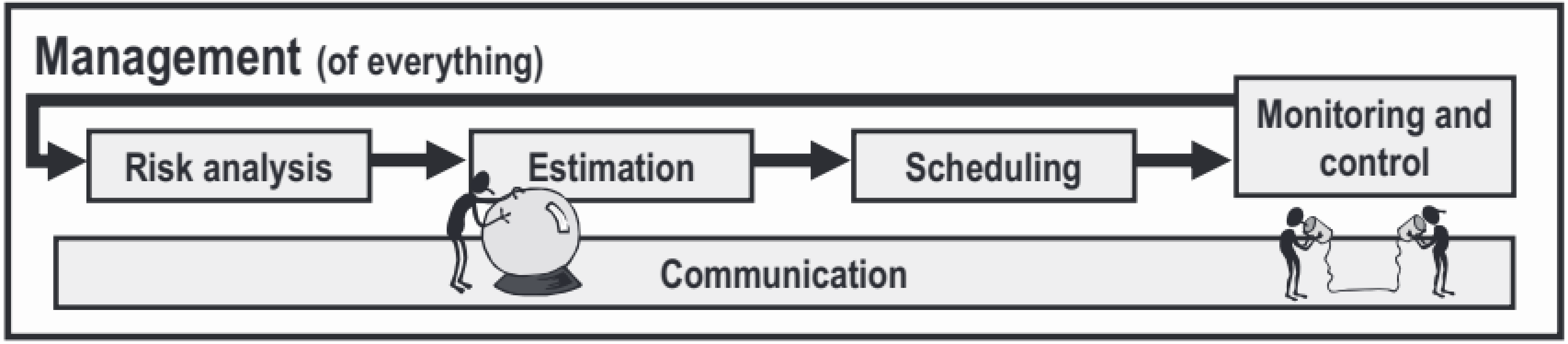
- Technical test analysts only
  - ✓ Static analysis tools
  - ✓ Dynamic analysis tools
  - ✓ Performance testing tools
- Programmers (or technical test analysts writing and maintaining test scripts)
  - ✓ Debugging, tracing, and troubleshooting tools

Testing Tool Categories Cont.

❖ Tools for All Testers

Test Management Tools

- Test management includes risk analysis, estimation, scheduling, monitoring and control, and communication.



- Test management tools cover these activities and support the **project management** aspects of testing.
- These tools can be used for registration of test activities, estimation, scheduling of tests, logging of results, and analysis and reporting of progress.

Testing Tool Categories Cont.

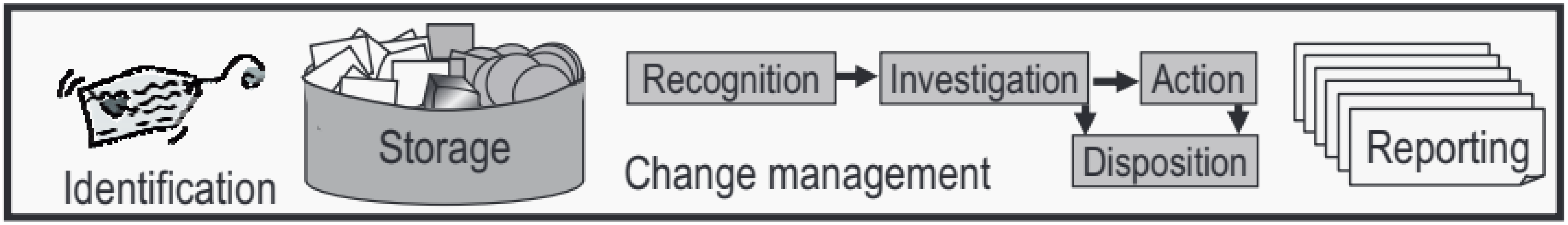
- Test management tools can support the handling of test documentation, such as plans, test specifications, test procedures, and even traces between test cases and requirements.
- **Advantage**
  - They can assist in the management of all the activities in testing.
  - They can provide an overview of the testing task and show progress.
- **Disadvantage**
  - There are no direct disadvantages of test management tools.
  - They are often wedged between other tools, such as project management tools and configuration management tools.
  - The confusion between tools borderlines between project management, configuration management, and test management are often blurred or not defined.

Testing Tool Categories Cont.

- This has got to do with the maturity of the organization.

❖ Tool Support for Configuration Management

- Configuration management is identification, storage, change management, and reporting of configuration items.



- Configuration items are all work products, product components, and components that we want to control.
- This includes testware, such as
  - test plans,
  - test specifications, and
  - test environments including tools and test results ; and
  - it includes requirements.

Testing Tool Categories Cont.

- Configuration management tools are used to support the configuration management activities.
- The main features of these tools are:
  - Identification and storage of items
  - Traceability between items
  - Incident reporting and management of the life cycle of faults
  - Reporting and analysis
- Requirements and test cases should be traced to each other.
  - traceability tools allow the link between test cases and their corresponding test coverage items to be recorded.
- Changes to configuration items should always be initiated by an incident report.
- Incident management tools (also known as defect tracking tools)
  - to track and control the allocation, correction, and retesting of incidents.

Testing Tool Categories Cont.

❖ Tools for Test Analysts and Technical Test Analysts

Test Design Tools

- Test design tools support the creation of test specifications
- They can analyze a specification of the product and generate high-level test cases and possibly test procedures or scripts based on this analysis.
- This type of testing tool can, for example:
  - ➢ Derive high-level test cases from formally specified requirements, often managed by the same tool
  - ➢ Generate test cases based on the specification of a model, for example, UML or state machines
  - ➢ Generate input for test cases based on input models, for example, input distribution specifications
  - ➢ Derive high-level test cases from actual source code

Testing Tool Categories Cont.

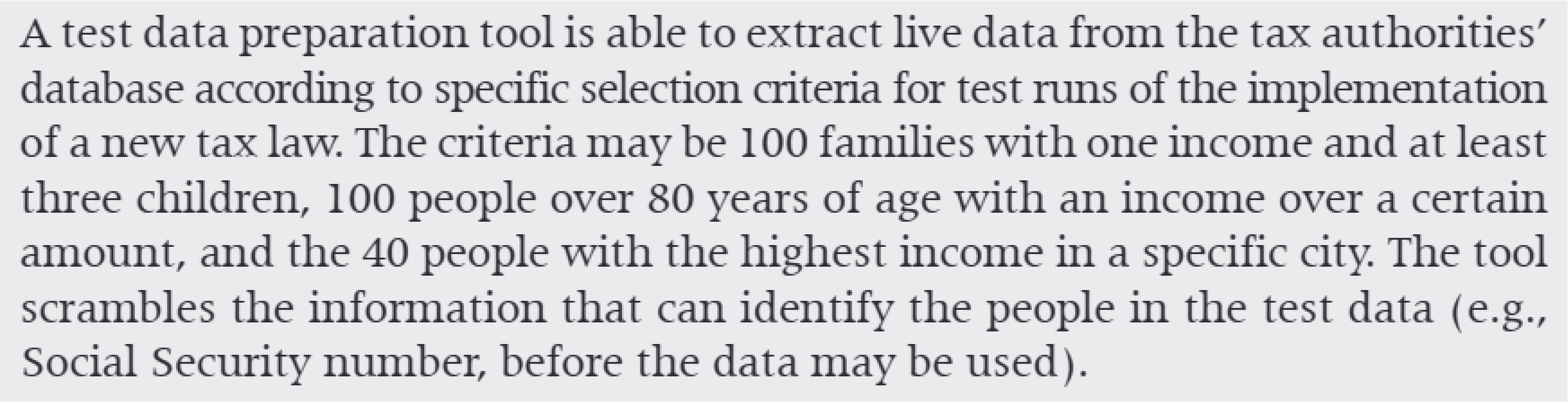
- **Advantage**
  - test cases are systematically and comprehensively derived from the basis documentation.
- **Disadvantage**
  - these testing tools only do half (or less) of the work.
    - ✓ They cannot specify the expected results.
    - ✓ The test design tools require very formally formatted basis documentation.

Test Data Preparation Tools

- Test input data preparation tools support:
  - Selection (e.g., from an existing database)
  - Creation
  - Generation
  - Manipulation
  - Editing

Testing Tool Categories Cont.

- Test data can be selected and extracted from live data and scrambled to hide person-sensitive information.
- This enables tests to be performed on real data .Example :



A test data preparation tool is able to extract live data from the tax authorities' database according to specific selection criteria for test runs of the implementation of a new tax law. The criteria may be 100 families with one income and at least three children, 100 people over 80 years of age with an income over a certain amount, and the 40 people with the highest income in a specific city. The tool scrambles the information that can identify the people in the test data (e.g., Social Security number, before the data may be used).

- **Advantage**
  - these tools make it possible to handle **great volumes** of data.
- **Disadvantage**
  - the tools may create too much useless data, if selection is not planned carefully.

Testing Tool Categories Cont.

Test Oracles

- A test oracle is a special concept in test automation.
  - it is used to determine expected results from inputs.
- Automated test oracles are tools that can generate the expected result for specific input.
  - hence facilitate the creation of test cases. Such “oracles” are hard to find.
- One of the situations where an oracle can be found and can be very useful is
  - when an old system is being replaced by a new one providing the **same functionality.**
  - where nonfunctional requirements can be disregarded and a system simulating the functionality only can be developed at a much lower cost.
- Advantage :
  - to generate the expected results
  - requires strict control over the oracle and the other testware

Testing Tool Categories Cont.

- Disadvantage

- their usage can give us a false sense of reliability.
- There is a risk that we repeat faults in the old system, or between an oracle system and the real system.
- There is also a risk of not getting sufficient test coverage.

Simulation Tools and Emulation Tools

- Simulators are used to support tests where necessary code or other systems are either unavailable or impracticable or even dangerous to use.
- Test harnesses and drivers fall into this category of tools.
- They are used where components or other test objects cannot be executed directly.
  - testing of a component in isolation, embedded software without a user interface, or execution of many unrelated automated test scripts.

Testing Tool Categories Cont.

- Some testing tools on the market provide harness and driver facilities, especially component testing tools.
- A special type of simulators is called emulators.
- **Advantage**
  - These tools can save us a lot of money.
  - Emulators can make it possible to test in “slow motion,” and they can act as debuggers as well.
  - require good configuration management of the testware
- **disadvantages**
  - These tools may give a false sense of reliability after all the simulators or emulators may be wrong.
  - “hide” defects. Example:- performance and other time-related defects
  - expensive to produce and set up compare cost with benefit.
  - they require the testers can code or have access to people who are able to do the coding

Testing Tool Categories Cont.

Test Execution Tools

- This type of testing tool has many names:
  - test execution tools, or
  - test running tools, or
  - capture and replay tools,
- The most widespread category of testing tool.
- These tools are primarily used for automation of **regression testing.**
- All the tools of this category work according to the same basic principles, namely:
  - Capture: A recording of all the tester's manual actions and the system's responses into a test script
  - Control points: A number of checkpoints added to the script by the tester during the capture
  - Playback: Automatic (re)execution of the test scrip

Testing Tool Categories Cont.

- Test execution tools exist for graphical user interface, and for character-based interfaces.
- For GUI applications the tools can simulate mouse movement and button clicks and can recognize GUI objects such as windows, fields, buttons, and other controls.
- **Advantage**
  - A lot of manual test execution can be done automatically.
  - In iterative development and other development projects where a large number of regression testing are needed.
  - These tools are indispensable in development where "frequent build and smoke test" principles are used.
  - The use of test scripts requires good configuration management to keep track of which versions of the test objects, test data, and test scripts belong together.

Testing Tool Categories Cont.

- **Disadvantage**
    - expensive to establish and maintain the test scripts.
    - The requirements, specifications, and code undergo changes in the course of the development, especially in iterative development.
    - The test scripts requires programming skills.
    - Defects are also introduced in test scripts.

- **Keyword-Driven Automation Tools**

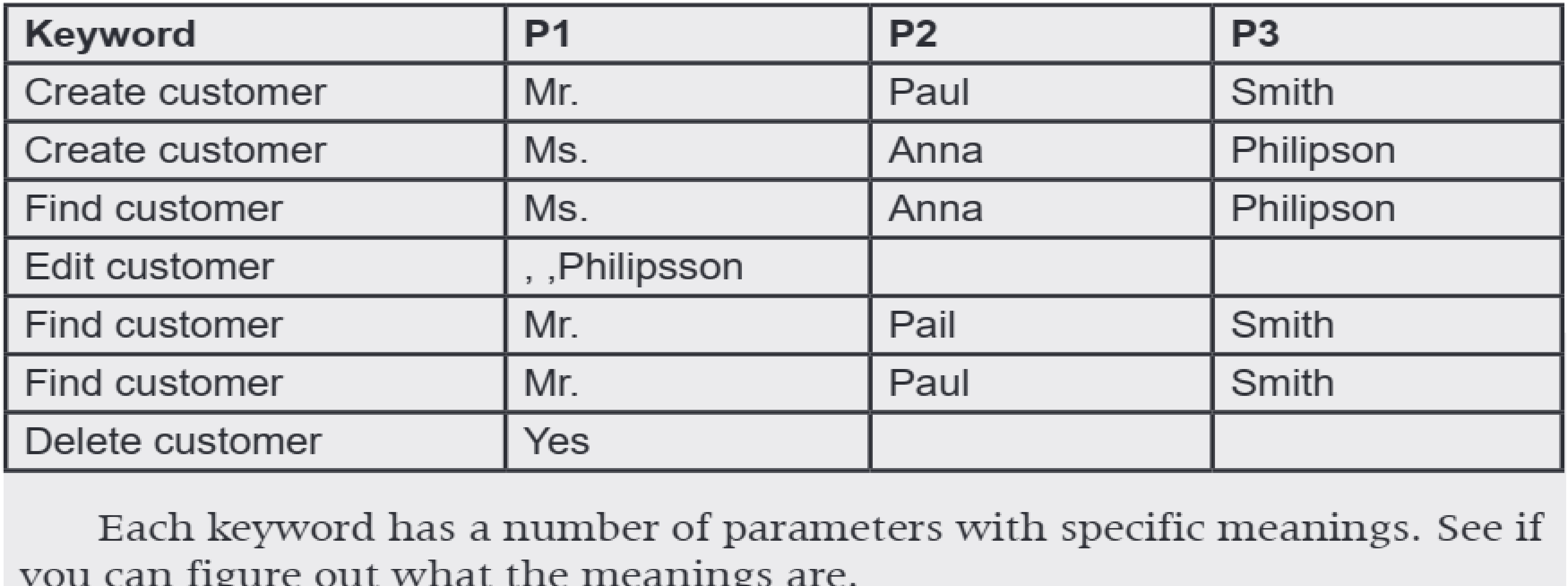
- Keyword-driven test is a way to execute test scripts at a higher level of abstraction.
    - The same code may be executed with different values.

- Keywords are defined to represent a script, and a tool can then act as **a link between** the keywords and the tool executing the corresponding test script.

- Values may be assigned for parameters associated with the keywords.

Testing Tool Categories Cont.

- This testing are also known as script wrappers
  - they wrap the technical part of the test (the actual test scripts and the test execution tool) so that the testers only need to know about the high-level keywords
- Keywords may be held in spreadsheets or tables, and longer sequences executions of test scripts can be specified by sequences of keywords.
- A test sequence defined by keywords in a table may look like this:



Keyword	P1	P2	P3
Create customer	Mr.	Paul	Smith
Create customer	Ms.	Anna	Philipson
Find customer	Ms.	Anna	Philipson
Edit customer	, ,Philipsson		
Find customer	Mr.	Pail	Smith
Find customer	Mr.	Paul	Smith
Delete customer	Yes		

Each keyword has a number of parameters with specific meanings. See if you can figure out what the meanings are.

Testing Tool Categories Cont.

- Keyword-driven test is getting more and more sophisticated,
  - ✓ introducing several levels of abstraction between the **tester** and the technical test scripts.
- Test wrapping tools are available **commercially** and as open-source, but they are also very often homemade.

- **advantages**

- it is easier to use keyword-driven testing rather than test script directly, because:
  - ✓ Keywords that reflect the business can be chosen
  - ✓ Test execution can be done automatically by nontechnical people based on the keyword lists
  - ✓ The keyword list is robust to minor changes in the software
  - ✓ The implementation of the keywords is independent of the implementation of the underlying scripts,
    - ✓ the same keyword lists may be used with scripts in a number of different scripting languages being executed in different execution tools

Testing Tool Categories Cont.

Disadvantage

- Extra layers are put in between the test executer and the product under testing.
  - ✓ It requires more coordination and communication between the people involved to maintain the integrity of the layers in the testware.
  - ✓ keyword-driven testing requires extra care in configuration management.

Comparison Tools

- Comparison tools are used to find differences between the expected and the actual results.
- The tools may be able to compare
  - values in files or on screens, bitmaps, and positions.

Testing Tool Categories Cont.

- **Advantage**
  - can compare large amounts of data very fast and without getting tired.
- **Disadvantage**
  - they may produce enormous amounts of reported data of which only a fraction is relevant.

Fault-Seeding and Fault-Injection Tools

- The tools create or inject faults (defects) into the software component under testing.
- The tools can work
  - either on the source code, changing the code in pre specified ways, or on the compiled code, changing the structure of the code.
- In both cases new versions of the component under test are created with the specified defects.

Testing Tool Categories Cont.

- **Advantage**
  - Many defects may be injected in a systematic way to support these defect based techniques.
- **Disadvantage**
  - the defects are not necessarily realistic and may not be found by the specified tests.
- **Web Tools**
- Products being Web-based means that some issues are out of our hands (for example, hyperlinks and server and network availability).
- Hyperlink testing tools are used to check that no broken hyperlinks are present on a Web site.
- These tools often have additional functionality such as HTML validation, spelling, and availability check.

Testing Tool Categories Cont.

- Monitoring tools are used for Web-based products.
  - e-commerce and e-business applications.
- The tools monitor the product's availability to customers and the service level.
  - performance and resource usage.
- The tools will issue warnings if the monitoring shows that something is not as expected.
- **Advantage**
  - they can check all hyperlinks very quickly.
  - It is important for the trustworthiness of a Web site that there are no broken links.
  - give us a chance to know if things are not working
- There is no disadvantage using these tools.

Testing Tool Categories Cont.

❖ Tools for Technical Test Analysts

Static Analysis Tools

- Static analysis can be performed on code as well as on architecture.
- Most static testing is performed by people, but some types are supported by tools.
- Static analysis tools examine the written code to detect, for example,
  - ➢ variable anomalies,
  - ➢ to check adherence to defined coding rules, and
  - ➢ to collect measurements concerning the code, for example,
    - ✓ cyclomatic complexity and
    - ✓ Web site balance.

Testing Tool Categories Cont.

- **Advantage**
  - The tools find all occurrences of the faults they are looking for.
  - Static analysis requires some coding standards to check against to find deviations
- **Disadvantage**
  - Specially some older tools may find a number of “incidents” that are not faults after all.
  - The reports for static analysis can be overwhelming with many things that can be disregarded, and that can make it difficult to find the “gold nuggets.”

Dynamic Analysis Tools

- Dynamic analysis tools are used to provide information about the behavior and state of software while it is being executed.

Testing Tool Categories Cont.

- These tools primarily give run-time information about memory handling and pointers.
  - Memory handling is concerned with allocation, usage, and deallocation of memory.
  - The tools can detect memory leaks,.
    - ✓ where memory is gradually being filled up during extended use, long before it actually happens.
  - Pointers are used to handle dynamic allocation of memory .
    - ✓ the dynamic analysis tools can identify unassigned pointers.
    - ✓ They can also detect faults in pointer arithmetic.
- **Advantage**
  - They can find faults that are almost impossible or very expensive to find in other ways.
  - They don’t need specific test cases
    - ✓ they report on what is going on while other test cases or scenarios are executed.

Testing Tool Categories Cont.

- **Disadvantage**
  - the code is instrumented by the tool in order for the tool to catch the run-time information.
    - ✓ This means that it is not strictly the “real” code we are testing.
  - It can also have an adverse impact on performance, and that can pose problems if we are testing real-time software.
- A special type of dynamic analysis tool is coverage measurement tools or analysis tools. Example
  - Statement coverage
  - Branch coverage

Performance Testing Tools

- Performance testing tools are used to:
  - Generate large volumes or loads on the product
  - Measure the performance of the product under the controlled circumstances

Testing Tool Categories Cont.

- volume testing tools may be the number of concurrent users, the amount of memory to be used, the number of information items of a given type. the number of transactions per time unit.
- The usage of the tools for stress testing is similar to the one described for volume testing.
- The tools can be used to measure what the performance is under given circumstances.
- It can provide very useful reports based on collected information, often in graphical form.
- **Advantage**
  - they can provide information about “bottleneck” areas relatively inexpensively before the product hits the real world.
  - There are *no disadvantages* of these tools

Testing Tool Categories Cont.

- **Tools for Programmers**

Debugging Tools

- Debugging tools are NOT testing tools!
- They are related to testing, since they are used by programmers to pinpoint defects.
- Debuggers allow programmers to:
  - Execute the code line by line
  - Insert break points
  - Control and set values of variables at break points.
- **Advantage**
  - They can save the programmers a lot of time during detailed fault hunting.
  - It can also be motivating for some testers with a development background to work with the programmers

Testing Tool Categories Cont.

- **Disadvantage**
  - programmers can waste a lot of time if the tools are used in an undisciplined way or to play with.

Thank You !!!