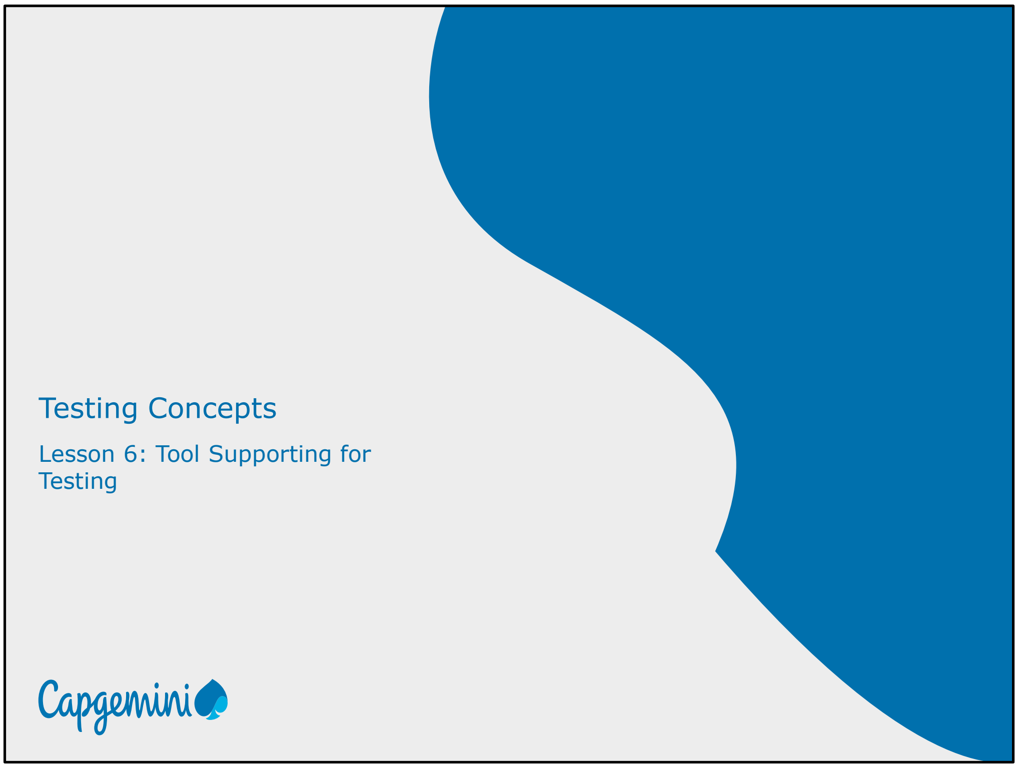
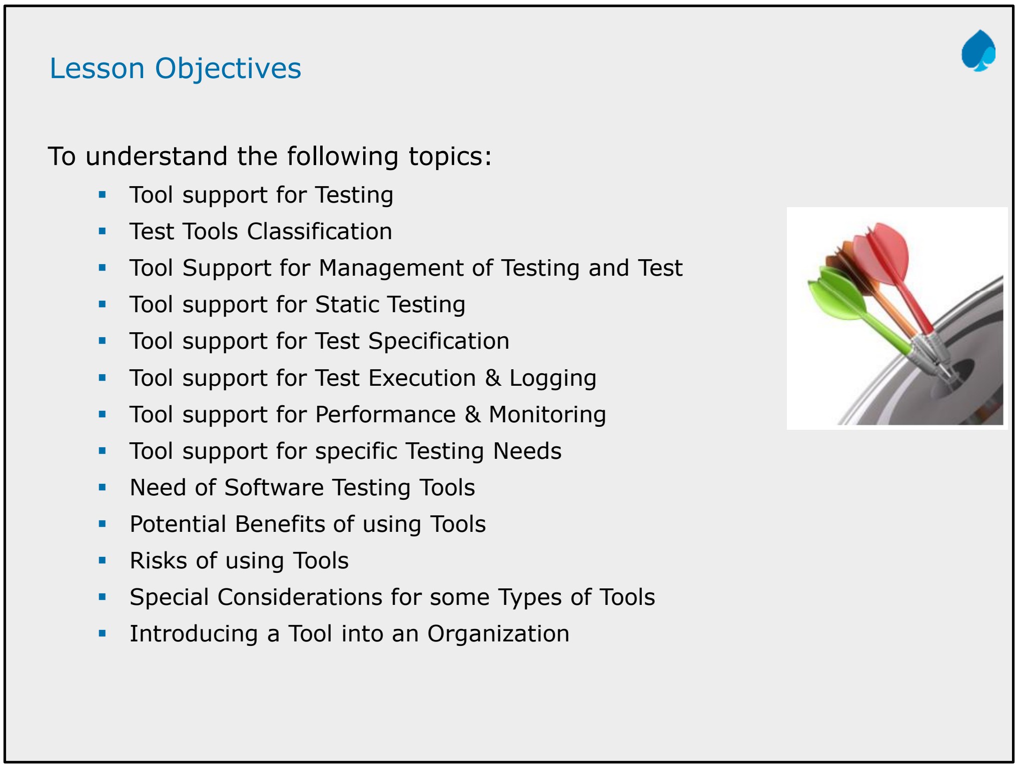


Testing Concepts

Lesson 6: Tool Supporting for Testing

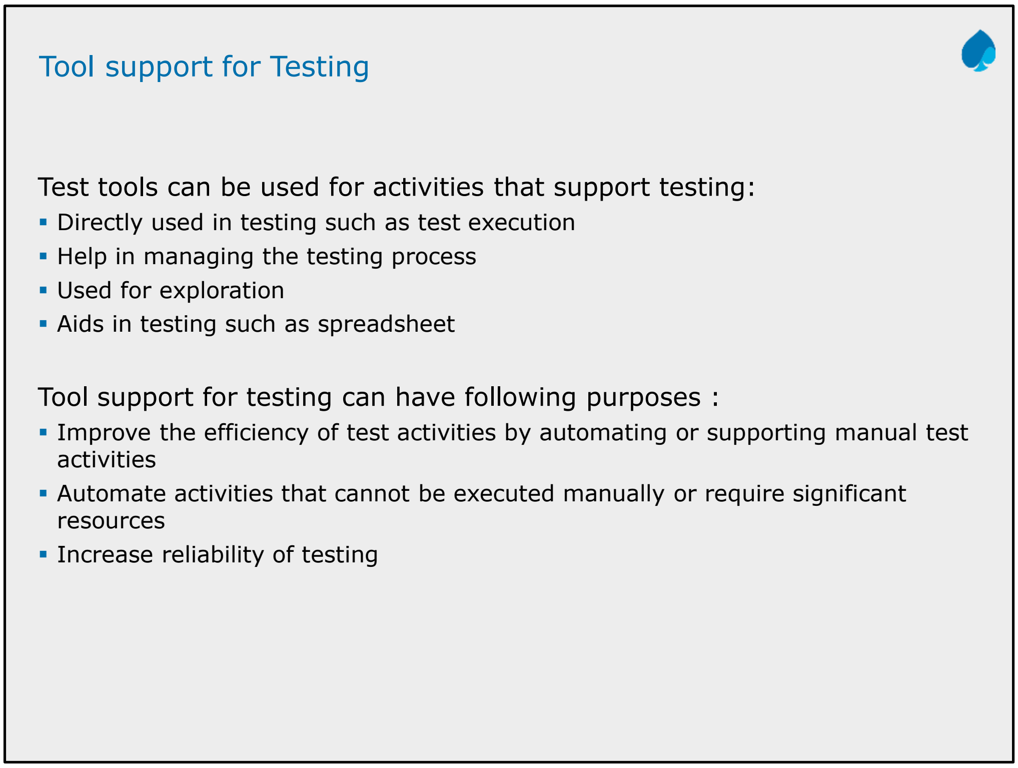
Capgemini



Lesson Objectives

To understand the following topics:

- Tool support for Testing
- Test Tools Classification
- Tool Support for Management of Testing and Test
- Tool support for Static Testing
- Tool support for Test Specification
- Tool support for Test Execution & Logging
- Tool support for Performance & Monitoring
- Tool support for specific Testing Needs
- Need of Software Testing Tools
- Potential Benefits of using Tools
- Risks of using Tools
- Special Considerations for some Types of Tools
- Introducing a Tool into an Organization



Tool support for Testing

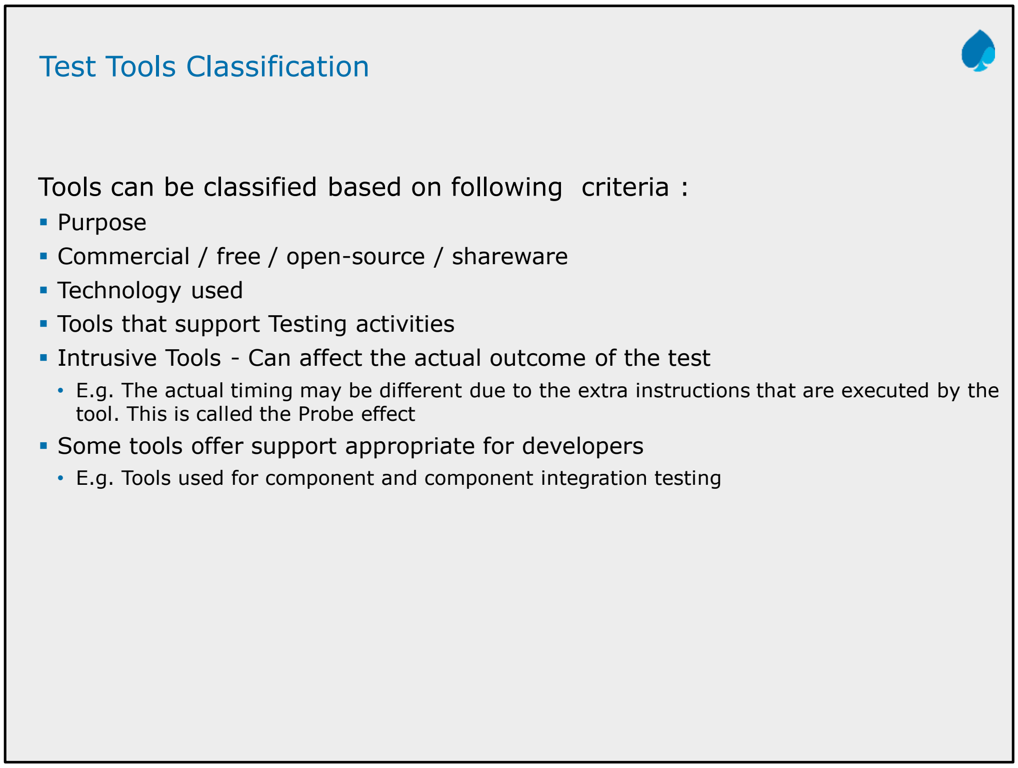
Test tools can be used for activities that support testing:

- Directly used in testing such as test execution
- Help in managing the testing process
- Used for exploration
- Aids in testing such as spreadsheet

Tool support for testing can have following purposes :

- Improve the efficiency of test activities by automating or supporting manual test activities
- Automate activities that cannot be executed manually or require significant resources
- Increase reliability of testing

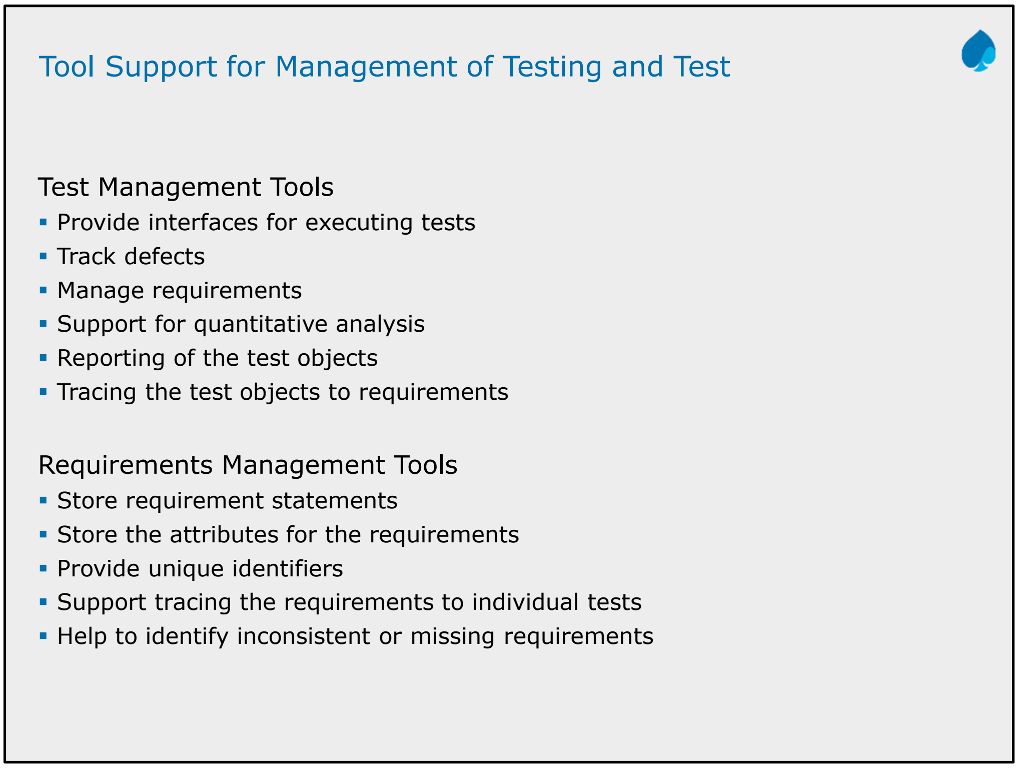
Tool support for Testing
Tools used for test activities in software life cycle are called Computer Aided Software Testing or CAST Tools.



Test Tools Classification

Tools can be classified based on following criteria :

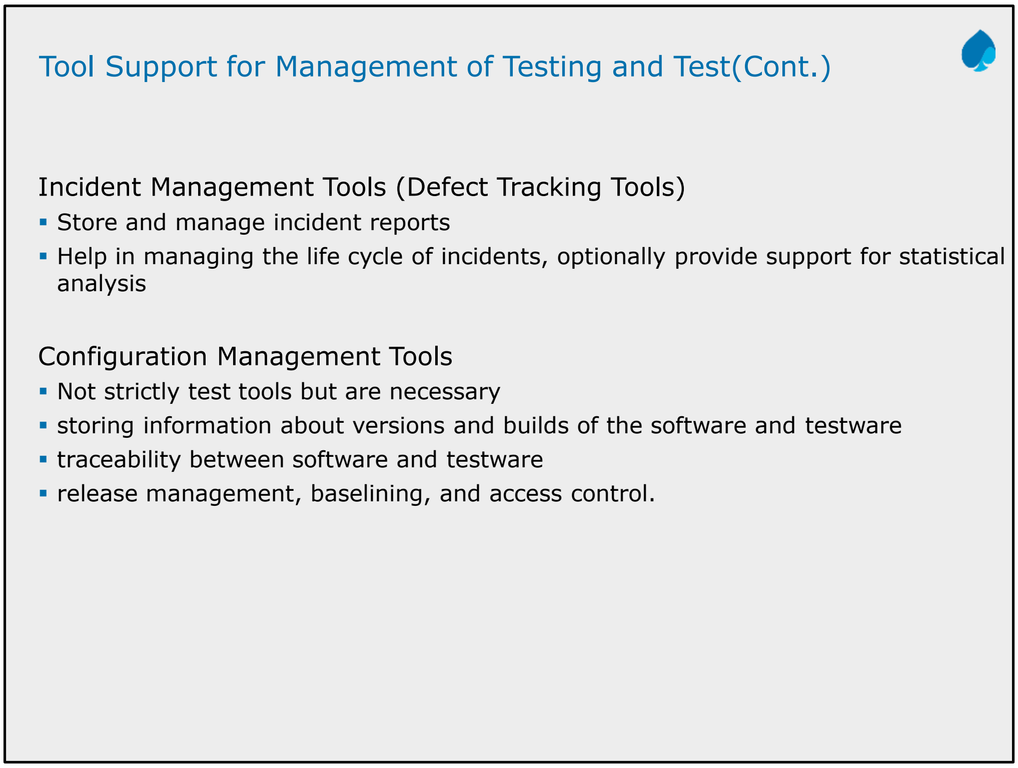
- Purpose
- Commercial / free / open-source / shareware
- Technology used
- Tools that support Testing activities
- Intrusive Tools - Can affect the actual outcome of the test
  - E.g. The actual timing may be different due to the extra instructions that are executed by the tool. This is called the Probe effect
- Some tools offer support appropriate for developers
  - E.g. Tools used for component and component integration testing



Tool Support for Management of Testing and Test

Test Management Tools
- Provide interfaces for executing tests
- Track defects
- Manage requirements
- Support for quantitative analysis
- Reporting of the test objects
- Tracing the test objects to requirements

Requirements Management Tools
- Store requirement statements
- Store the attributes for the requirements
- Provide unique identifiers
- Support tracing the requirements to individual tests
- Help to identify inconsistent or missing requirements



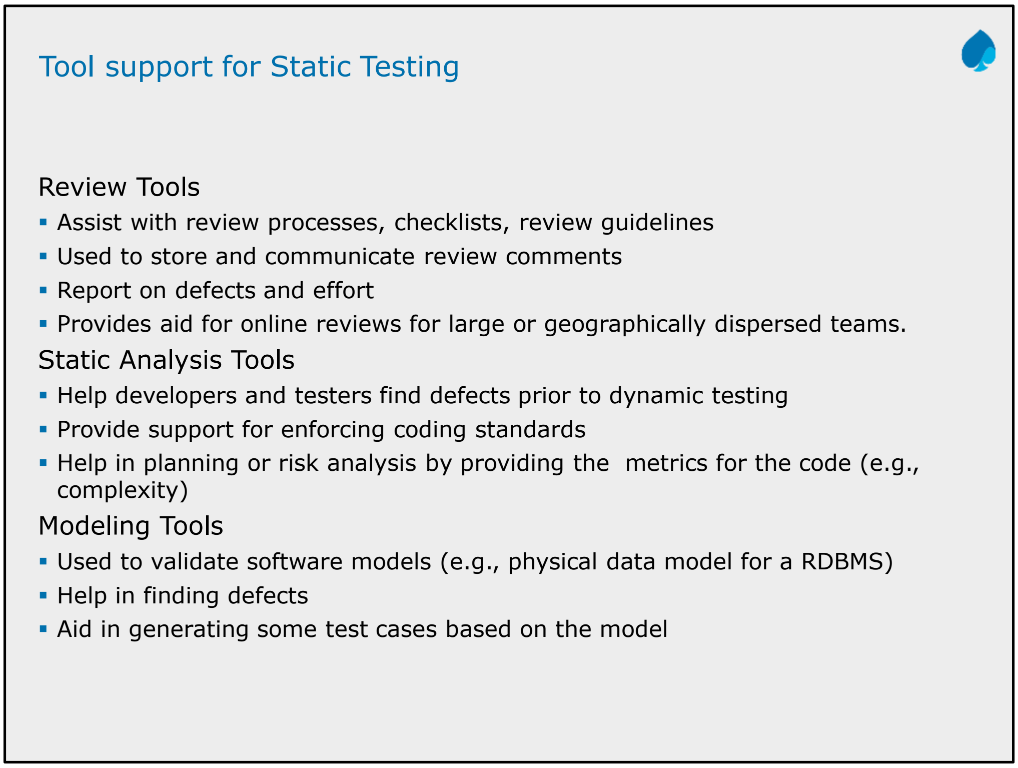
Tool Support for Management of Testing and Test(Cont.)

Incident Management Tools (Defect Tracking Tools)

- Store and manage incident reports
- Help in managing the life cycle of incidents, optionally provide support for statistical analysis

Configuration Management Tools

- Not strictly test tools but are necessary
- storing information about versions and builds of the software and testware
- traceability between software and testware
- release management, baselining, and access control.



Tool support for Static Testing

Review Tools

- Assist with review processes, checklists, review guidelines
- Used to store and communicate review comments
- Report on defects and effort
- Provides aid for online reviews for large or geographically dispersed teams.

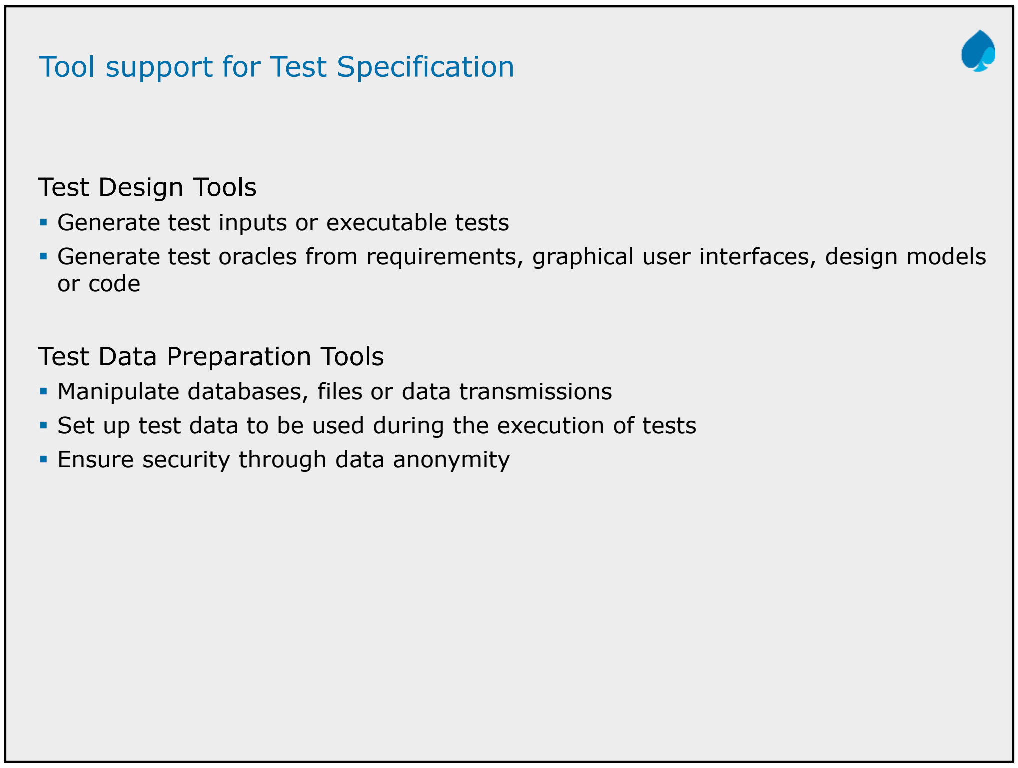
Static Analysis Tools

- Help developers and testers find defects prior to dynamic testing
- Provide support for enforcing coding standards
- Help in planning or risk analysis by providing the metrics for the code (e.g., complexity)

Modeling Tools

- Used to validate software models (e.g., physical data model for a RDBMS)
- Help in finding defects
- Aid in generating some test cases based on the model

Static analysis tools are useful during design and coding phases by identifying gaps in the code.



Tool support for Test Specification

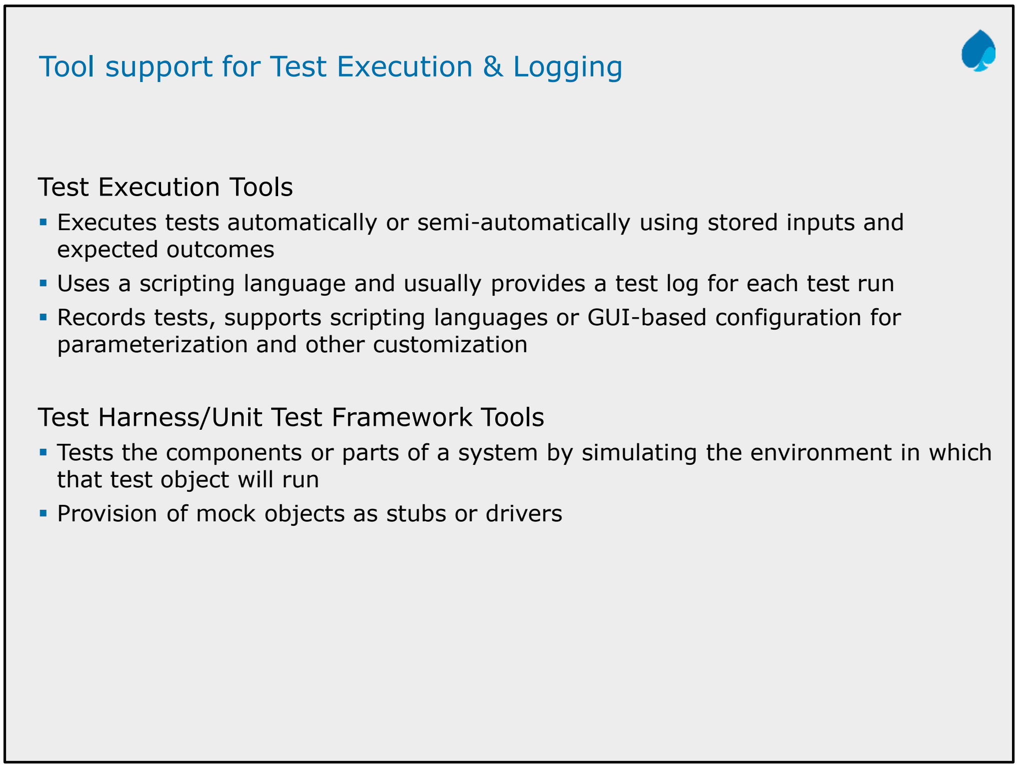
Test Design Tools

- Generate test inputs or executable tests
- Generate test oracles from requirements, graphical user interfaces, design models or code

Test Data Preparation Tools

- Manipulate databases, files or data transmissions
- Set up test data to be used during the execution of tests
- Ensure security through data anonymity

Test design tool supports the test design activity by generating test inputs, from a specification that may be held in a CASE tool repository. For example, if the requirements are kept in a requirements management or test management tool, or in a CASE tool used by developers, then it is possible to identify the input fields, including the range of valid values. If the valid range is stored, the tool can distinguish between values that accepts and generates an error message.



Tool support for Test Execution & Logging

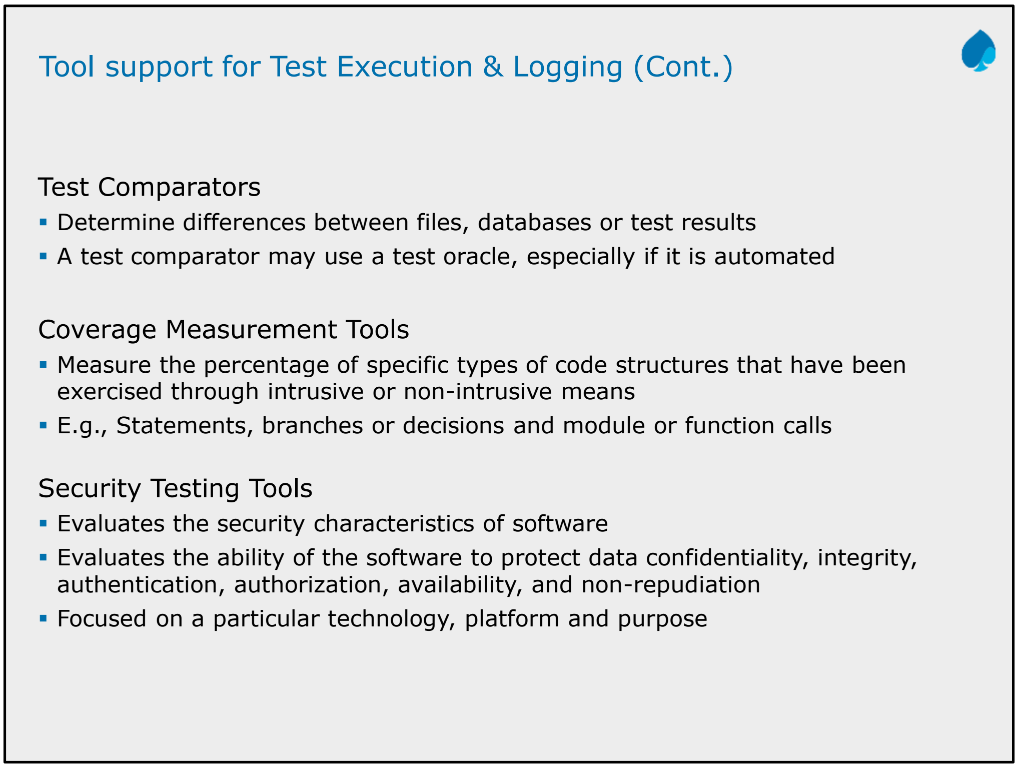
Test Execution Tools

- Executes tests automatically or semi-automatically using stored inputs and expected outcomes
- Uses a scripting language and usually provides a test log for each test run
- Records tests, supports scripting languages or GUI-based configuration for parameterization and other customization

Test Harness/Unit Test Framework Tools

- Tests the components or parts of a system by simulating the environment in which that test object will run
- Provision of mock objects as stubs or drivers

Test harness and drivers, and dynamic analysis tools are used during integration testing.



Tool support for Test Execution & Logging (Cont.)

Test Comparators

- Determine differences between files, databases or test results
- A test comparator may use a test oracle, especially if it is automated

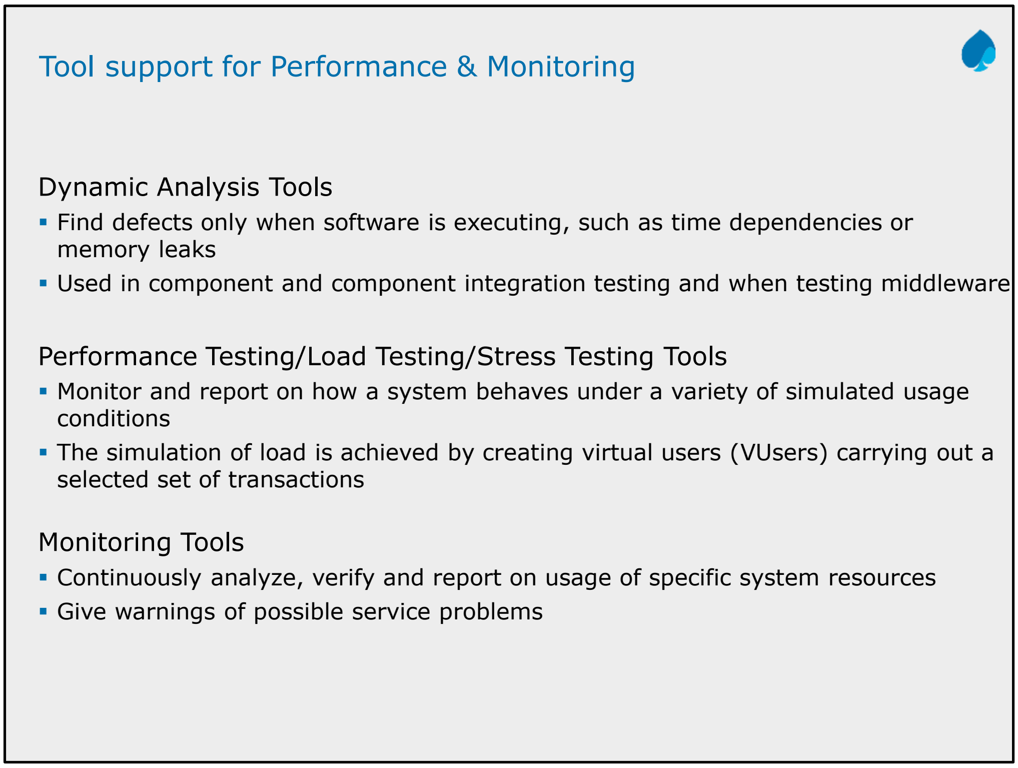
Coverage Measurement Tools

- Measure the percentage of specific types of code structures that have been exercised through intrusive or non-intrusive means
- E.g., Statements, branches or decisions and module or function calls

Security Testing Tools

- Evaluates the security characteristics of software
- Evaluates the ability of the software to protect data confidentiality, integrity, authentication, authorization, availability, and non-repudiation
- Focused on a particular technology, platform and purpose

Test run and comparison tools are used in all levels of testing to compare results. Security testing can be started during coding phase and can continue till the system is moved to production, and sometimes after.



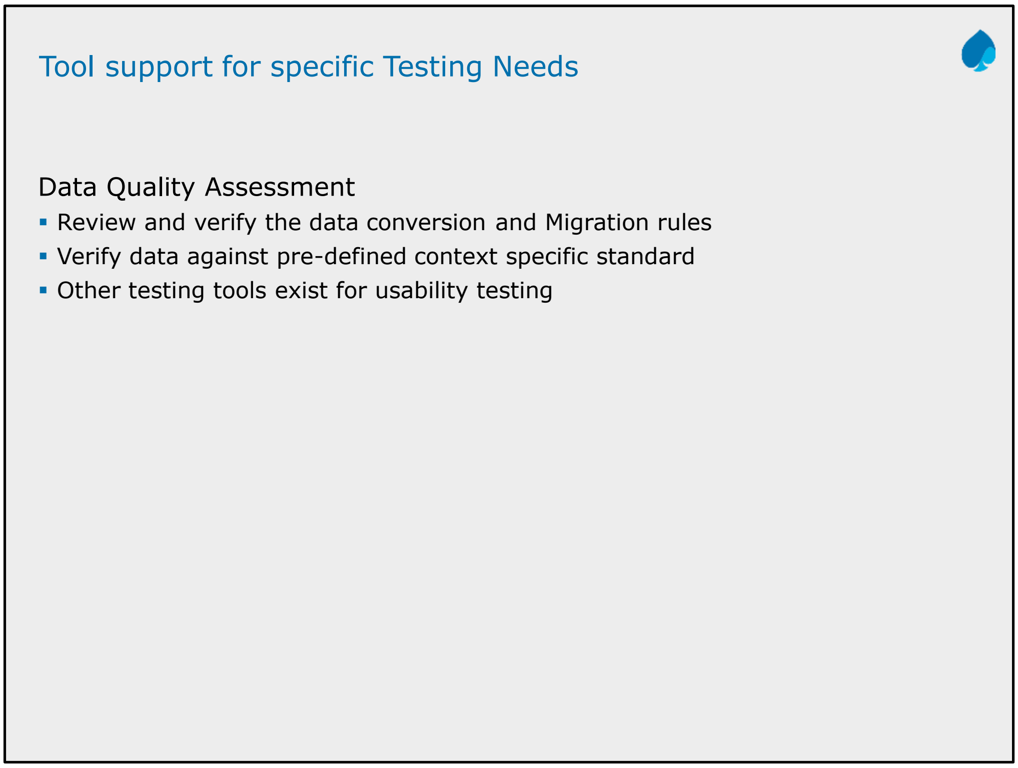
Tool support for Performance & Monitoring

Dynamic Analysis Tools
- Find defects only when software is executing, such as time dependencies or memory leaks
- Used in component and component integration testing and when testing middleware

Performance Testing/Load Testing/Stress Testing Tools
- Monitor and report on how a system behaves under a variety of simulated usage conditions
- The simulation of load is achieved by creating virtual users (VUsers) carrying out a selected set of transactions

Monitoring Tools
- Continuously analyze, verify and report on usage of specific system resources
- Give warnings of possible service problems

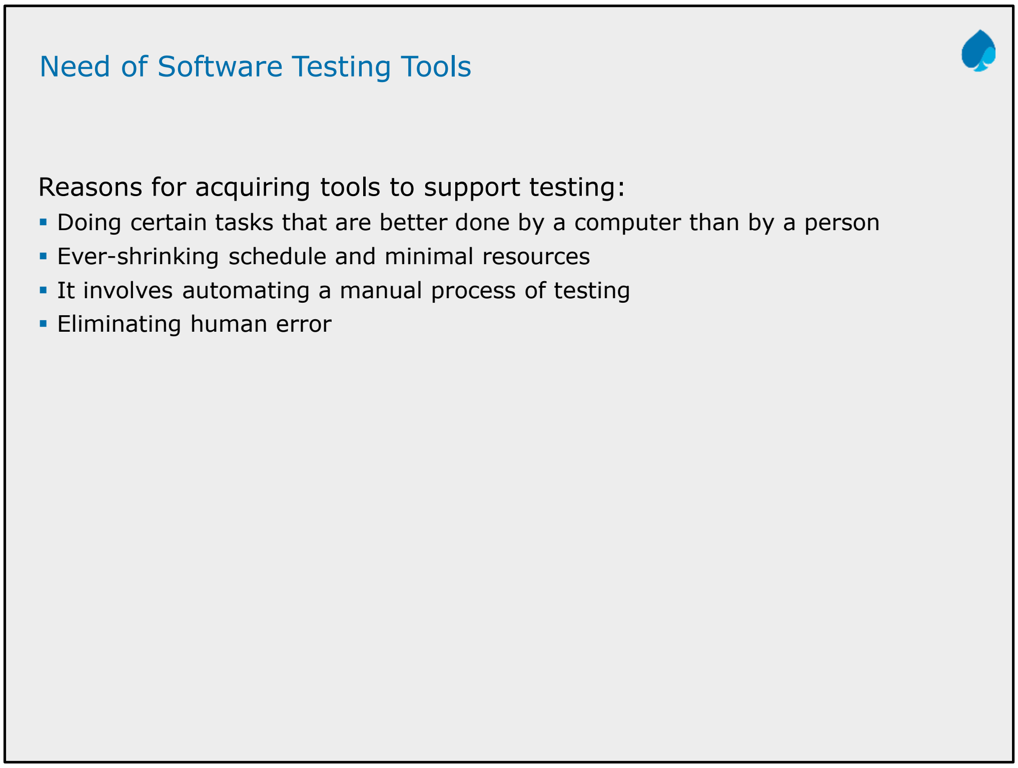
Performance measurement tools are used in system and acceptance testing to measure the system performance.
Dynamic analysis tool provides runtime information on the state of the software code that includes allocation, use, and de-allocation of resources, and flag of unassigned pointers or pointer arithmetic faults.



Tool support for specific Testing Needs

Data Quality Assessment

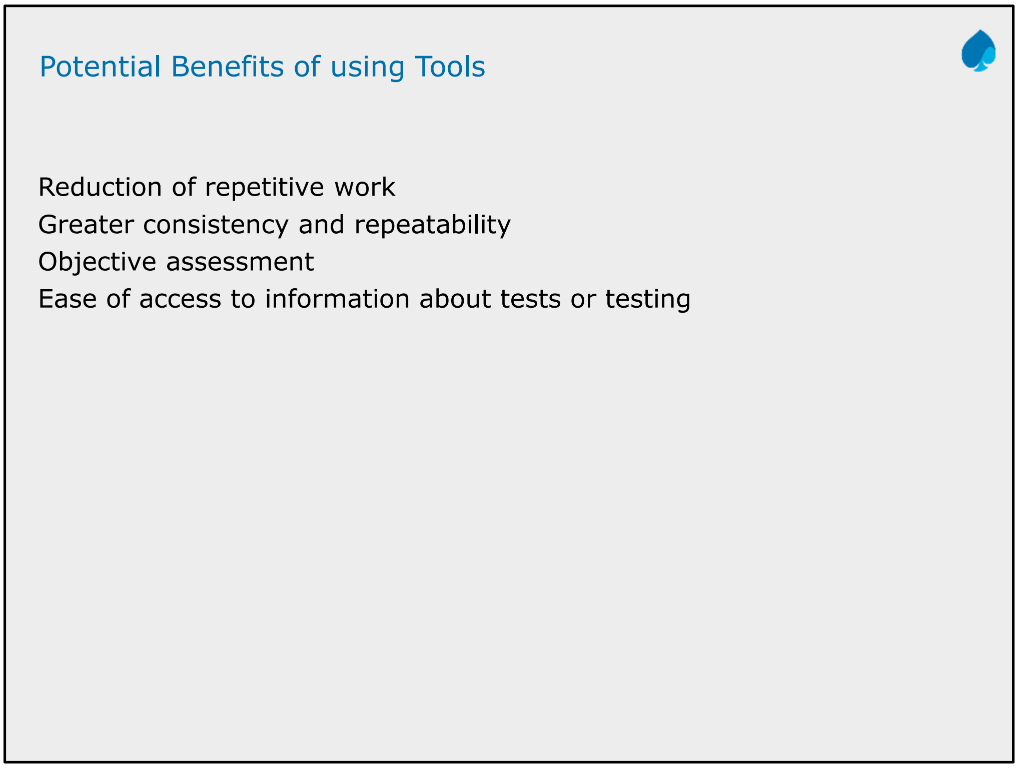
- Review and verify the data conversion and Migration rules
- Verify data against pre-defined context specific standard
- Other testing tools exist for usability testing



Need of Software Testing Tools

Reasons for acquiring tools to support testing:

- Doing certain tasks that are better done by a computer than by a person
- Ever-shrinking schedule and minimal resources
- It involves automating a manual process of testing
- Eliminating human error



Potential Benefits of using Tools

Reduction of repetitive work
Greater consistency and repeatability
Objective assessment
Ease of access to information about tests or testing

Manual testing is dependent on the style and nature of the individual performing the test. Hence, it differs from person to person. Use of tools remove this variation as they can only perform the task they are programmed for.

Risks of using Tools

Unrealistic expectations from the tool

Under estimating the time, cost and effort while initial introduction of a tool

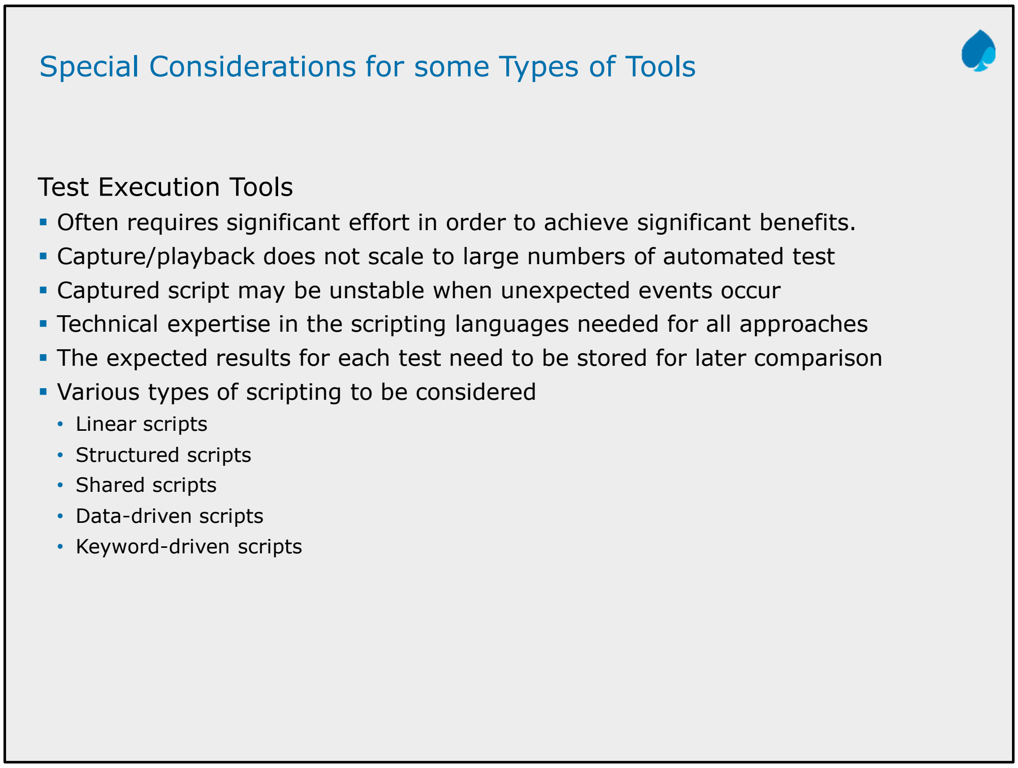
Under estimating the time and effort needed to achieve significant and continuing benefits from the tool

Under estimating the effort required to maintain the test assets

Over-reliance on the tool

Poor response from vendor for support, upgrades and defect fixes

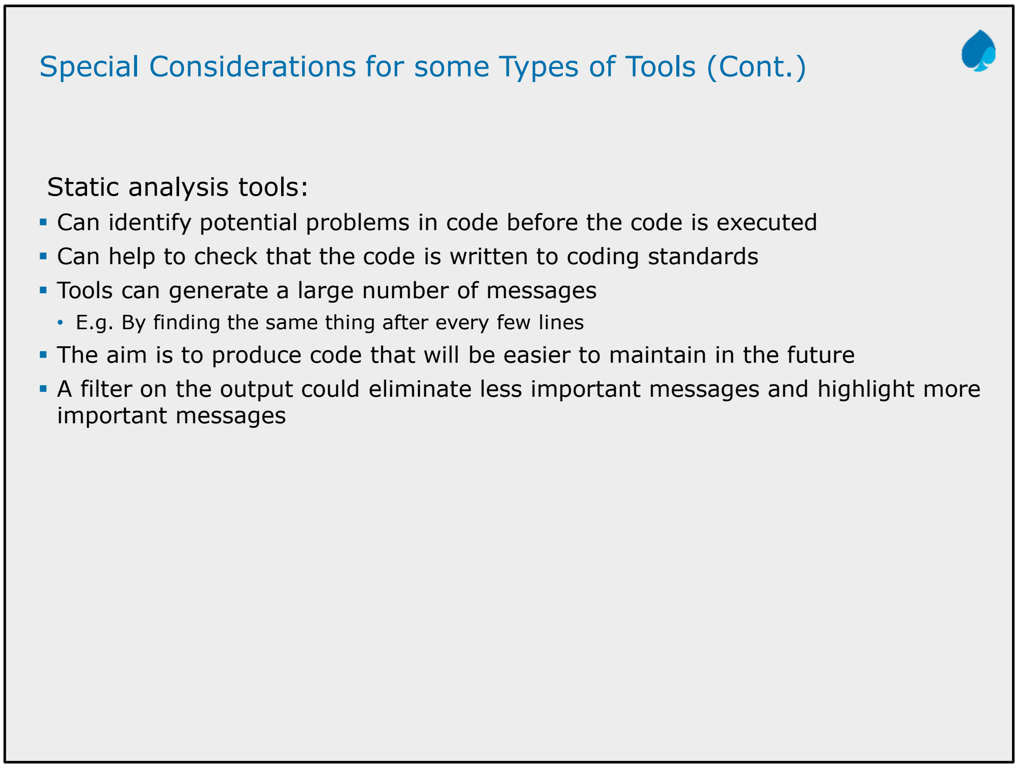
Risk of suspension of open-source / free tool project



Special Considerations for some Types of Tools

Test Execution Tools

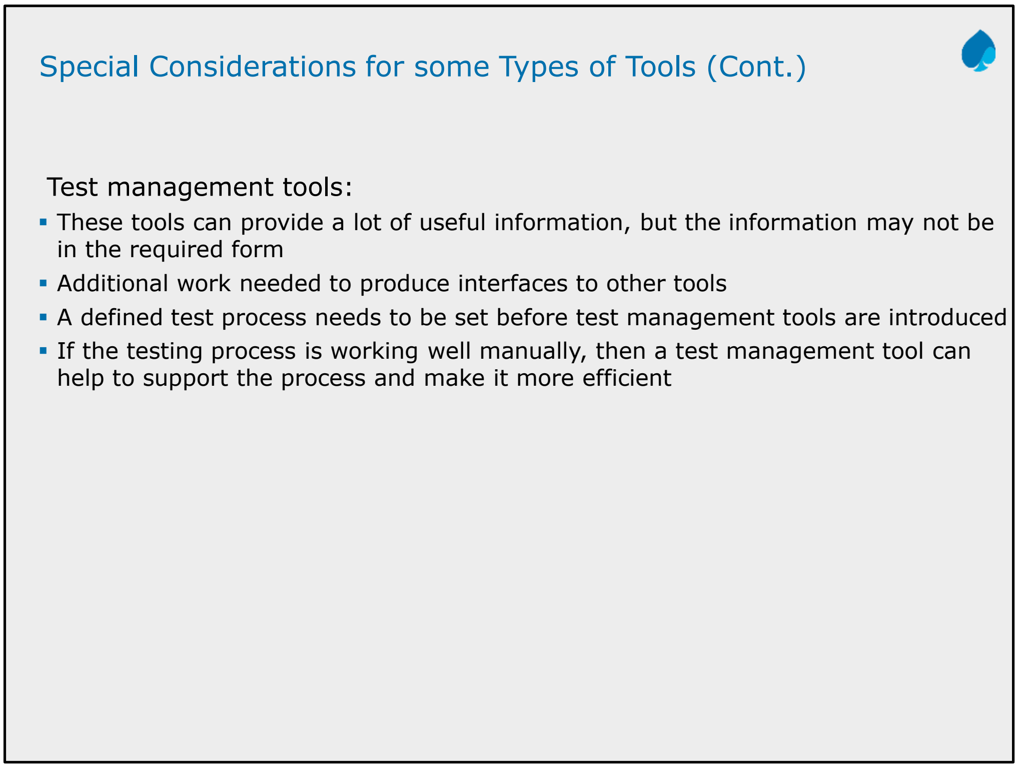
- Often requires significant effort in order to achieve significant benefits.
- Capture/playback does not scale to large numbers of automated test
- Captured script may be unstable when unexpected events occur
- Technical expertise in the scripting languages needed for all approaches
- The expected results for each test need to be stored for later comparison
- Various types of scripting to be considered
  - Linear scripts
  - Structured scripts
  - Shared scripts
  - Data-driven scripts
  - Keyword-driven scripts



Special Considerations for some Types of Tools (Cont.)

Static analysis tools:

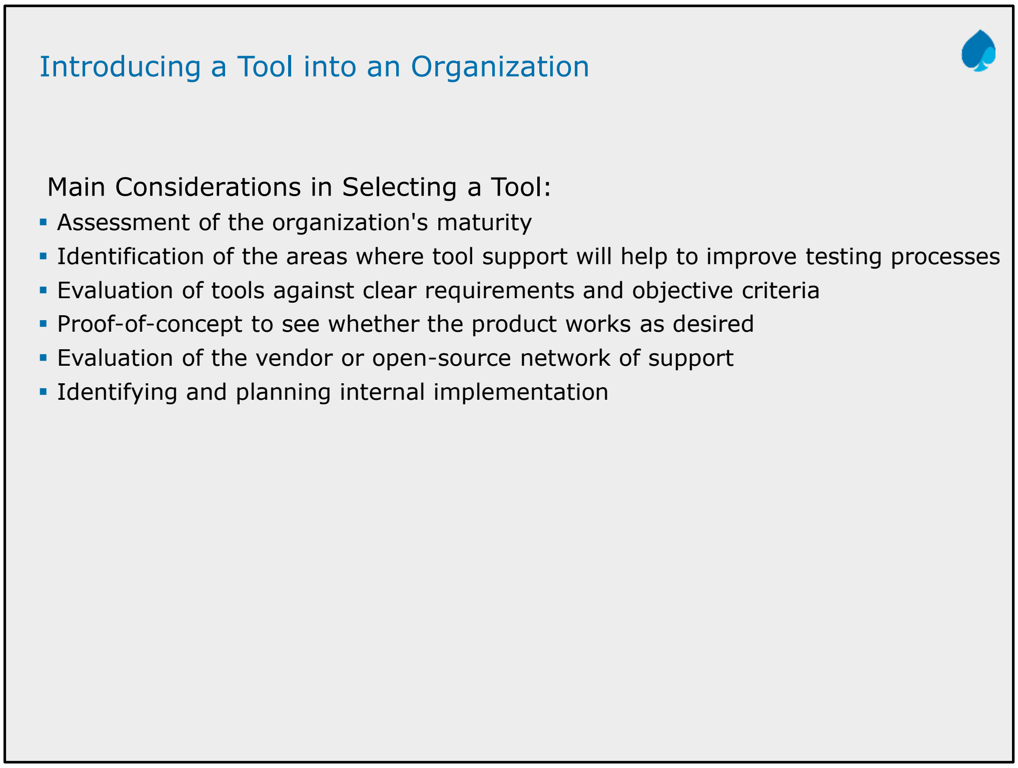
- Can identify potential problems in code before the code is executed
- Can help to check that the code is written to coding standards
- Tools can generate a large number of messages
  - E.g. By finding the same thing after every few lines
- The aim is to produce code that will be easier to maintain in the future
- A filter on the output could eliminate less important messages and highlight more important messages



Special Considerations for some Types of Tools (Cont.)

Test management tools:

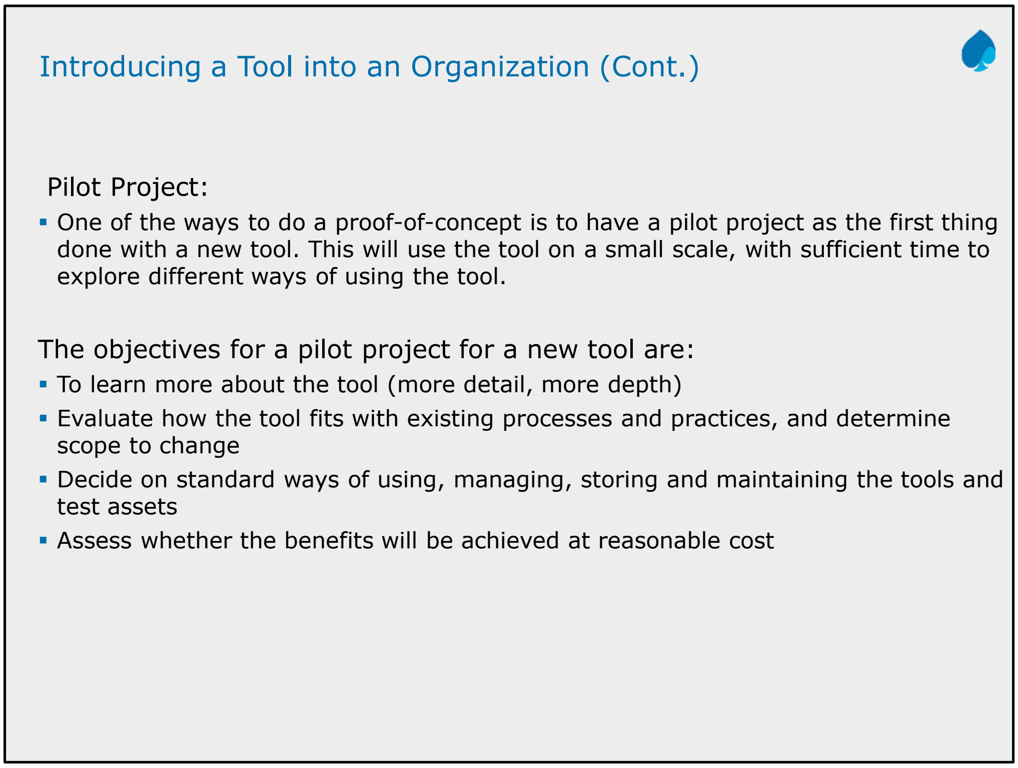
- These tools can provide a lot of useful information, but the information may not be in the required form
- Additional work needed to produce interfaces to other tools
- A defined test process needs to be set before test management tools are introduced
- If the testing process is working well manually, then a test management tool can help to support the process and make it more efficient



Introducing a Tool into an Organization

Main Considerations in Selecting a Tool:

- Assessment of the organization's maturity
- Identification of the areas where tool support will help to improve testing processes
- Evaluation of tools against clear requirements and objective criteria
- Proof-of-concept to see whether the product works as desired
- Evaluation of the vendor or open-source network of support
- Identifying and planning internal implementation



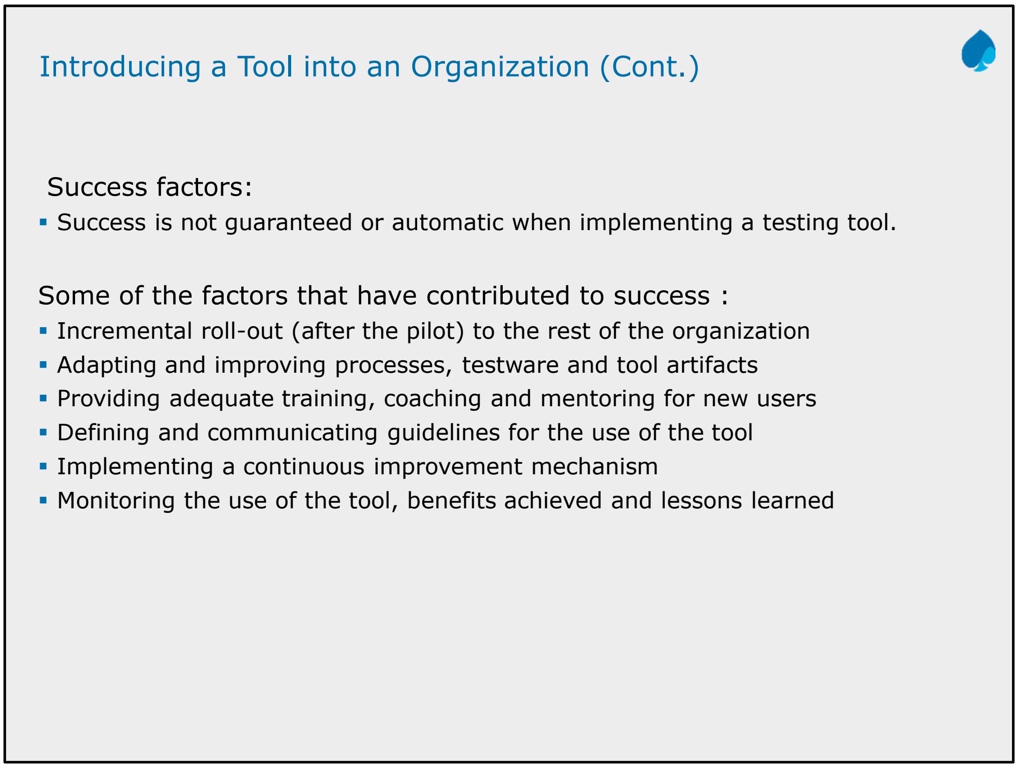
Introducing a Tool into an Organization (Cont.)

Pilot Project:

- One of the ways to do a proof-of-concept is to have a pilot project as the first thing done with a new tool. This will use the tool on a small scale, with sufficient time to explore different ways of using the tool.

The objectives for a pilot project for a new tool are:

- To learn more about the tool (more detail, more depth)
- Evaluate how the tool fits with existing processes and practices, and determine scope to change
- Decide on standard ways of using, managing, storing and maintaining the tools and test assets
- Assess whether the benefits will be achieved at reasonable cost



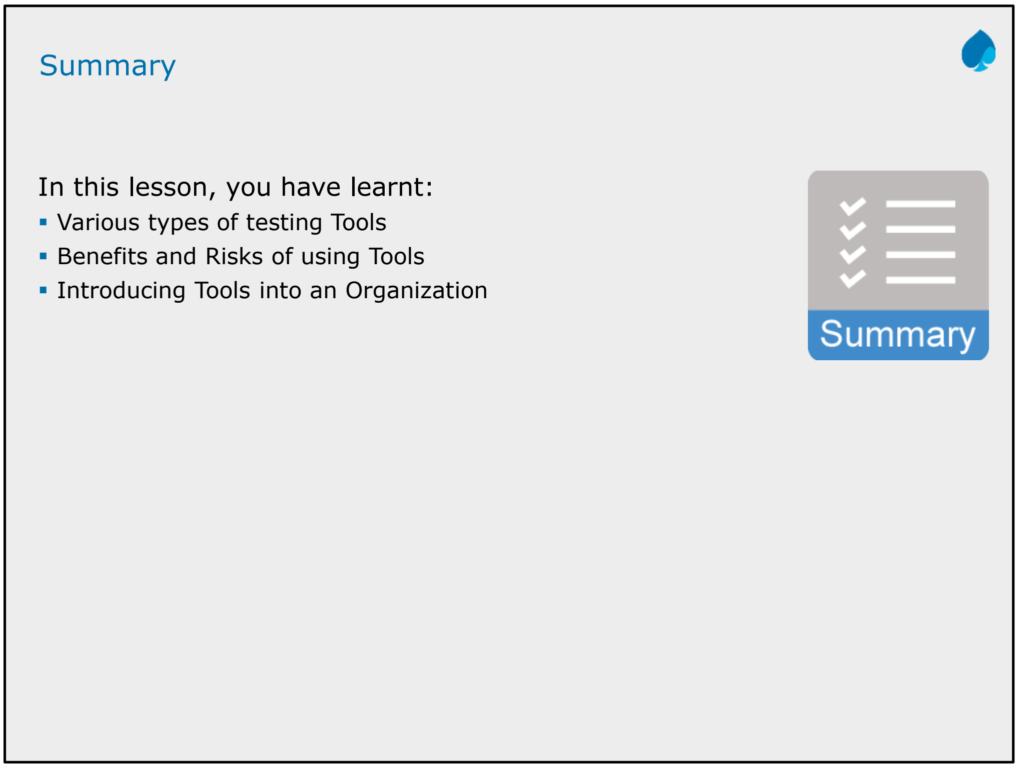
Introducing a Tool into an Organization (Cont.)

Success factors:

- Success is not guaranteed or automatic when implementing a testing tool.

Some of the factors that have contributed to success :

- Incremental roll-out (after the pilot) to the rest of the organization
- Adapting and improving processes, testware and tool artifacts
- Providing adequate training, coaching and mentoring for new users
- Defining and communicating guidelines for the use of the tool
- Implementing a continuous improvement mechanism
- Monitoring the use of the tool, benefits achieved and lessons learned



Summary

In this lesson, you have learnt:
- Various types of testing Tools
- Benefits and Risks of using Tools
- Introducing Tools into an Organization

Summary

Review - Questions

Question 1: The ____________________tool is used to detect a memory leak.

Question 2Tool supported for static testing is a good way to force failures into the software.

- Option: True / False

Question 3: Goal of Pilot Project for tool evaluation is to evaluate how the tool fits with existing processes and practices.

- Option: True / False