

# Tool support for testing

## Chapter 6

- 1. Test tool considerations and classification**
- 2. Potential benefits and risks**
- 3. Effective use of tools – introducing a test tool to an organization**



# Types of test tools

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

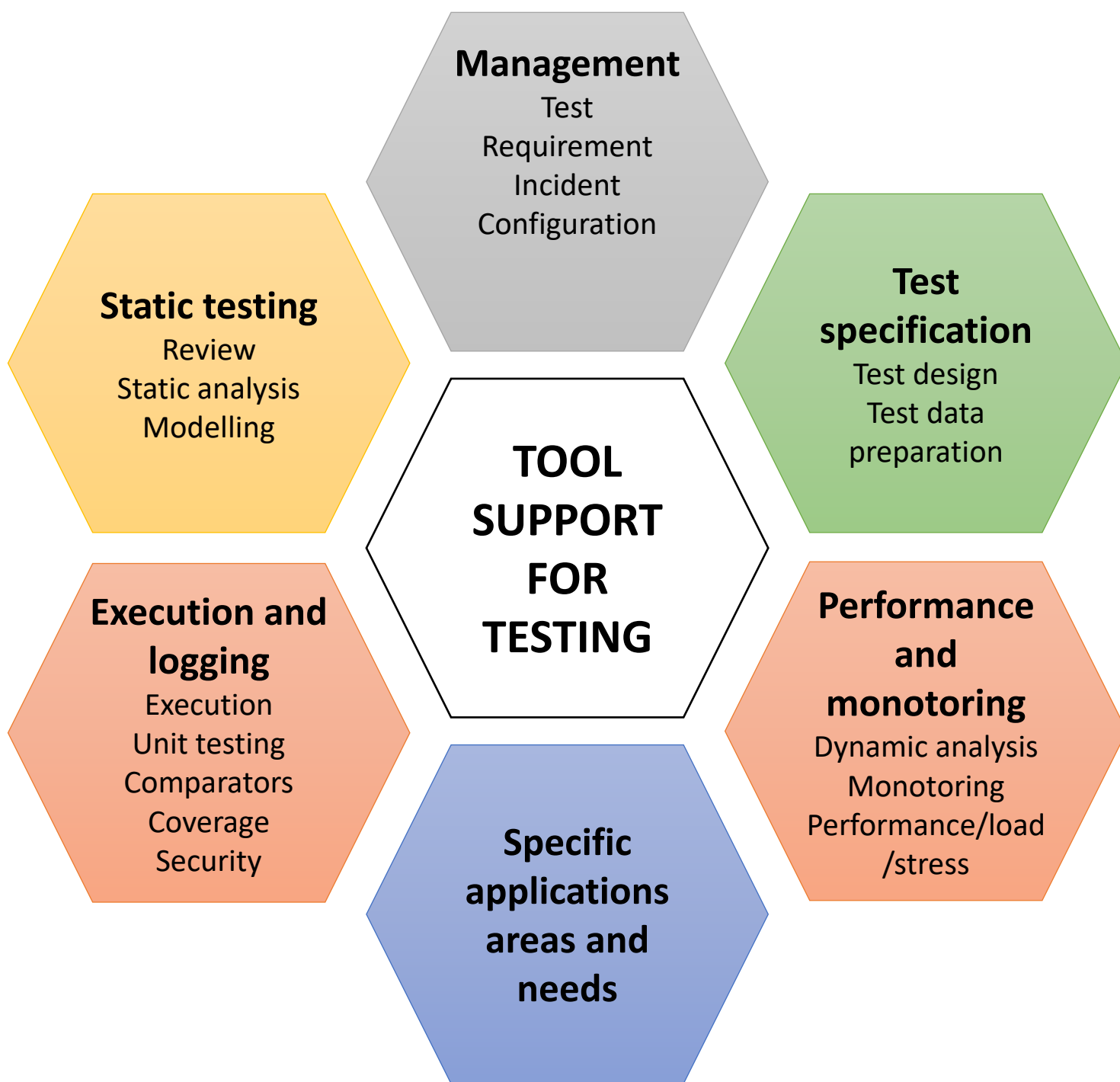
## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

✓ **LO: Classify different types of test tools according to their purpose and to the activities in the test process and in the software life-cycle**

✓ **LO: For each type of test tool, explain how it supports testing**



# Tool support for testing – types of tools

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

**Test tools** can be used for one or more **activities that support testing**:

- Tools that are **directly used** in testing (e.g. test execution tools, test data generation tools, result comparison tools)
- Tools that help in **managing the testing process** (e.g. test results, requirements, incidents, defects) and for **monitoring** and **reporting** the test execution
- Tools that are used in **exploration** (e.g. tools that monitor the file activity for an application)
- Any tool that aids in testing



# Tool support for testing - purposes

## 1. Types of test tools

- **1.1 Tool support for testing**
- **1.2 Test tool classification**
- **1.3 Tools for test management**
- **1.4 Tools for static testing**
- **1.5 Tools for test design and specification**
- **1.6 Tools for execution and logging**
- **1.7 Tools for performance & monitoring**
- **1.8 Tools for specific testing needs**

## 2. Effective use of test tools

- **2.1 Potential benefits and risks of tools**
- **2.2 Special considerations for tools**

## 3. Introducing a test tool into an organization

- **Tools support** for testing can have one or more of the following purposes, depending on the context:
  - **improve the efficiency** of the test activities (e.g.: by automating repetitive tasks)
  - **automate activities** that require **significant resources** when done manually (e.g. static testing)
  - **automate activities** that **cannot be done manually** (e.g. large-scale performance testing of client-server applications)
  - **increase reliability** of testing (by automating large data comparisons or simulating complex behavior)



# Test tool classification

## 1. Types of test tools

- 1.1 Tool support for testing
- **1.2 Test tool classification**
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

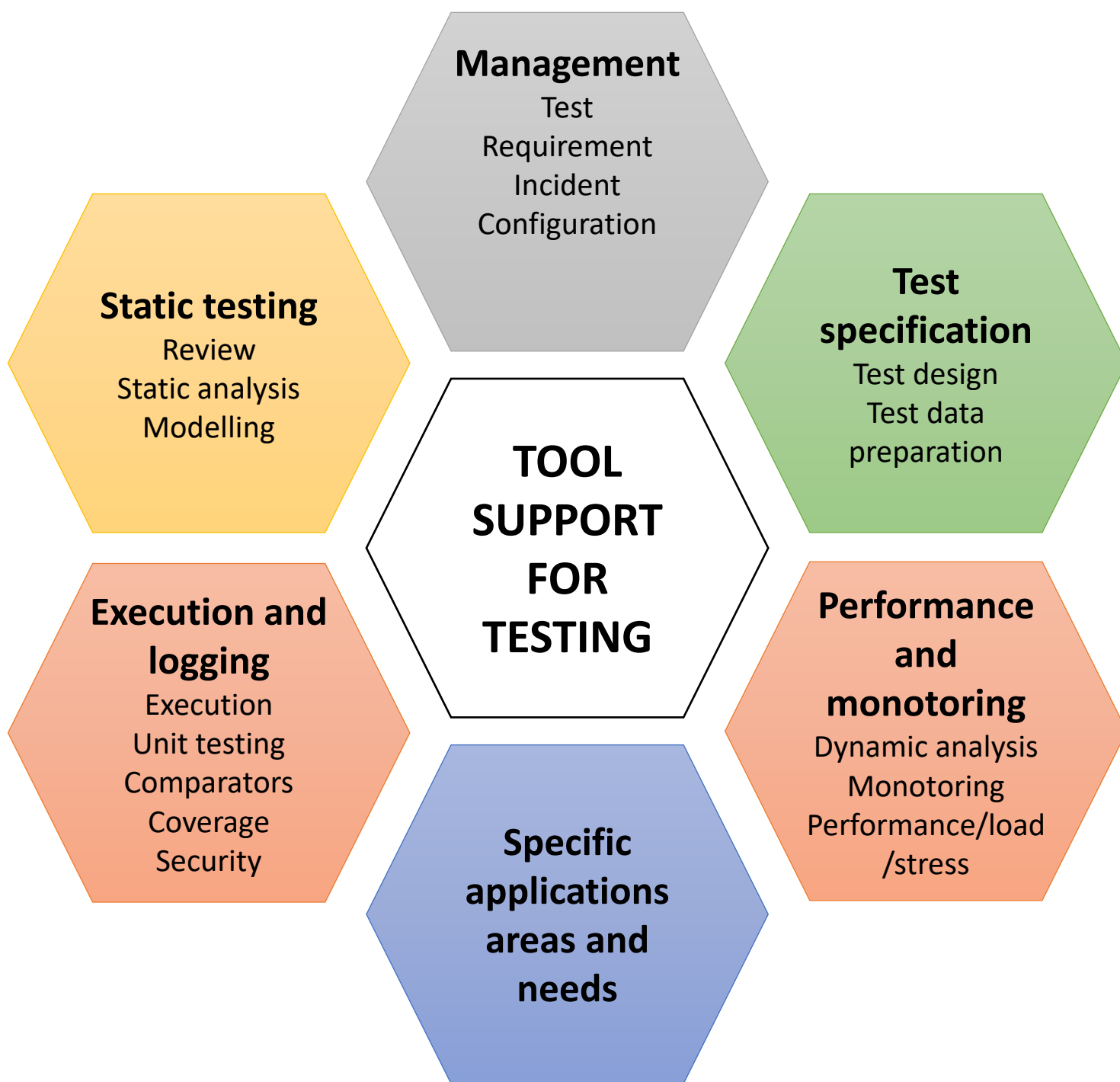
**Tools** are **classified** according to the **testing activities** that they support.

- **one** activity
- **more than one** activity, but classification falls under **the main activity**

## Notes

- Some types of test tool can be **intrusive** - the tool itself can **affect the outcome of the test**. (e.g. timing measurements may be different depending on how you measure it with different performance tools).
- The **consequence** of intrusive tools is called the **probe effect**.
- Some tools offer support more appropriate for **developers**. Such tools are marked with “(D)” in this chapter.





# Tools support for **management** of testing & tests

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- **1.3 Tools for test management**
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Characteristics

- Support for the **management** of **tests** and the **testing activities**.
- Support for **traceability** of **tests**, test results and incidents **to source documents**, such as requirements specifications.
- Generation of **progress reports**
- Logging test **results**
- Monitoring
- Offer info on **metrics** related to the tests.





# Tools support for **management** of testing & tests

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- **1.3 Tools for test management**
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Types

### **Test management tools** and **application lifecycle management tool (ALM)**

- Requirements management tools
- Defect management tools
- Configuration management tools
- Continuous integrations tool (D)



# Tools support for **management** of testing & tests

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- **1.3 Tools for test management**
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Types

- Test management tools and application lifecycle management tool (ALM)

## Requirements management tools

- Store requirements
- Check for consistency and undefined (missing) requirements
- Allow prioritization
- Enable individual tests to be traceable to requirements

- Defect management tools

- Configuration management tools

- Continuous integrations tool (D)



# Tools support for **management** of testing & tests

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- **1.3 Tools for test management**
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Types

- Test management tools and application lifecycle management tool (ALM)
- Requirements management tools

## Defect management tools

- Configuration management tools
- Continuous integrations tool (D)



# Tools support for **management** of testing & tests

## 1. Types of test tools

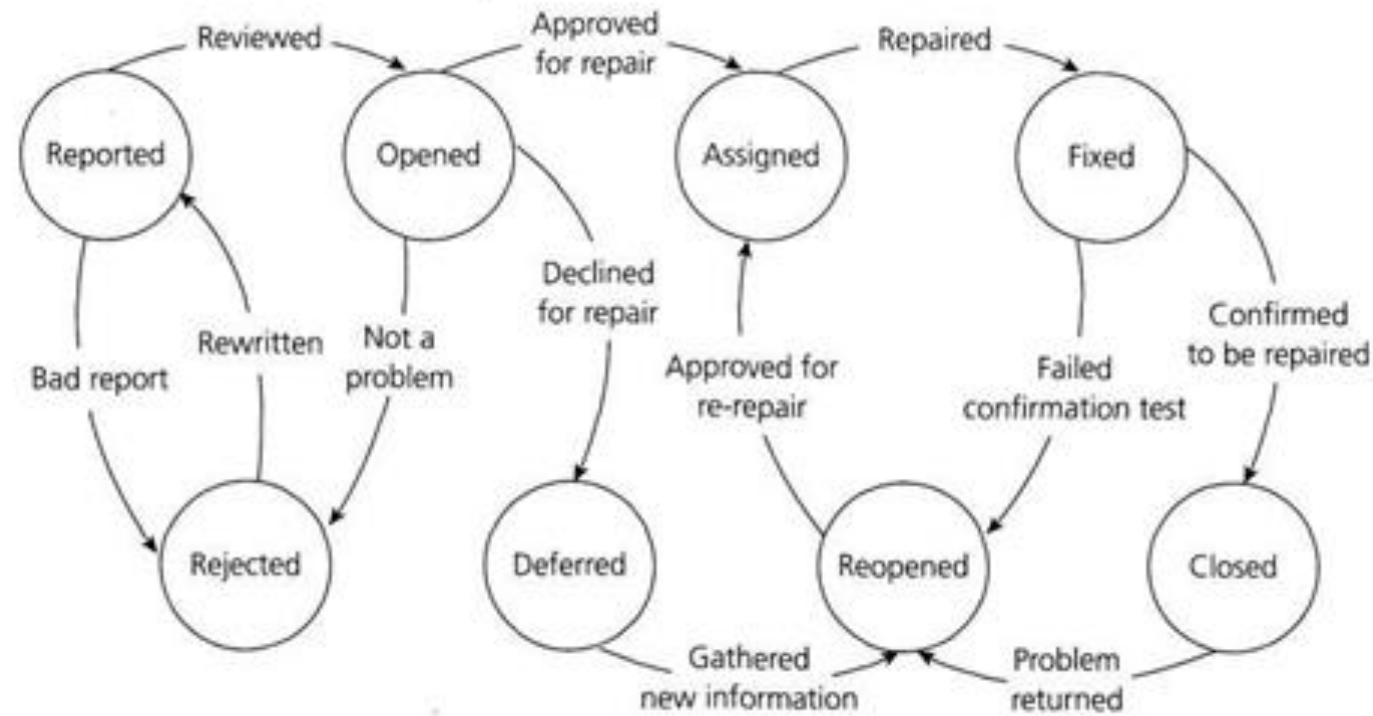
- 1.1 Tool support for testing
- 1.2 Test tool classification
- **1.3 Tools for test management**
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

- Statuses of incident reports:



**FIGURE 5.3** Incident report life cycle



# Tools support for **management** of testing & tests

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- **1.3 Tools for test management**
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Types

- Test management tools and application lifecycle management tool (ALM)
- Requirements management tools
- Defect management tools

## **Configuration** management tools

- Are necessary to keep track of different versions and builds of SW and tests
- Are particularly useful when developing on more than one configuration of the HW/SW environment
- Continuous integrations tool (D)



# Tools support for **management** of testing & tests

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- **1.3 Tools for test management**
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

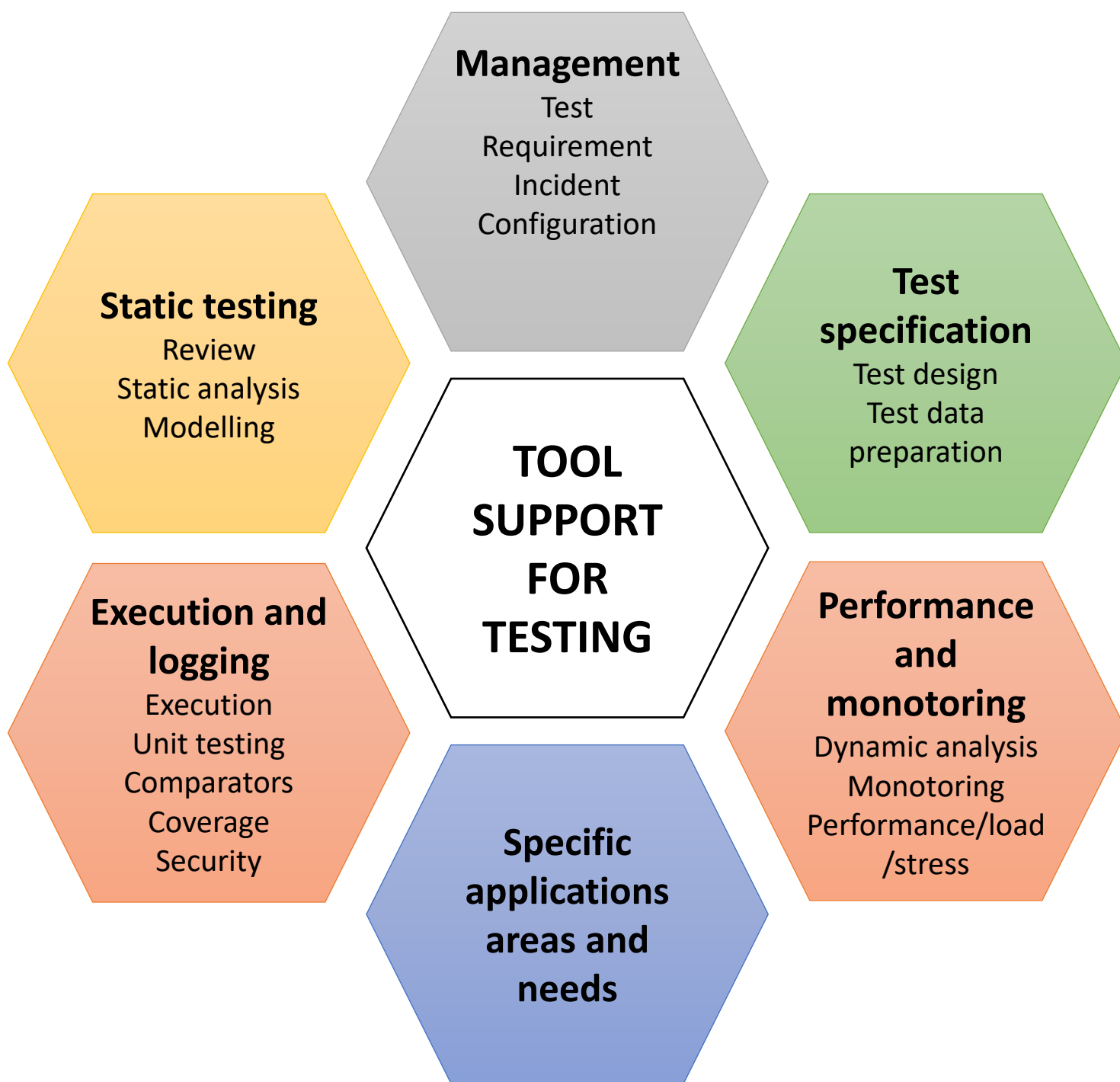
## 3. Introducing a test tool into an organization

## Types

- Test management tools and application lifecycle management tool (ALM)
- Requirements management tools
- Defect management tools
- Configuration management tools

**Continuous integrations tool (D)**





# Tools support for **static testing**

## 1. Types of test tools

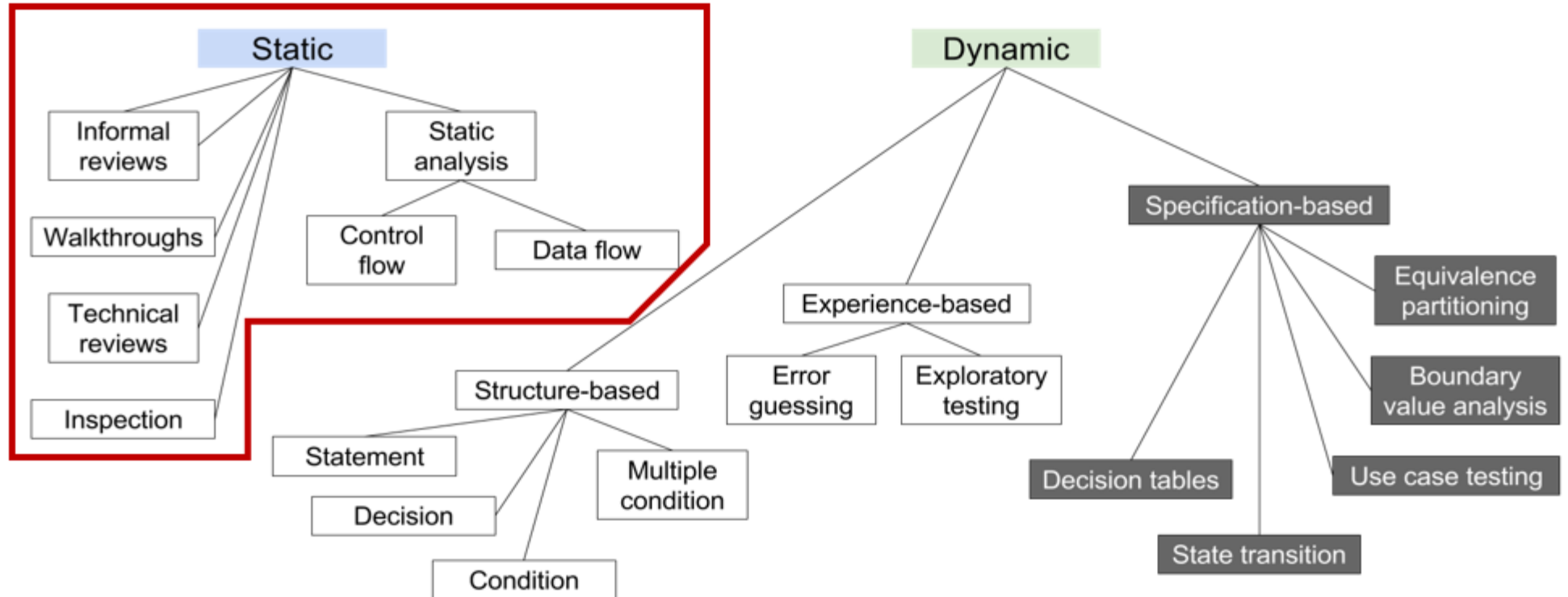
- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- **1.4 Tools for static testing**
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

Recall the different **types** of testing





# Tools support for **static testing**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- **1.4 Tools for static testing**
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

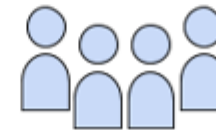
## Tools for static testing

Tools that **aid** in **improving** the **code** / work product, **without executing** it

## Categories

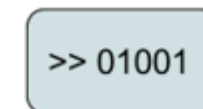
### Review tools

Supports the review process



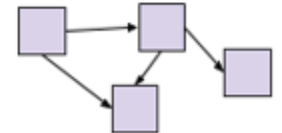
### Static analysis tools

Supports code examination



### Modelling tools

Validate models of system / software



# Tools support for **static testing**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- **1.4 Tools for static testing**
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Review process tools

Common **reference** for the **review** processes conducted

Keep **track** of all the **information** from the review process

Store and **communicate** review **comments**, report on **defects** and **effort**

**Monitoring** review status → Passed, passed with corrections, requires re-review

## When to use?

Suitable for more **formal** review processes

**Geographically dispersed** teams



# Tools support for static testing

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

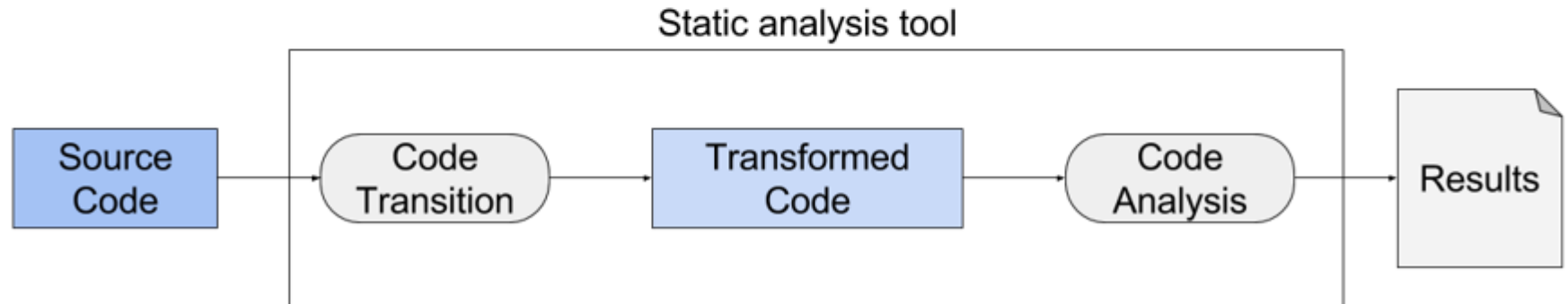
## Static analysis tools (D)

Mostly used by **developers** → **Component** (unit) testing

**Tool** is **executed** → Code is **not**

The **source code** serves as **input** data to the **tool**

Extension of **compiler** technology



# Tools support for **static testing**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- **1.4 Tools for static testing**
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Static analysis tools (D)

Support developers and testers in **finding defects** before dynamic testing

## Purpose

To better **understand** the code, and find ways of **improving** it

## Common features

Calculate **metrics** → **Complexity, nesting** levels → Identify areas of **risk**

Enforce coding **standards**

Analyse code **structures** and **dependencies**

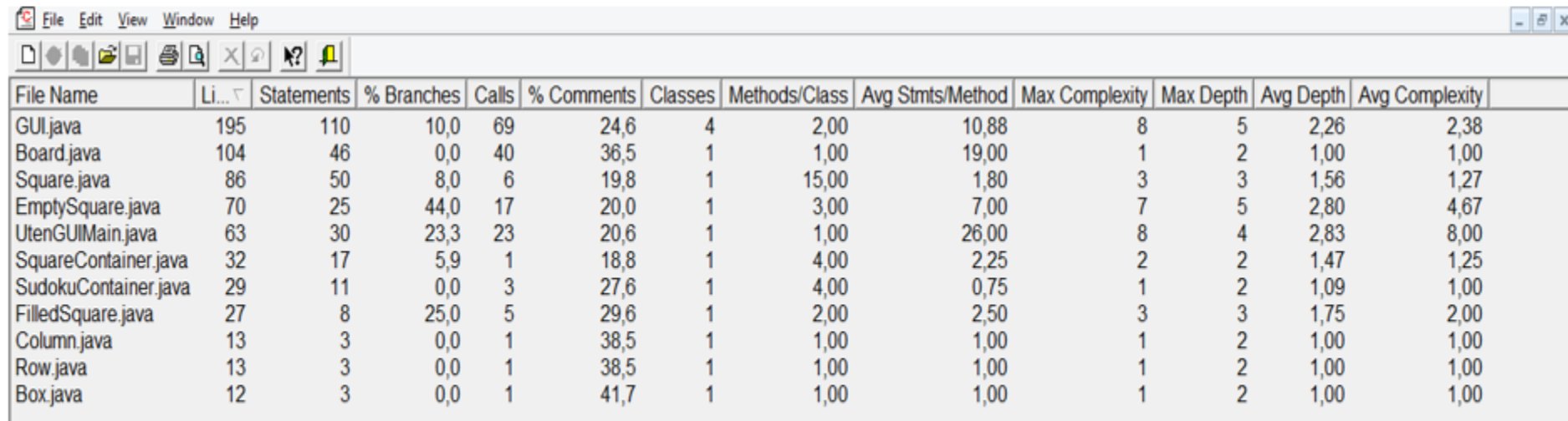


# Static analysis tools (D)

## Static analysis tool example: Source Monitor

Collects **metrics** from **source** code files

**Displays** and prints metrics in **tables** and **charts**



The screenshot shows a window titled "Source Monitor" with a menu bar (File, Edit, View, Window, Help) and a toolbar. Below the toolbar is a table with 13 columns: File Name, Li..., Statements, % Branches, Calls, % Comments, Classes, Methods/Class, Avg Stmt/Method, Max Complexity, Max Depth, Avg Depth, and Avg Complexity. The table lists metrics for 12 Java files.

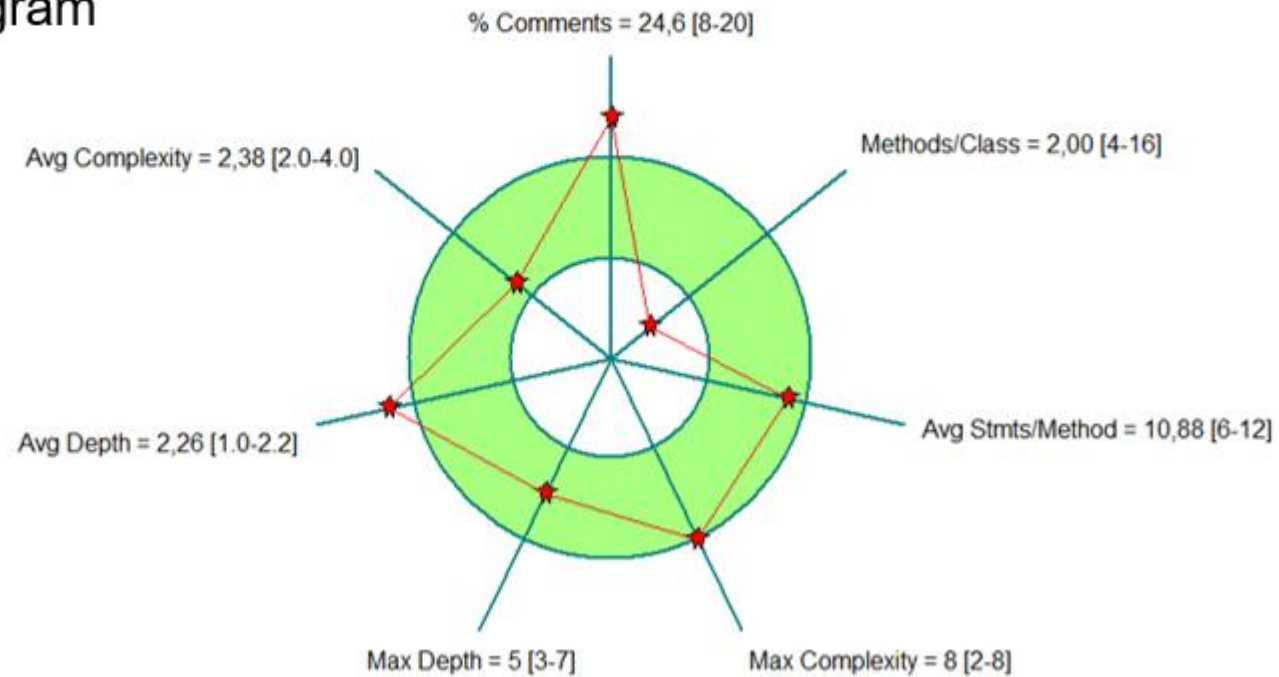
File Name	Li...	Statements	% Branches	Calls	% Comments	Classes	Methods/Class	Avg Stmt/Method	Max Complexity	Max Depth	Avg Depth	Avg Complexity
GUI.java	195	110	10,0	69	24,6	4	2,00	10,88	8	5	2,26	2,38
Board.java	104	46	0,0	40	36,5	1	1,00	19,00	1	2	1,00	1,00
Square.java	86	50	8,0	6	19,8	1	15,00	1,80	3	3	1,56	1,27
EmptySquare.java	70	25	44,0	17	20,0	1	3,00	7,00	7	5	2,80	4,67
UtenGUIMain.java	63	30	23,3	23	20,6	1	1,00	26,00	8	4	2,83	8,00
SquareContainer.java	32	17	5,9	1	18,8	1	4,00	2,25	2	2	1,47	1,25
SudokuContainer.java	29	11	0,0	3	27,6	1	4,00	0,75	1	2	1,09	1,00
FilledSquare.java	27	8	25,0	5	29,6	1	2,00	2,50	3	3	1,75	2,00
Column.java	13	3	0,0	1	38,5	1	1,00	1,00	1	2	1,00	1,00
Row.java	13	3	0,0	1	38,5	1	1,00	1,00	1	2	1,00	1,00
Box.java	12	3	0,0	1	41,7	1	1,00	1,00	1	2	1,00	1,00



# Static analysis tools (D)

## Static analysis tool example: Source Monitor

### Kiviat diagram



# Tools support for static testing

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Modelling tools (D)

Validate models of the system / software

## Purpose

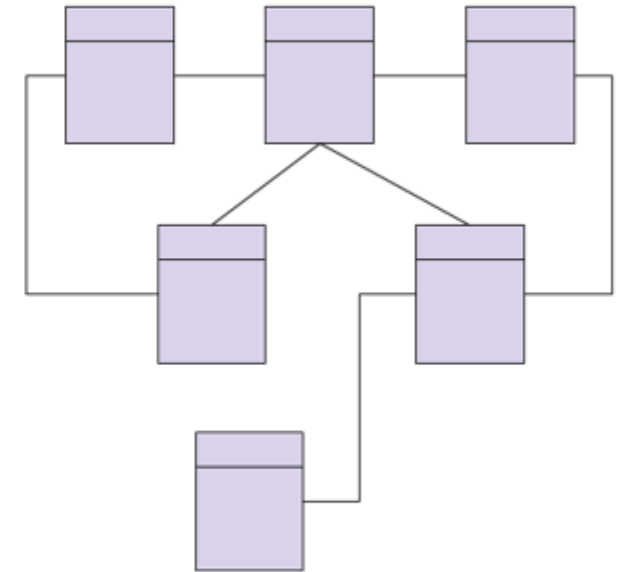
To better aid in designing the software

## Common features and characteristics

Identify inconsistencies and defects within the models

Identify and prioritise risk areas

Predicting system response and behaviour under various situations



# Tools support for **static testing**

## Modelling tool example: **Star UML**

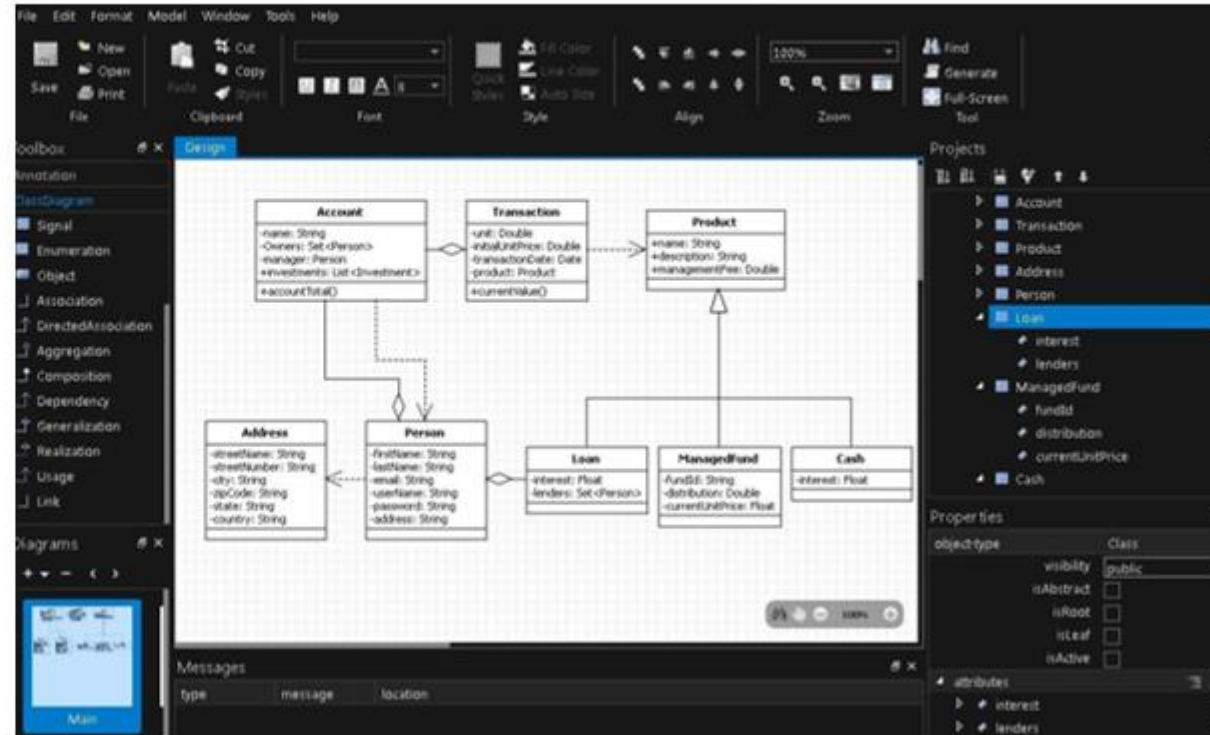
UML tool

Variety of **diagrams**

Class / Domain

Use case

Sequence





# Tools support for **static testing**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- **1.4 Tools for static testing**
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

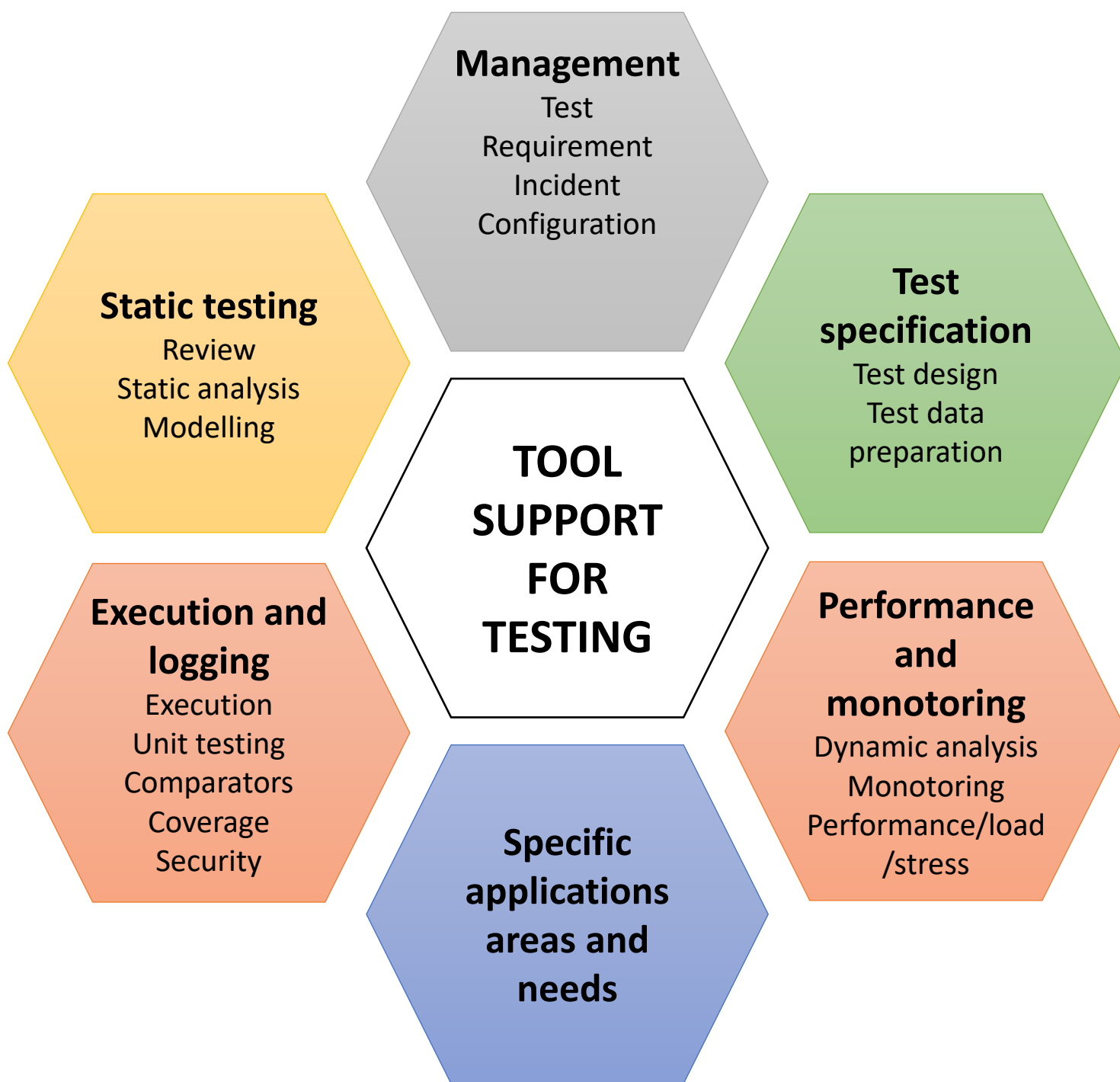
- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

The major **benefit** of **static testing tools** and **modeling tools** is **the cost effectiveness** of finding more defects at an **earlier time** in the development process.

As a result, the development process may **accelerate** and **improve** by having less rework.





# Tools support for test design and specification

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

- Test design tools
- Model-based testing tools
- Test data preparation tools
- TDD - Test driven development tool (D)
- ATDD - Acceptance test-driven development and BDD - Behavior-driven development



## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- **1.5 Tools for test design and specification**
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Test **design** tools

- generating test inputs values from:
  - requirements
  - test conditions
  - design models (state, data or object)
  - code
  - graphical user interfaces
- generating expected results, if an oracle is available to the tool.
- **Model-based testing tools**
- **Test data preparation tools**
- **TDD - Test driven development tool (D)**
- **ATDD - Acceptance test-driven development and BDD - Behavior-driven development**



# Tools support for **test design** and **specification**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- **1.5 Tools for test design and specification**
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

- Test design tools

### **Model-based testing**

- generating test inputs values or test cases from stored information the describes a model of the system, e.g. state transition model
- Test data preparation tools
- TDD - Test driven development tool (D)
- ATDD - Acceptance test-driven development and  
BDD - Behavior-driven development



# Tools support for **test design** and **specification**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- **1.5 Tools for test design and specification**
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

- Test design tools
- Model-based testing tools

## **Test data** preparation tools

- Generating extensive range or volume of data, if needed
  - Manipulate databases or files to set up test data to be used during the execution of the tests
- 
- TDD - Test driven development tool (D)
  - ATDD - Acceptance test-driven development and BDD - Behavior-driven development



# Tools support for **test design** and **specification**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- **1.5 Tools for test design and specification**
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Which **test activities** are supported by **test data preparation** tools?

### Common **features** of test **data preparation** tools

Data can be **selected** from an **existing** database

Data can be **created**, **generated**, and **altered** for use in **tests**

Construct a **large number** of similar records → Volume tests

### When to use?

During test **specification** and **control** → Test data **management** is **difficult**

**Ensure** the system under **test** is being tested **realistically**

Useful for **performance** and **reliability** testing

---



# Tools support for test design and specification

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

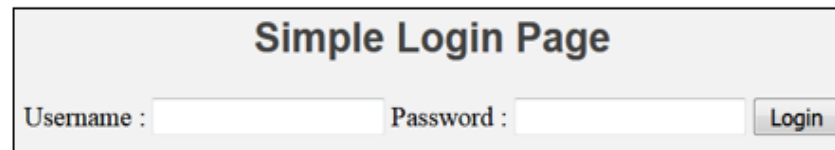
## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

Which test activities are supported by test data preparation tools?

Example: Simple login site



A screenshot of a web form titled "Simple Login Page". It contains two input fields: "Username :" and "Password :", followed by a "Login" button.

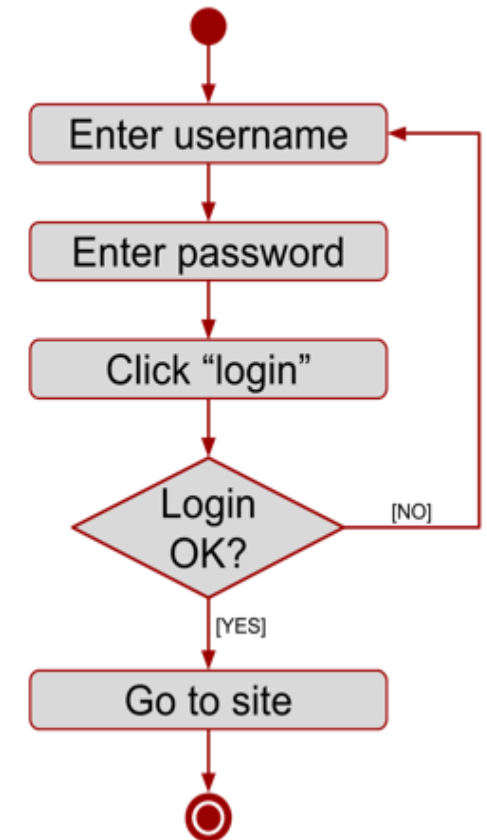
Data need for testing website

List of various usernames

List of various passwords

Database of existing users

We do not want to create all this by hand!





# Tools support for test design and specification

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

- Test design tools
- Model-based testing tools
- Test data preparation tools

## TDD - Test driven development tool (D)

- part of EXtreme Programming
  - used in Agile development
  - Recall the first guest lecture!
- 
- ATDD - Acceptance test-driven development and BDD - Behavior-driven development



# Tools support for **test design** and **specification**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- **1.5 Tools for test design and specification**
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

- Test design tools
- Model-based testing tools
- Test data preparation tools
- TDD - Test driven development tool (D)

**ATDD - Acceptance test-driven** development and

**BDD - Behavior-driven** development

- Natural language syntax, Given/When/Then:

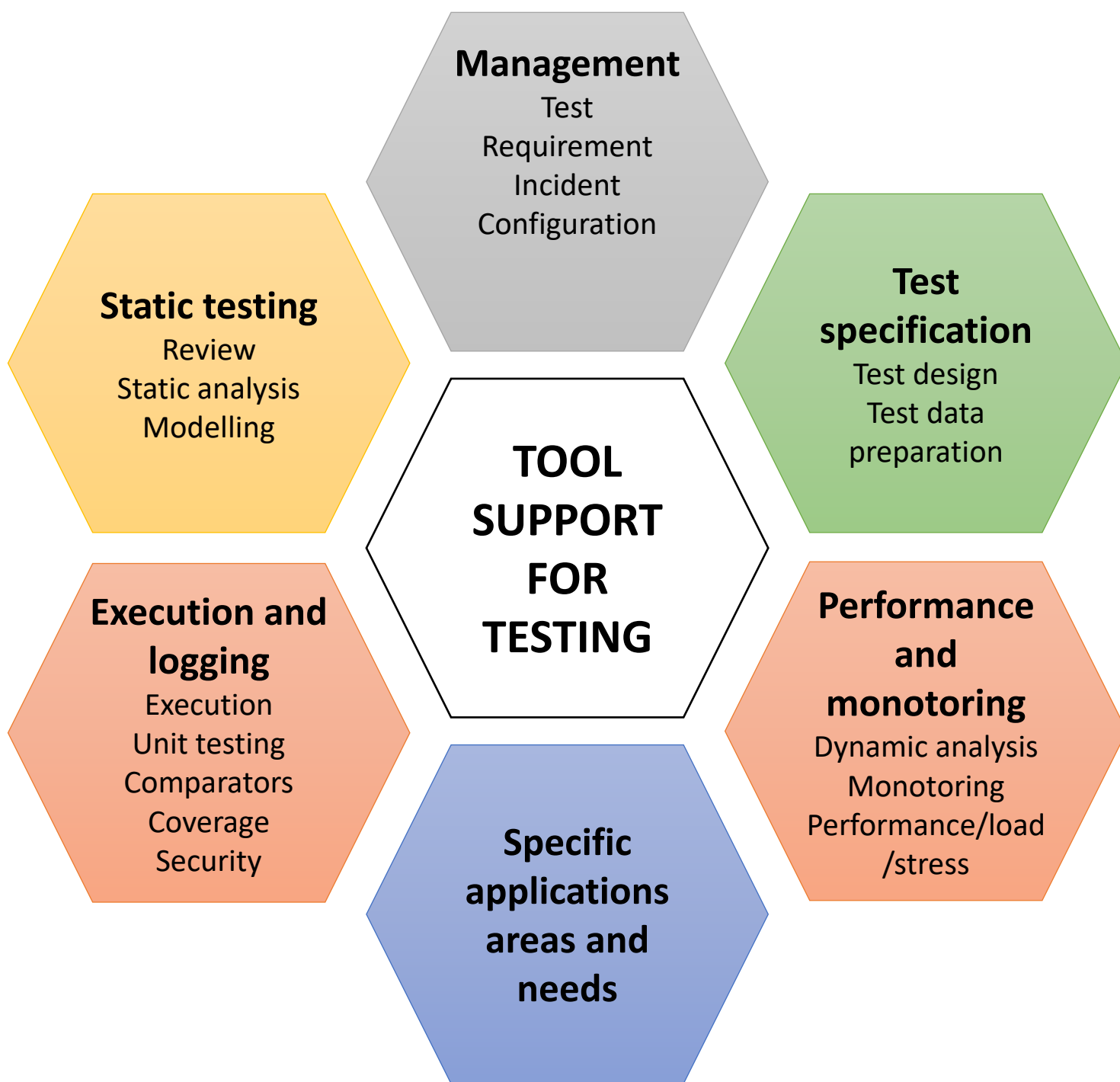
**Given** <some condition> , **When** <something is done>, **Then** <result should happen>

**As a** <role, e.g. costumer>

**In order to** <achieve something, e.g. product>

**I want** <do something>





# Tools support for **test execution & logging**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Test **execution** tools

- Enable tests to be executed automatically using stored
  - **inputs** and
  - **expected** results
- Coverage tools
- Test harnesses (D) and Unit test framework tools (D)
- Test comparators



# Tools support for **test execution & logging**

## 1. Types of test tools

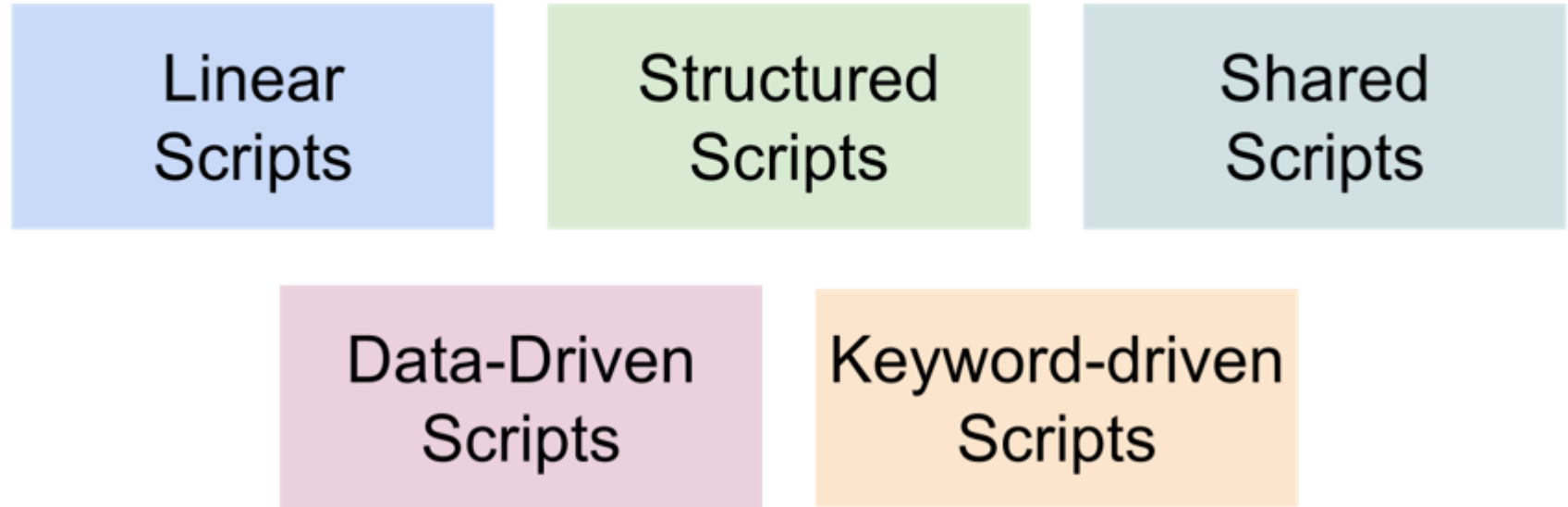
- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

### Levels of scripting



# Tools support for **test execution & logging**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Example: Simple Login Form

Test with **different** combinations of **username** and **password**

**Simple Login Form**

Username :  Password :

**Problem:** Necessary to write three scripts for three different combinations?

1. Go to login page
2. Type username "Hansen"
3. Type password "oslo123"
4. Click "Login" button

1. Go to login page
2. Type username "Olsen"
3. Type password "bergen456"
4. Click "Login" button

1. Go to login page
2. Type username "Jensen"
3. Type password "harstad789"
4. Click "Login" button



# Tools support for **test execution & logging**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

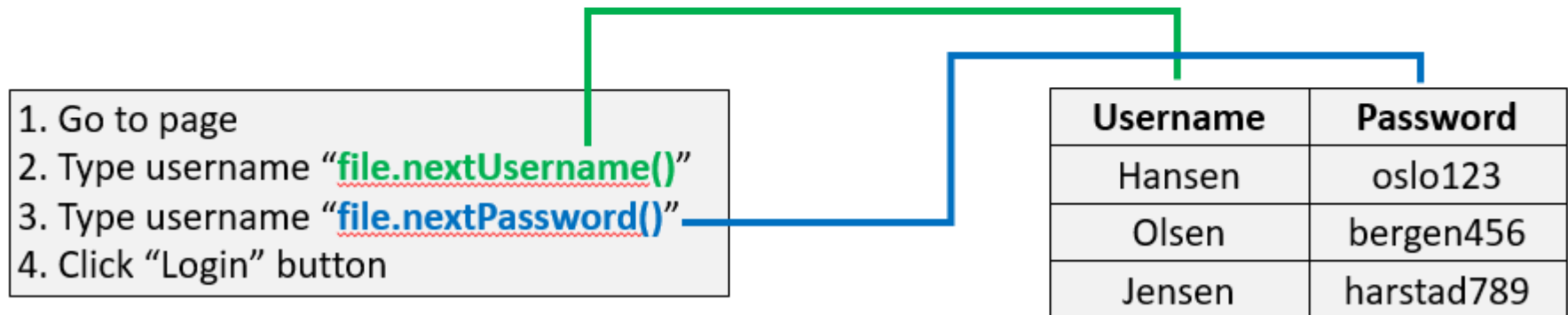
## 3. Introducing a test tool into an organization

### Example: Simple Login Form

This **test** approach is **time-consuming**

**Solution:** Separate test script from data (username, password) → No hard-coding

One script retrieves different combinations of username and password



# Tools support for **test execution & logging**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Keyword-driven scripting

**Keywords** symbolising **actions** (functionality)

“One level up” from data-driven scripting

Can write tests using keywords

“*What to test, rather than how to test it*”

Keyword	Script
Login	script1
<u>CH_password</u>	script2
Logout	script3

[script1]  
1. Go to page  
2. Type username “file.nextUsername()”  
3. Type username “file.nextPassword()”  
4. Click “Login” button

[script2]  
1. Click on user avatar  
2. Click “Change password”  
3. Type current password  
4. Type new password  
5. Click “Confirm” button

[script3]  
1. Click on user avatar  
2. Click “Logout” button





# Tools support for **test execution & logging**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

- Test execution tools

### **Coverage tools**

- **identifying** coverage **items** (instrumenting the code)
  - calculating the **percentage** of coverage items that were **exercised by** a suite of tests
  - reporting coverage items that have not yet been exercised
  - identifying test inputs to exercise items that have not yet been uncovered
  - generating stubs and drivers (if part of a unit test framework).
- Test harnesses and Unit test framework tools (D)
  - Test comparators

# Tools support for **test execution & logging**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

- Test execution tools
- Coverage tools

## Test **harnesses** and **Unit test** framework tools (D)

- supplying **inputs** to the software being tested
- receiving **outputs** generated by the software being tested
- **executing a set of tests** within the framework or using the test harness
- **recording the pass/fail** results of each test (framework tools)
- **storing tests** (framework tools)
- support for debugging (framework tools)
- **coverage measurement** at code level (framework tools)



# Tools support for test execution & logging

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Drivers

Calls the component to be tested

In other words: A component that calls the *Tested Unit*

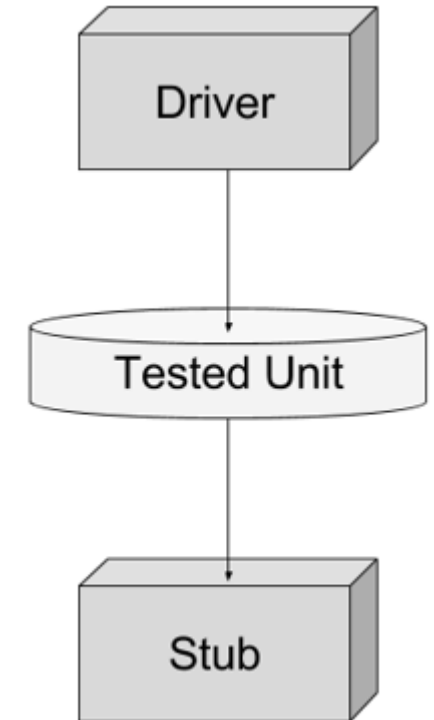
## Stubs

Called *from* the software component to be tested

In other words: A component the *Tested Unit* depends on

Partial implementation

Fake values



# Tools support for **test execution & logging**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Test **harness** and **unit test** framework tools (D)

The two **types** are **similar**

**Support** tools for testing **individual components** or software units

Harness: **Stubs** and **drivers** → Small programs that interact with software

Unit test framework tools → Support for object-oriented software

**When** are these tools used?

During test **execution** and **logging**



# Tools support for **test execution & logging**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Characteristics of test **harness** and **unit test** framework tools

Supply **inputs** to the software being tested

Receive **outputs** generated by the software being tested

**Execute** a **set of tests** within the framework

Record **pass** / **fail** results of each test

**Store** tests

**Coverage** measurement at code level

Provide **support** for debugging



# Tools support for **test execution & logging**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

- Test execution tools
- Coverage tools
- Test harnesses (D) and Unit test framework tools (D)
- **Test comparators**



# Tools support for **test execution & logging**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

Test **comparators** are used when the executed **test generates a lot of output**.

Testing is **more** than providing **inputs**

Need to **check** if software **produces** the **correct** result

**Compare actual** outcomes to **expected** results

Two ways of comparing results

**Dynamic** comparison → Comparison done **during** test execution

**Post-execution** comparison → Comparison performed **after** test has finished

Software under test is no longer executing



# Tools support for **test execution & logging**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

Test **comparators** are used when the executed **test** generates a lot of **output**.

### Dynamic comparison

Best done by test **execution tools**

**Useful** when actual results do not match expected results in the **middle** of a **test**

Tool may be **programmed** to take **recovery actions** / go to a **different** set of **tests**

### Example

Good for comparing wording of an error message

Does the pop-up message match the correct wording for that error message?



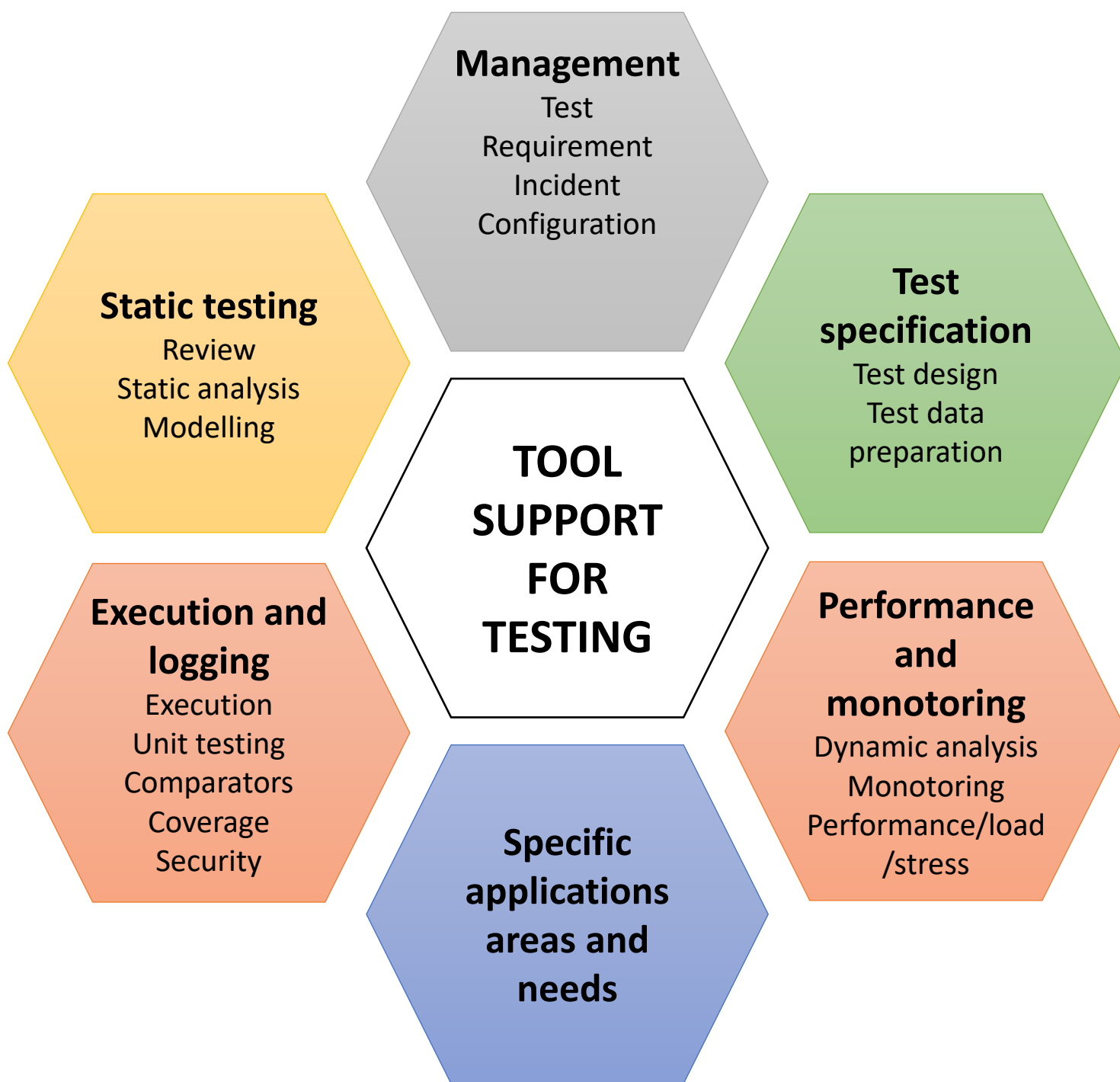


# Test comparators

Action	Type	Test Data		
		In	Expected	Actual
<input type="checkbox"/> objMoney = new Money(amount, curr...)	xY			
amount	int	9	[0..8]	9
currency	java.lang.String			
objMoney	junit.samples.money.Mo...			
<expected exception>	Throwable		<no exception>	<no exception>
<input type="checkbox"/> retValue = objMoney.equals(anObject)	xY			
anObject	java.lang.Object			
retValue	boolean			
<expected exception>	Throwable		<no exception>	<no exception>

Test Data Comparator loaded.





# Tools support for **performance & monitoring**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- **1.7 Tools for performance & monitoring**
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Test data for **performance** testing

### **Real** data

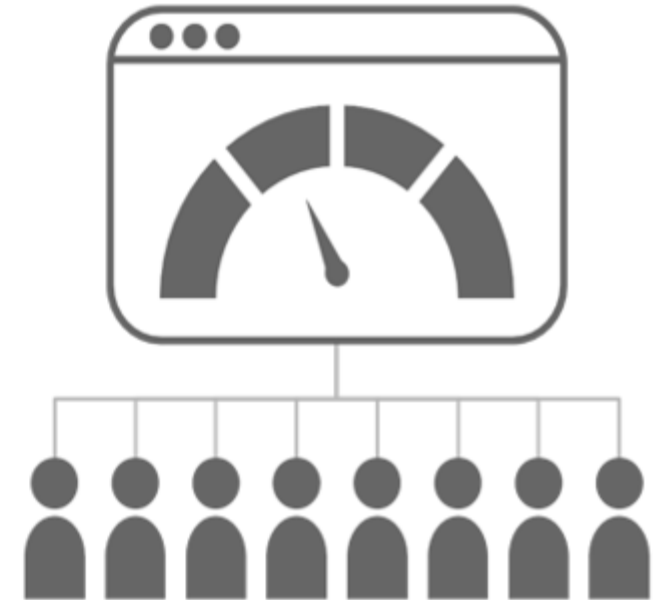
Test data obtained from **users**

### **Load**

Large amounts of test data can be produced

### **Maintenance**

Test data from the production environment



# Tools support for performance & monitoring

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

Tests should reflect realistic (correct) scenarios

Systems are often required to handle significant load / interactions

Inadequate / insufficient testing compromises system quality

Setting up test data → Significant effort

Extensive range or volume of data needed

Creating this data can be very resource-consuming

Test data preparation tools help us manage this effort



# Tools support for performance & monitoring

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Common features of test data preparation tools

Data can be **selected** from an **existing** database

Data can be **created**, **generated**, and **altered** for use in **tests**

Construct a **large number** of similar records → Volume tests

## When to use?

During test **specification** and **control** → Test data **management** is **difficult**

**Ensure** the system under **test** is being tested *realistically*

Useful for **performance** and **reliability** testing



# Monitoring tools

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

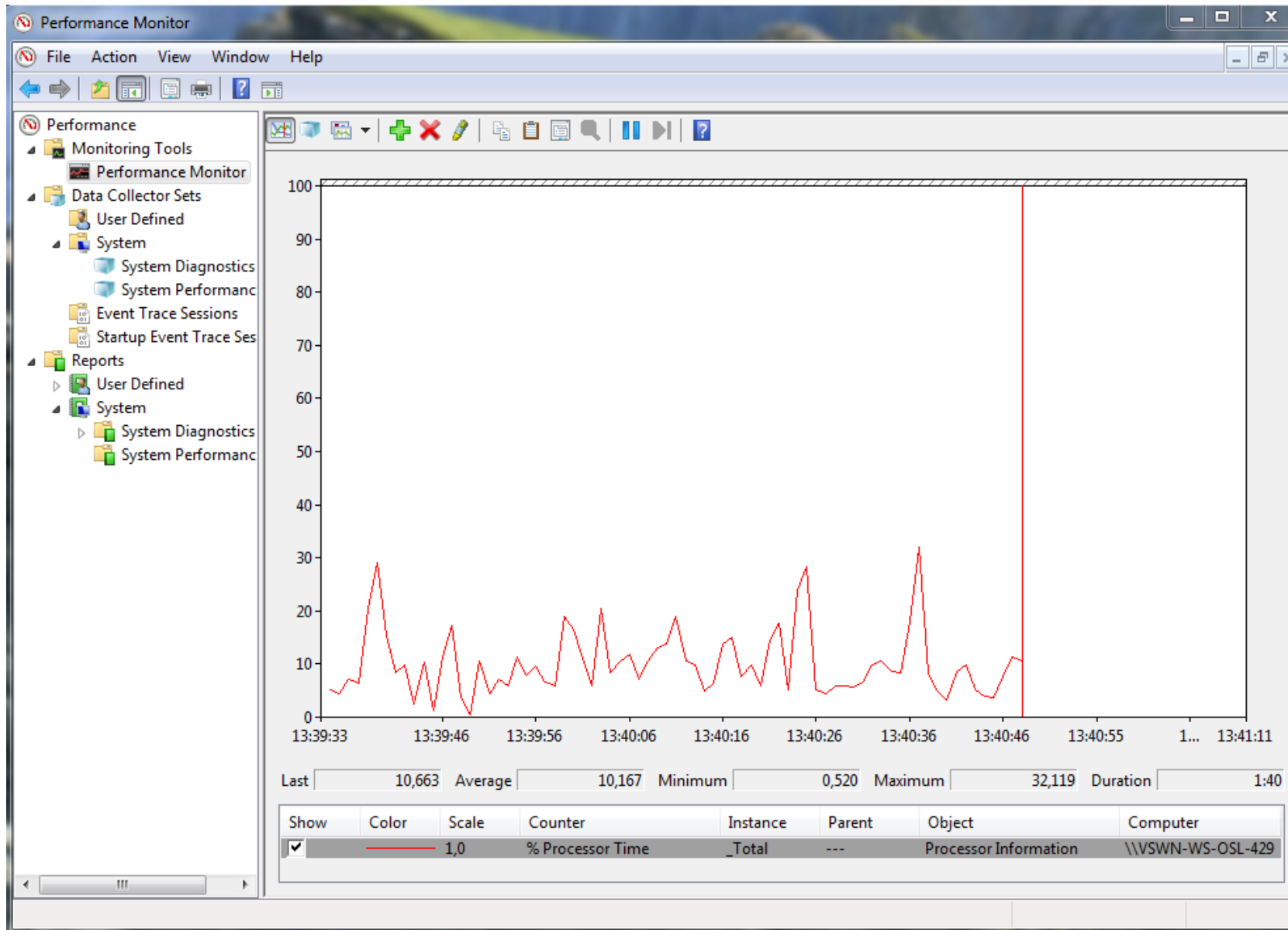
## 3. Introducing a test tool into an organization

**Monitoring** tools identifying problems and sending an alert message to the administrator (e.g. network administrator)

- logging **real-time** and **historical** information
- finding **optimal settings**
- monitoring the **number of users** on a network
- monitoring **network traffic** (either in real time or covering a given length of time of operation with the analysis performed afterwards).



# Monitoring tools



# Tools support for **performance & monitoring**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- **1.7 Tools for performance & monitoring**
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

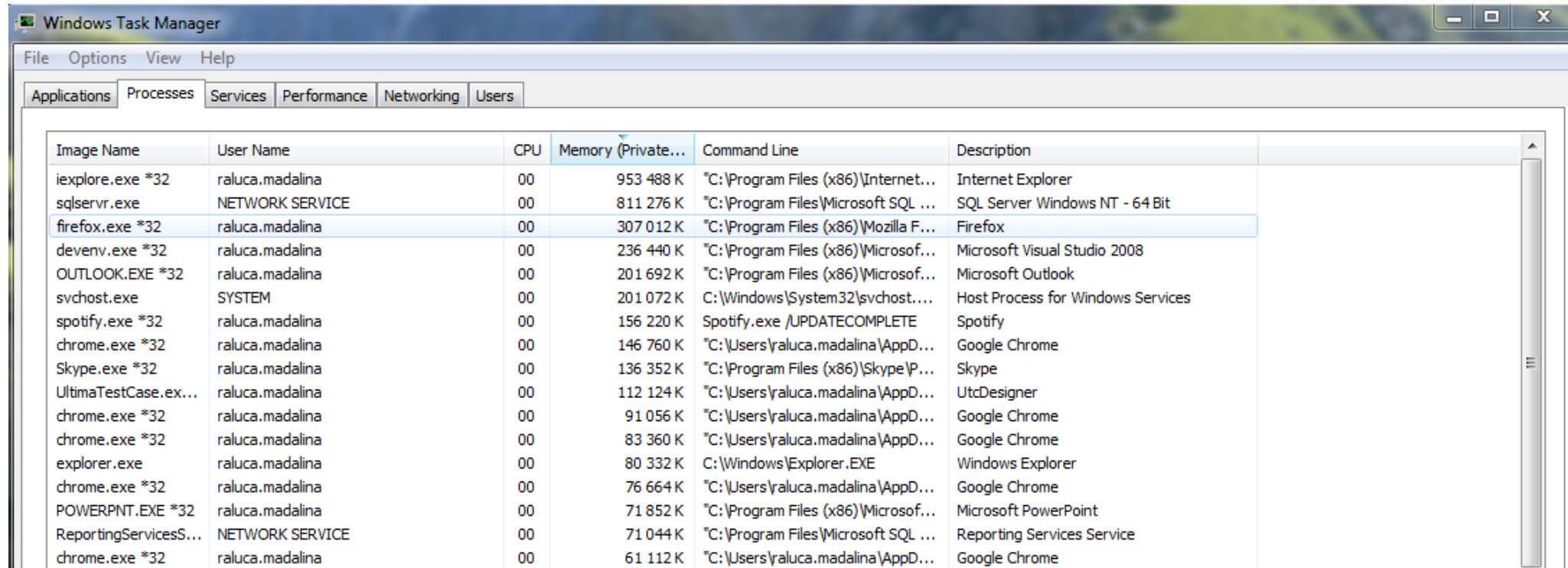
Features or characteristics of **dynamic analysis** tools include support for

- detecting **memory leaks**;
- identifying pointer arithmetic errors such as null pointers;
- identifying time **dependencies**





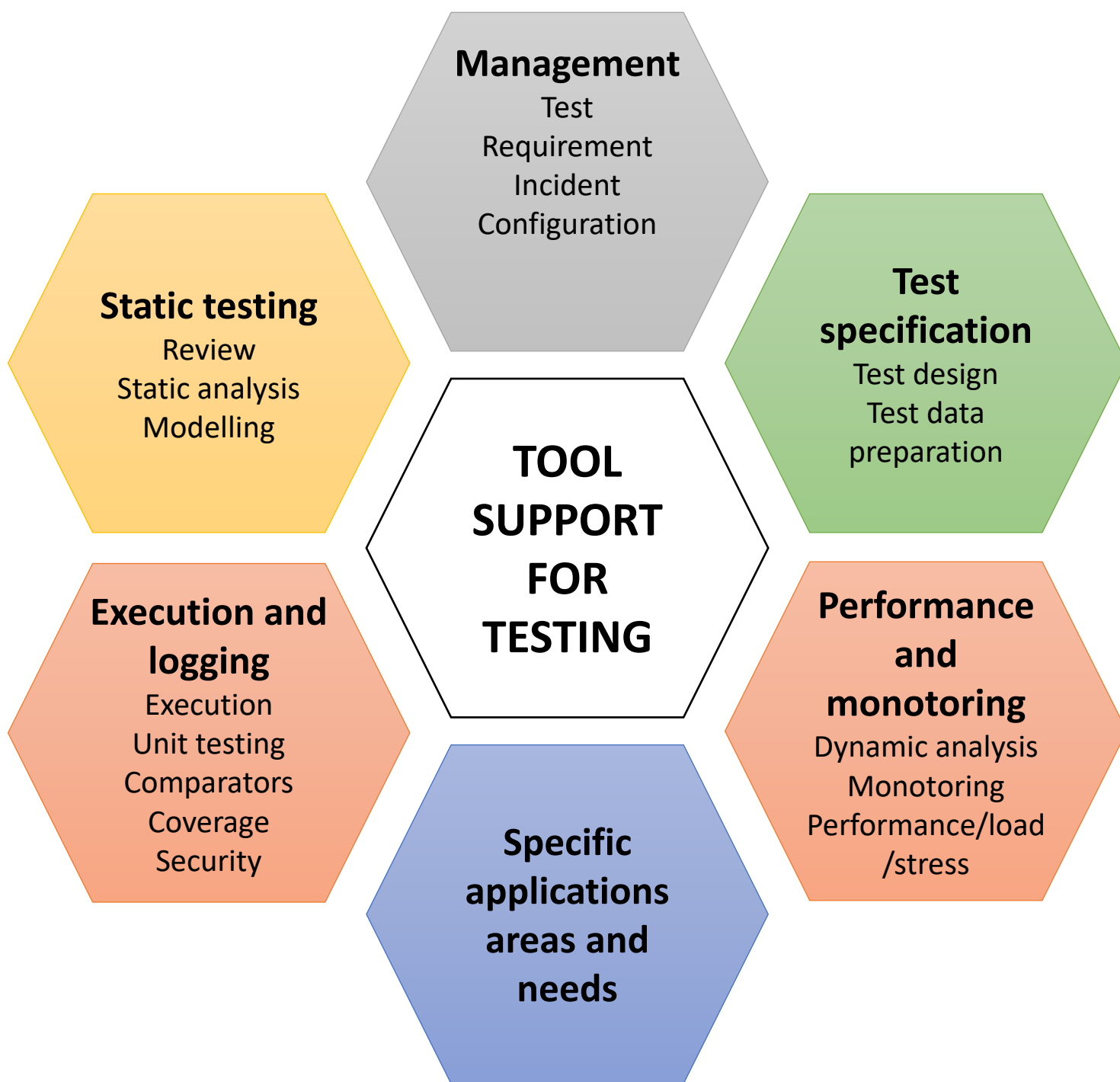
# Dynamic analysis tools (D)



The image shows a screenshot of the Windows Task Manager application. The 'Processes' tab is selected, displaying a list of running processes. The table below represents the data shown in the Task Manager window.

Image Name	User Name	CPU	Memory (Private...)	Command Line	Description
iexplore.exe *32	raluca.madalina	00	953 488 K	"C:\Program Files (x86)\Internet...	Internet Explorer
sqlservr.exe	NETWORK SERVICE	00	811 276 K	"C:\Program Files\Microsoft SQL ...	SQL Server Windows NT - 64 Bit
firefox.exe *32	raluca.madalina	00	307 012 K	"C:\Program Files (x86)\Mozilla F...	Firefox
devenv.exe *32	raluca.madalina	00	236 440 K	"C:\Program Files (x86)\Microsof...	Microsoft Visual Studio 2008
OUTLOOK.EXE *32	raluca.madalina	00	201 692 K	"C:\Program Files (x86)\Microsof...	Microsoft Outlook
svchost.exe	SYSTEM	00	201 072 K	C:\Windows\System32\svchost...	Host Process for Windows Services
spotify.exe *32	raluca.madalina	00	156 220 K	Spotify.exe /UPDATECOMPLETE	Spotify
chrome.exe *32	raluca.madalina	00	146 760 K	"C:\Users\raluca.madalina\AppData...	Google Chrome
Skype.exe *32	raluca.madalina	00	136 352 K	"C:\Program Files (x86)\Skype\P...	Skype
UltimaTestCase.ex...	raluca.madalina	00	112 124 K	"C:\Users\raluca.madalina\AppData...	UtcDesigner
chrome.exe *32	raluca.madalina	00	91 056 K	"C:\Users\raluca.madalina\AppData...	Google Chrome
chrome.exe *32	raluca.madalina	00	83 360 K	"C:\Users\raluca.madalina\AppData...	Google Chrome
explorer.exe	raluca.madalina	00	80 332 K	C:\Windows\Explorer.EXE	Windows Explorer
chrome.exe *32	raluca.madalina	00	76 664 K	"C:\Users\raluca.madalina\AppData...	Google Chrome
POWERPNT.EXE *32	raluca.madalina	00	71 852 K	"C:\Program Files (x86)\Microsof...	Microsoft PowerPoint
ReportingServicesS...	NETWORK SERVICE	00	71 044 K	"C:\Program Files\Microsoft SQL ...	Reporting Services Service
chrome.exe *32	raluca.madalina	00	61 112 K	"C:\Users\raluca.madalina\AppData...	Google Chrome





# Tools support for **specific application** areas

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

There are tools **specialized** for use in a **particular type of application**

- Data **quality** assessment
- Data **conversion** and **migration**
- **Usability** testing and **Accessibility** testing
- **Localization** testing
- **Security** testing
- **Portability** testing



# Tools support for **specific application** areas

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

- There are tools **specialized** for use in a **particular type of application**.

Example:

performance testing tools specifically for web-based applications

dynamic analysis tools specifically for testing security aspects.

- Example of targeted areas: embedded systems.



# Tools support for **specific application** areas

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

### • Testers may use:

- word processor
- spreadsheets

as a testing tool, but they are often used to store:

- test designs
- test scripts
- test data.

- Testers may also use **SQL** to set up and **query databases** containing test data.
- Tools used by developers when **debugging**, to help localize defects and check their fixes, are also testing tools.
- It is a good idea to look at **any type of tool available** to you for ways it could be used to help **support any of the testing activities**.



# Effective use of test tools

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

**✓ LO: Summarize the potential benefits of using test tools in the software life-cycle**

**✓ LO: Summarize the potential risks of using test tools in the software life-cycle**

**✓ LO: Summarize the potential risks of test automation in the software life-cycle**

**✓ LO: Remember the special considerations for test execution tools, static analysis tools and test management tools**

# Potential benefits and risks

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 **Potential benefits and risks of tools**
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

- Simply purchasing or leasing a tool **does not guarantee success** with that tool!
- Each type of tool may **require additional effort** to achieve real and lasting benefits.



# Potential benefits and risks

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 **Potential benefits and risks of tools**
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Greater consistency and repeatability

People tend to do the same tasks in a slightly different way

Distractions affect human performance

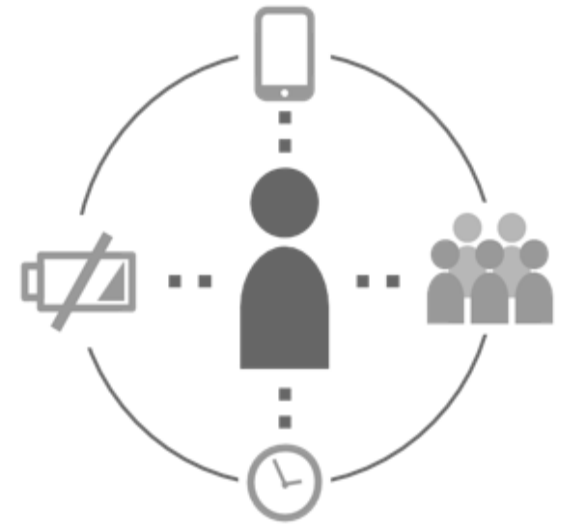
Doing more than one task simultaneously

Interruptions by peers / co-workers

Fatigue and personal issues

External pressures

Tools will reproduce the exact same procedure as previously





# Potential benefits and risks

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

## Objective assessment

Humans are prone to make errors

Subjective preconceived notions and bias toward verification

Testing tools on the other hand ...

Objective “preconceived notions”

Assessment → Repeatable and consistently calculated

Cyclomatic complexity, nesting levels

Coverage, system behaviour, incident statistics



# Potential benefits and risks

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

**Ease of access to information** about the tests or test effort

Information presented **visually**

**Easier** for the **human** mind to **understand**

Chart, graphs > Long list of numbers

**Special purpose** tools provide **features directly**

Statistics and graphs

Incident rates

Performance



# Potential benefits and risks

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- **2.1 Potential benefits and risks of tools**
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

### Potential benefits of using tools :

Reduced repetitive work ( running regression tests, re-entering the same test data. Etc)

Greater consistency and repeatability ( tests executed by a tool, tests derived from requirements).

Objective assessment (static measures, coverage).

Ease of access to information about tests or testing (statistics / graphs about test progress, incident rates, performance)



# Potential benefits and risks

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- **2.1 Potential benefits and risks of tools**
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

### Potential risks of using tools:

Unrealistic expectations for the tool (functionality & ease of use).

Underestimating time, cost and effort for the introduction of a tool (training, external expertise).

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

Underestimating the effort required to maintain the test assets generated by the tool.

Over-reliance on the tool (replacement where manual testing would be better).



# Potential benefits and risks

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

### Tools are not magic!

They can do very well what they have been **designed to do**, but they **cannot do everything**.

The tester concentrates on

- **what** should be tested
- what the **test cases** should be
- how to **prioritize** the testing

The tool user concentrates on

- **how** best to get **the tool to do its job** effectively
- **how** to give **increasing benefit** from tool use



# Special considerations: **Test execution tools**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 **Special considerations for tools**

## 3. Introducing a test tool into an organization

This type of tool often **requires significant effort** in order to achieve significant benefits.

- Capturing tests by recording the actions of a manual tester seems attractive, but **this approach does not scale to large numbers of automated tests**. This type of script may be unstable when unexpected events occur.
- **Data-driven approach**: **separates out the test inputs** (the data) and uses a more **generic script** that can read the test data and perform the **same test with different data**.
- **In a keyword-driven approach**: the spreadsheet contains **keywords** with the **actions** to be taken (also called action words), and test data. Testers can then **define tests using the keywords**.



# Special considerations: Performance testing tools

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

- The **design of the load** to be generated by the tool
- **Timing** aspects – probe effect
- **How** to interpret the information gathered.
- These tools need **tester with expertise** in performance testing to **design the tests** and **interpret results**.



# Special considerations: **Static analysis tools**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 **Special considerations for tools**

## 3. Introducing a test tool into an organization

- There is a risk that the changes to make **old code** to conform to **new standard** will introduce an **unexpected side-effect**.
- These tools applied to source code can enforce coding standards, but **if applied to existing code** may generate **a lot of messages**.
- A gradual **implementation with initial filters** to exclude some messages would be an effective approach.





# Special considerations: **Test management tools**

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 **Special considerations for tools**

## 3. Introducing a test tool into an organization

- They need to **interface with other tools** or spreadsheets in order to produce information in the best format for the current needs of the organization.



# Introducing a test tool into an organization

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

**✓ LO: State the main considerations for introducing a new test tool to an organization**

**✓ LO: State the goals of a proof-of-concept for a test tool, with the scope of evaluation and pilot**

**✓ LO: Explain the success factors for the deployment of a new test tool into an organization**

# Introducing a test tool into an organization

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

The ***main considerations*** in selecting a tool for an organization include:

- **Assess** the organizational **maturity**, strengths and weaknesses
- **Evaluate** against **clear requirements** and **objective** criteria.
- A **proof-of-concept** to test the required functionality and determine whether the product **meets its objectives**.
- **Evaluation** of the **vendor** (including training, support and commercial aspects).
- Identification of **internal requirements** for **coaching** and **mentoring** in the use of the tool.



# Introducing a test tool into an organization

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

Introducing the selected tool into an organization starts with a ***pilot project***, ***with the following objectives***:

- **Learn** more detail about the tool.
- **Evaluate** how the tool **fits with existing processes** and **practices**, and determine what would need to change.
- **Decide** on standard ways of **using** and **maintaining** the tool and the test.
- **Assess** whether the **benefits** will be achieved at reasonable **cost**.



# Introducing a test tool into an organization

## 1. Types of test tools

- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test design and specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

## 2. Effective use of test tools

- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

## 3. Introducing a test tool into an organization

**Success factors** for the deployment of the tool within an organization include:

- **Roll out** the tool to the rest of the organization **incrementally**.
- **Adapt** and **improve processes** to fit with the use of the tool.
- Provide **training** and **coaching**/mentoring for new users.
- **Define** usage **guidelines**.
- Implement a way to **learn lessons** from tool use.
- **Monitor** the tool **use** and **benefits**.

