

Tool support for testing

Chapter 6

- 1. Types of test tools**
- 2. Effective use of test tools: potential benefits and risks**
- 3. 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 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: Explain the term “test tool”**

✓ **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 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** (i.e: 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 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 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**. (i.e. 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 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**.
- 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 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

Requirements management tools

store requirements

check for consistency and undefined (missing) requirements

allow prioritization

enable individual tests to be traceable to requirements

Requirements management tools

The screenshot displays the TestLink 1.8.5 web interface in a Mozilla Firefox browser. The browser's address bar shows the URL `https://newhope.lanranet.jamk.fi/ProjectTESTLINK/`. The page header includes the 'nest project platform' logo and navigation tabs for 'Home', 'Knowledge Sharing', 'Communication', 'Work Collaboration', and 'Administration'. Below this, there are more specific navigation options like 'Tickets & Tasking', 'Project Management', 'Error Management', 'Change Management', 'Test Management', and 'SCM Statistics'. The user is logged in as 'Admin User'.

The main content area is titled 'Test Case' and shows details for 'TC_-2:Resetting calory counter from UI'. A message states: 'You can not edit this version because it has been executed'. The test case details include:

- Version 1**: Created on 10/11/2010 19:34:01 by AdminUser
- Summary**: Wristband + Vibrator/bare hands
- Steps**:
 - Open Calory Meter tab
 - Select "Reset Calory Meter"
 - Check value on screen (should be 0 cal/h)
 - Activate Play mode and use some energy or use vibrator for 20 seconds
 - check calory counter value
- Expected Results**:
 - PASS Criteria: Value are counting from zero after reset
 - FAIL criteria: Values are not resetting correctly
- Execution type**: Manual
- Test importance**: Medium
- Keywords**: None
- Requirements**: [IFDK System Requirements] UserStoryId5003:UserStoryId5003
- Test Plan usage**:

Version	Test Plan
1	EXAMPLE TEST PLAN v0.1
1	IFDK System Test Plan 0.1
1	GroupCancerAxe
1	Team2
1	FarEasternFirePig
- Attached files**: (empty)

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 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

Incident management tools

store and manage incident reports

support management of incident reports

- Statuses of incident reports:

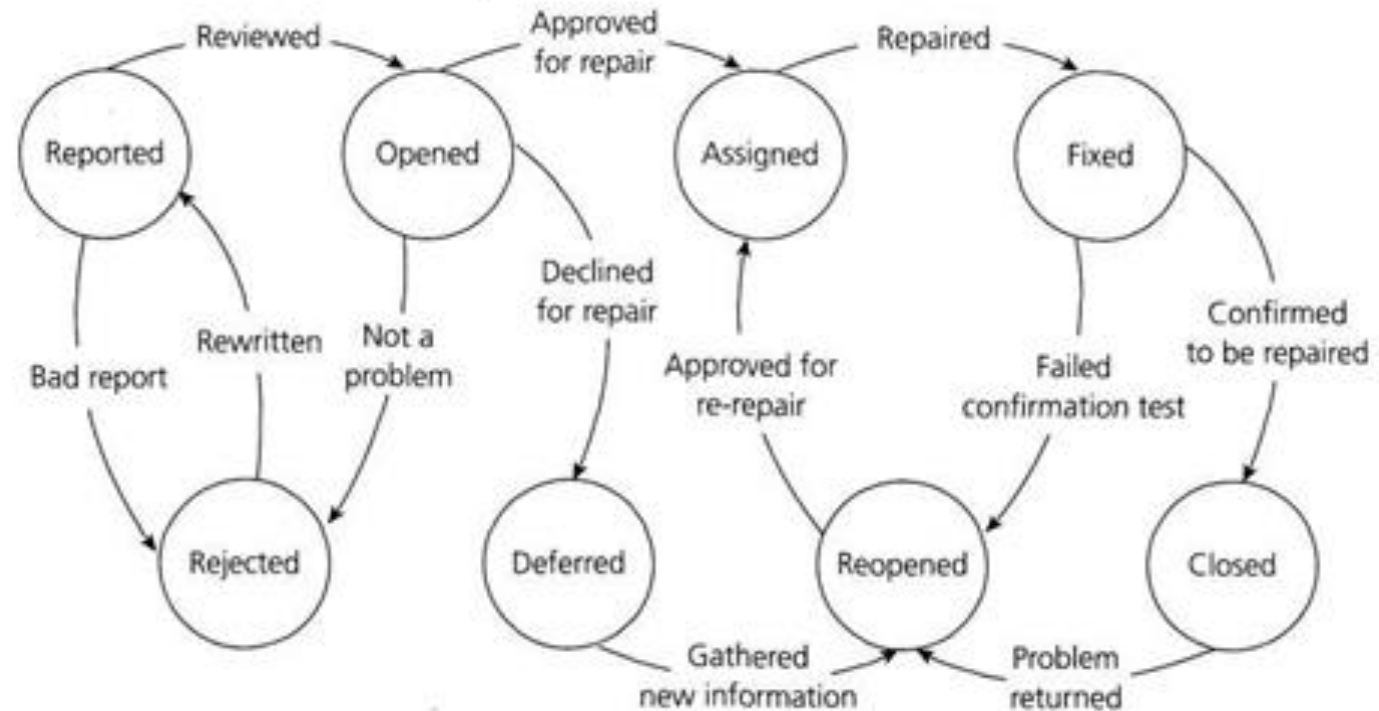


FIGURE 5.3 Incident report life cycle

Incident management tools

The screenshot shows a web browser window displaying a Bugzilla bug report. The browser's title bar reads "Bug 305134 - Remove FeedView from Firefox 1.5 - Mozilla Firefox 1.0+ (Build 2005082813) - Gecko 1.8b4". The address bar shows the URL "https://bugzilla.mozilla.org/show_bug.cgi?id=305134".

The Bugzilla header includes the logo and version "Bugzilla Version 2.19.1+". The main heading is "Bugzilla Bug 305134" with a red "Description" link and the text "Remove FeedView from Firefox 1.5". It also shows the last modified date: "Last modified: 2005-08-28 01:41 PDT".

The bug details form includes the following fields:

- Bug#:** 305134
- alias:** (empty)
- Hardware:** All
- Product:** Firefox
- OS:** All
- Component:** RSS Discovery and Preview
- Version:** unspecified
- Status:** RESOLVED
- Priority:** -
- Resolution:** FIXED
- Severity:** normal
- Assigned To:** Ben Goodger (use ben at mozilla dot org for email) <bugs@bengoodger.com>
- Target Milestone:** -
- QA Contact:** nobody@mozilla.org
- URL:** (empty)
- Summary:** Remove FeedView from Firefox 1.5
- Status Whiteboard:** (empty)
- Keywords:** fixed1.8

The **Reporter:** is Ben Goodger (use ben dot org for email) <bugs@bengoodger.c>. The **CC:** list includes alex@spamcop.net, axel@pike.org, bugs.mano@sent.com, bugtrap@psychoticwol, and bugzilla@dougweb.org. There is a checkbox for "Remove selected C".

The **Flags:** section includes a "(Help!)" link and a list of flags: mtschrep: blocking1.8b4rc, bugs: blocking1.8b4, blocking1.9a1, blocking-aviary1.0.7, blocking-aviary2.0, and testcase.

At the bottom, there is a table with columns: Attachment, Type, Created, Size, Flags, and Actions. The table is currently empty.

The browser's status bar at the bottom shows "Terminé" on the left and "bugzilla.mozilla.org" and "Adblock" on the right.

Incident management tools

The screenshot displays the Microsoft Team System Web Access interface within a Windows Internet Explorer browser. The browser's address bar shows the URL `http://tfsrtnsp1:8090/index.aspx?pid=7`. The page title is "Adventure Works - Microsoft Team System Web Access".

The interface features a navigation bar with tabs for "Home", "Work Items", "Reports", "Documents", "Source", and "Build". The "Home" tab is active, and a "New Work Item" dropdown menu is open, listing options: Bug, Task, Issue, Change Request, Risk, Requirement, and Review. Other options visible in the main area include Reports, Documents, and Source Control.

The left sidebar contains several sections:

- Favorites:** Active Bugs, Development Tasks
- New Work Item:** Bug, Task, Change Request, Requirement, Risk
- My Queries:** No recent items.
- Team Queries:** Active Bugs, Development Tasks, All Work Items, All My Team Project Work Items, All Tasks

The main content area displays "WI's Assigned to Me" with a [Refresh] button and a (Customize) link. It shows a summary of work items:

- 4 Bug (1 Active, 3 Resolved)
- 2 Task (2 Active)
- 1 Risk (1 Active)
- 2 Requirement (2 Proposed)
- 1 Change Request (1 Active)

Below this, the "Recently Accessed Work Items" section lists a table of items:

ID	Type	Status	Description
41	Task	Active	Implement support for Integrated authentic...
40	Change Request	Active	Support Integrated authentication
39	Requirement	Proposed	Application should work with limited disk sp...
38	Requirement	Proposed	About page displays the version information
37	Risk	Active	Temporary files can fill up the disk

The bottom status bar shows "Done" on the left and "Local intranet" and "100%" on the right.

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 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

Configuration management tools

are necessary to keep track of different versions and builds of the SW and tests

are particularly useful when developing on more than one configuration of the HW/SW environment

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 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

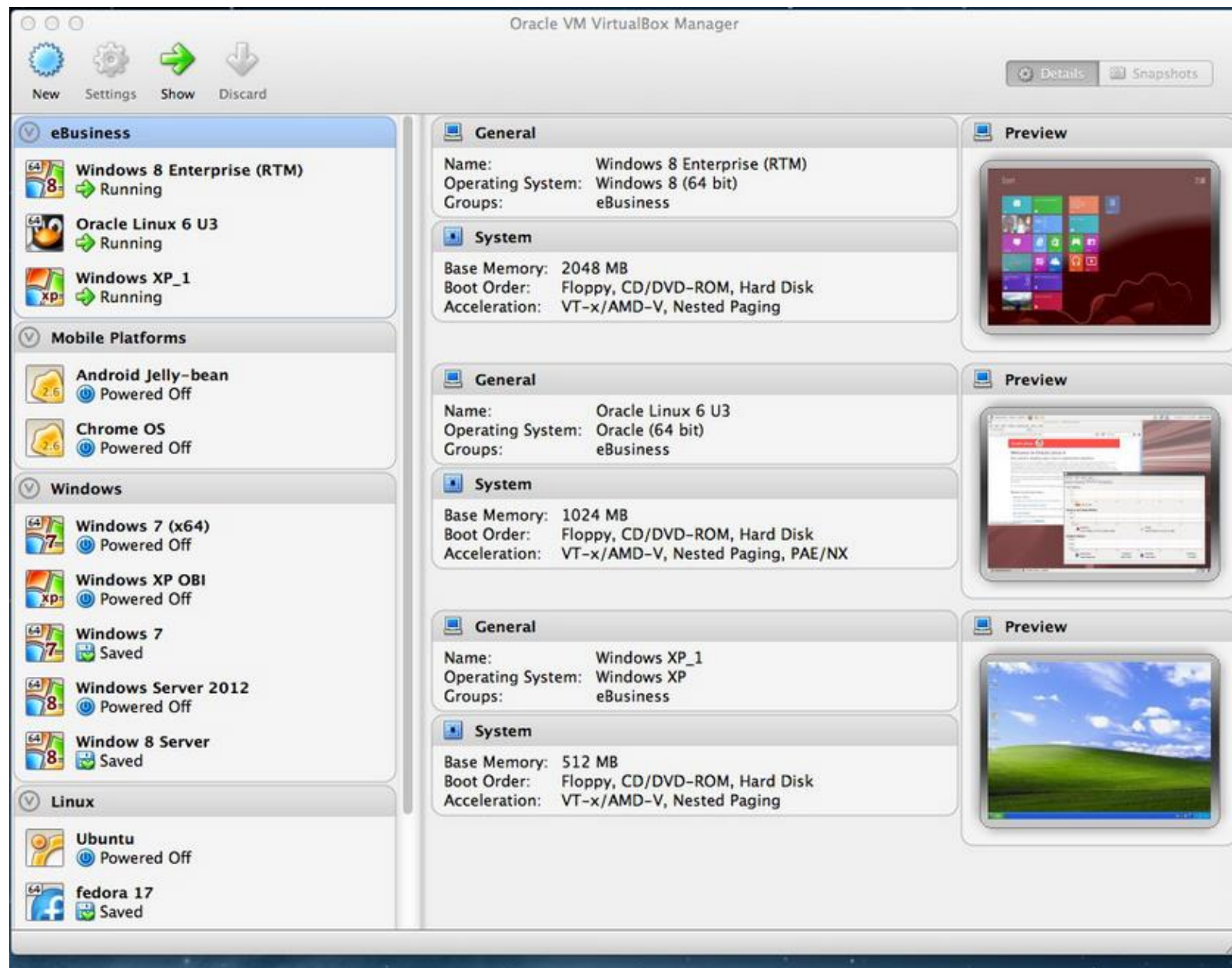
Configuration management tools

History

[\[edit\]](#)

Configuration Management (CM) as a formal management approach was developed by the [USAF](#) for the [DoD](#) in the 1950s as a technical management discipline for hardware material items—and it is now a standard practice in virtually every industry. The CM process became its own technical discipline sometime in the late 1960s when the DoD developed a series of [military standards](#) called the "480 series" (i.e., MIL-STD-480 and MIL-STD-481) that were subsequently issued in the 1970s. In 1991, the "480 series" was consolidated into a single standard known as the MIL-STD-973 that was then replaced by MIL-HDBK-61 pursuant to a general DoD goal that reduced the number of military standards in favor of industry [technical standards](#) supported by [Standards Developing Organizations](#) (SDO).^[8] This marked the beginning of what has now evolved into the most widely distributed and accepted standard on CM, ANSI-EIA-649-1998.^[9] Now widely adopted by numerous organizations and agencies, the CM discipline's concepts include [systems engineering](#) (SE), [integrated logistics support](#) (ILS), [Capability Maturity Model Integration](#) (CMMI), [ISO 9000](#), [Prince2](#) project management methodology, [COBIT](#), [Information Technology Infrastructure Library](#) (ITIL), [product lifecycle management](#), and [application lifecycle management](#). Many of these functions and models have redefined CM from its traditional holistic approach to technical management. Some treat CM as being similar to a librarian activity, and break out change control or [change management](#) as a separate or stand alone discipline.

Configuration management tools



Tools support for static testing

1. Types of test tools

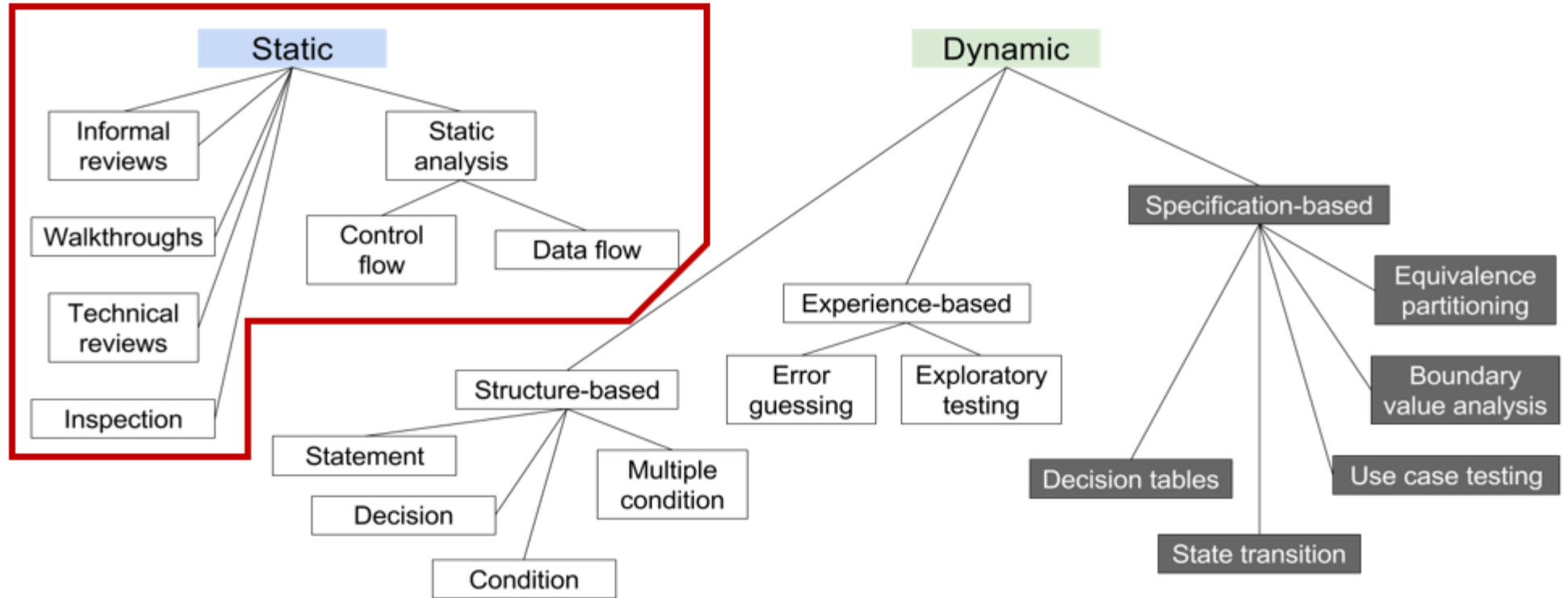
- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- **1.4 Tools for static testing**
- 1.5 Tools for test specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

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 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

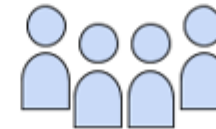
Tools for static testing

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

Categories

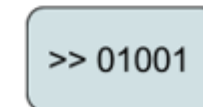
Review tools

Supports the review process



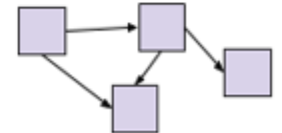
Static analysis tools

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 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 tools

store information about review processes

store and communicate review comments, report on defects and effort

They can provide aid for online reviews, which is useful if the team is geographically dispersed.

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 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

Review tools

The screenshot shows the Review Board 1.6.3 dashboard. The browser address bar is `reviews.reviewboard.org/dashboard/`. The page title is "Review Board 1.6.3". The navigation menu includes "My Dashboard", "New Review Request", "All review requests", "Groups", and "Submitters". A sidebar on the left shows navigation options: "Starred Reviews" (0), "Outgoing Reviews" (0), "Incoming Reviews" (0), "To Me" (0), "reviewboard" (62), and "All My Requests" (0). The main content area is titled "All Incoming Review Requests" and contains a table with the following data:

Summary	Submitter	Posted	Last Updated
Response to feature 2282	wilsonyeung	January 28th, 2012, 3:21 p.m.	12 hours ago
Added issue tracking integration for drafts and review requests.	ammok	February 11th, 2012, 9:12 p.m.	1 day ago
Added Show Links for Interdiffs	cim1	January 21st, 2012, 2:22 p.m.	1 day, 1 hour ago
Added * flag to required review request fields	medanat	February 13th, 2012, 5:43 p.m.	1 day, 1 hour ago
WebHooks Extension [WIP]	smacleod	February 10th, 2012, 6:28 p.m.	1 day, 12 hours ago
Navigation and action hooks missing extension parameter	bartek	February 1st, 2012, 3:49 a.m.	2 days, 18 hours ago
Created a new script for generating scaffolding for new extensions.	ammok	January 28th, 2012, 5:05 p.m.	3 days, 17 hours ago
Reviewboard-Social Extension	cim1	February 4th, 2012, 8:36 a.m.	6 days, 8 hours ago
Switch to using middleware for initialization.	chipx86	January 30th, 2012, 3:09 a.m.	1 week, 2 days ago
Remove all usage of ifuserorperm/ifnotuserandperm.	chipx86	January 30th, 2012, 3:37 a.m.	1 week, 2 days ago
Added support for authentication against Active Directory subdomains.	aineko	January 31st, 2012, 5:48 a.m.	2 weeks ago
Use Django's smarter if tag.	chipx86	January 30th, 2012, 3:11 a.m.	2 weeks, 1 day ago
Add optional enable_highlighting parameter in get_file_chunks_in_range	kamlani	April 5th, 2011, 8:16 p.m.	2 weeks, 3 days ago
Localized timezone support (Review Board)	ddruska	December 1st, 2011, 4:01 p.m.	2 weeks, 6 days ago
Fix authentication issue with perforce repository	stid	December 24th, 2011, 5:34 a.m.	1 month, 3 weeks ago

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 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)

Major purpose:

- The enforcement of coding standards.
- The analysis of structures and dependencies (e.g. linked webpages)
- Aiding in understanding the code.

support developers, testers and quality assurers in finding defects before dynamic testing.

Static analysis tools can calculate metrics from the code (e.g. complexity), which can give valuable information for planning or risk analysis.

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 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

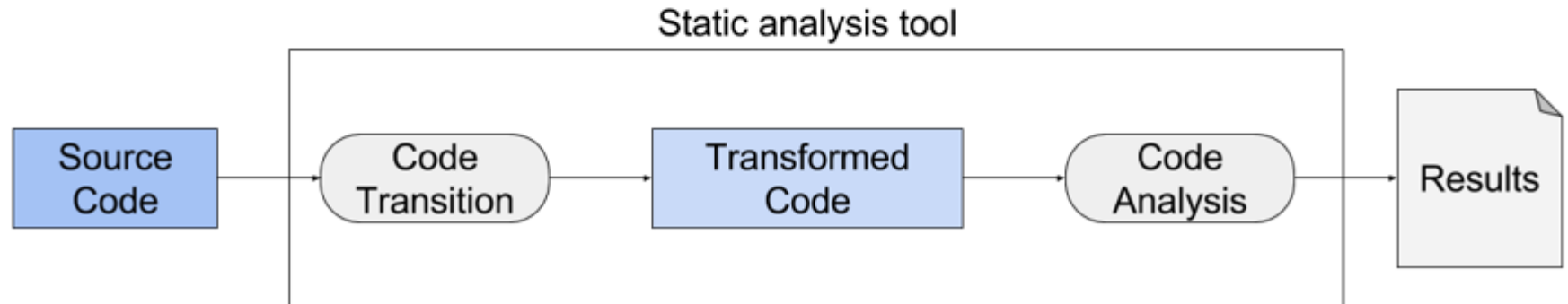
Static analysis tools (D)

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 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**

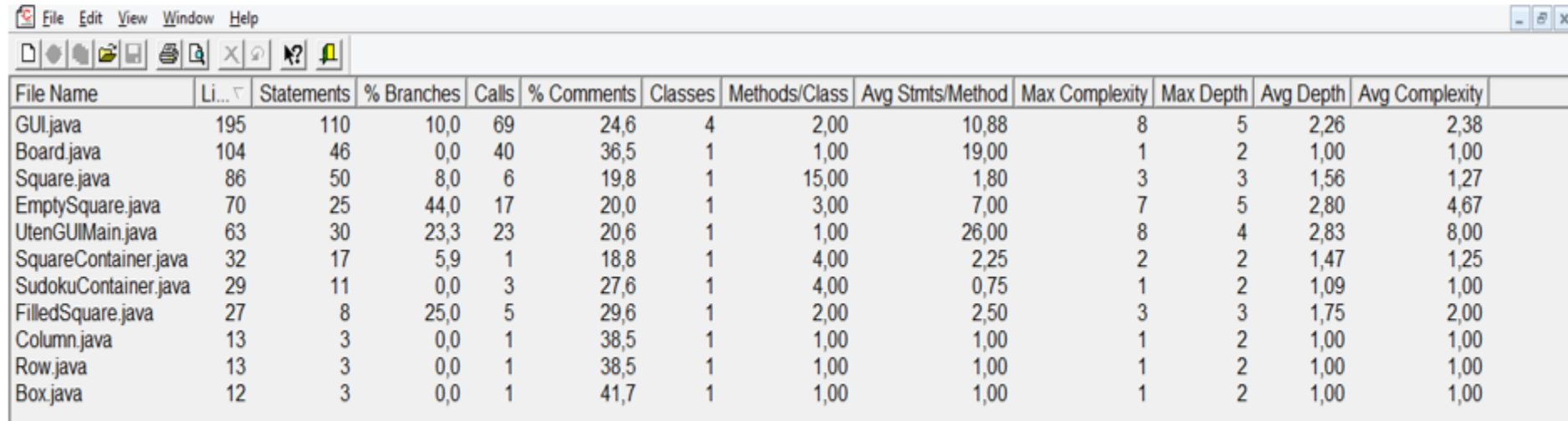
Analyse code **structures** and **dependencies**

Static analysis tools (D)

Static analysis tool example: Source Monitor

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

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



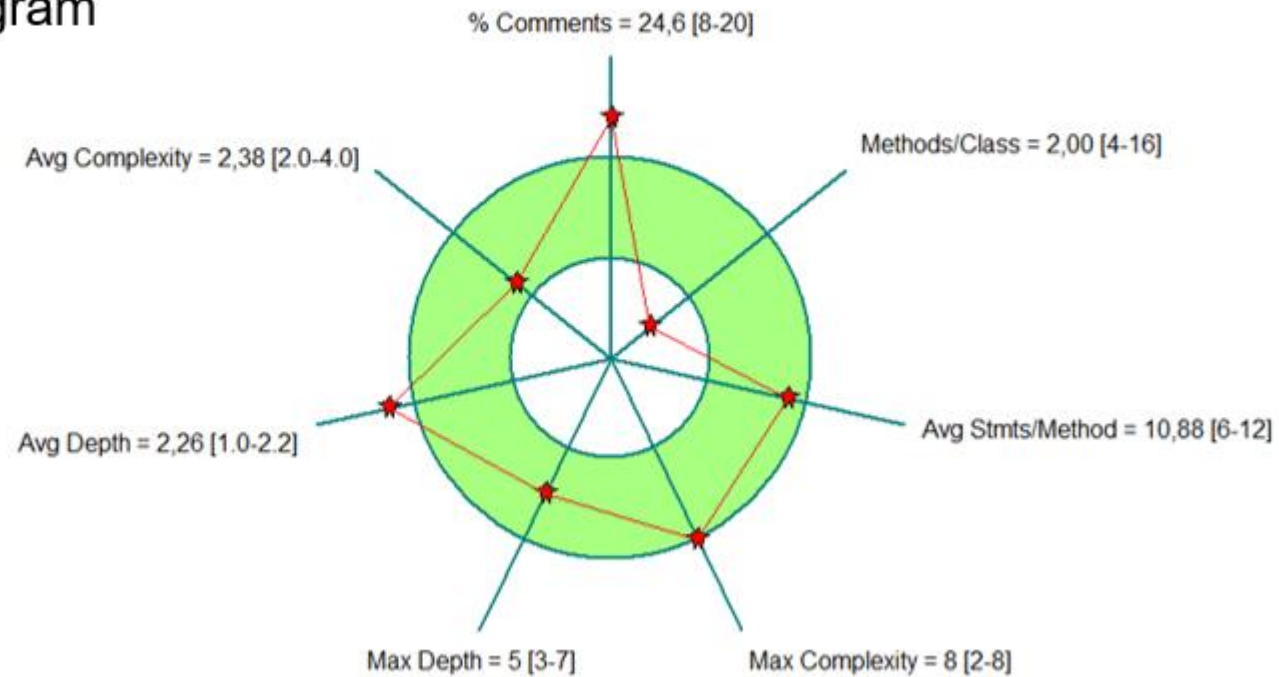
The screenshot shows a window titled "Source Monitor" with a menu bar (File, Edit, View, Window, Help) and a toolbar. Below the toolbar is a table with 13 columns: File Name, Li..., Statements, % Branches, Calls, % Comments, Classes, Methods/Class, Avg Stmts/Method, Max Complexity, Max Depth, Avg Depth,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

File Name	Li...	Statements	% Branches	Calls	% Comments	Classes	Methods/Class	Avg Stmts/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

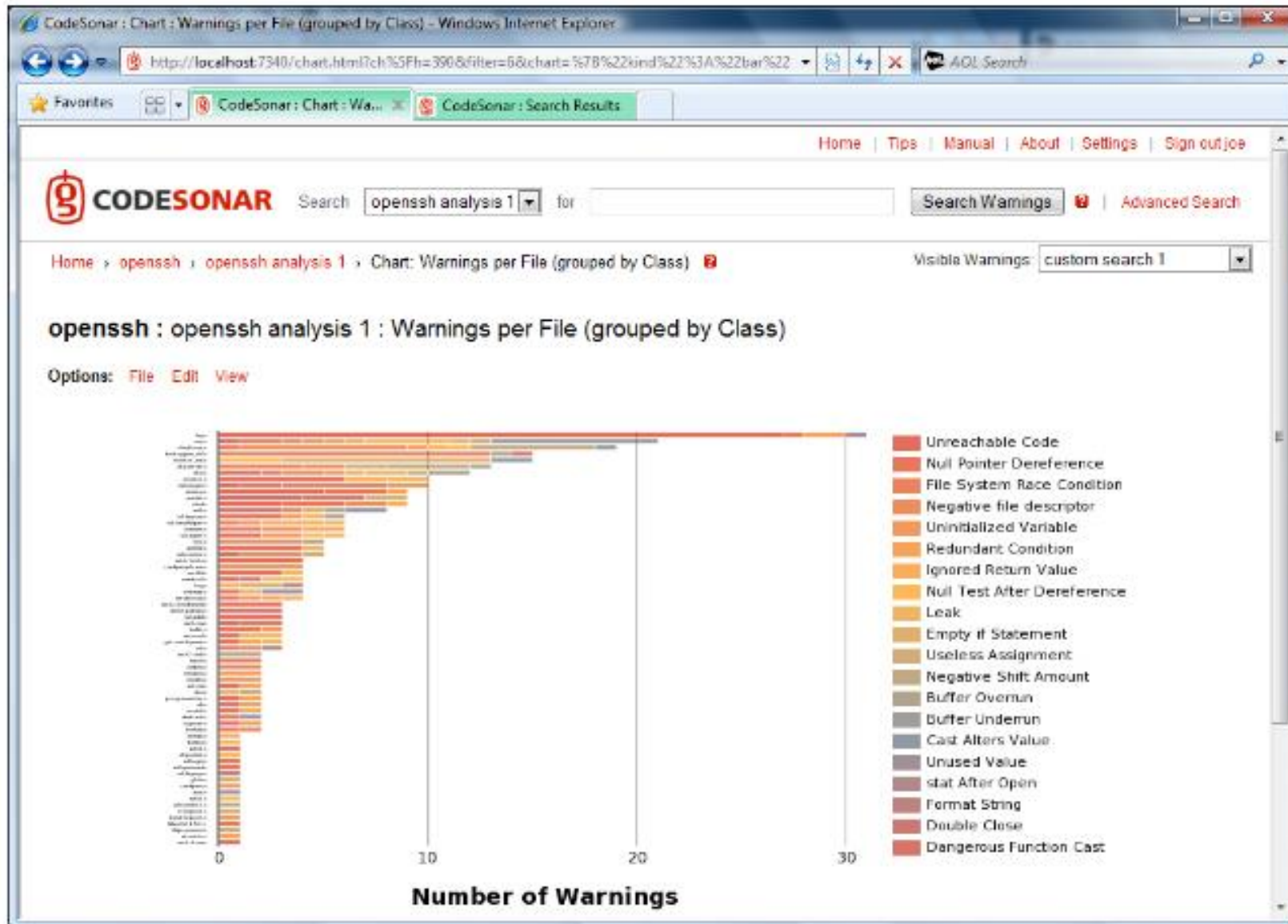
Static analysis tools (D)

Static analysis tool example: Source Monitor

Kiviati diagram

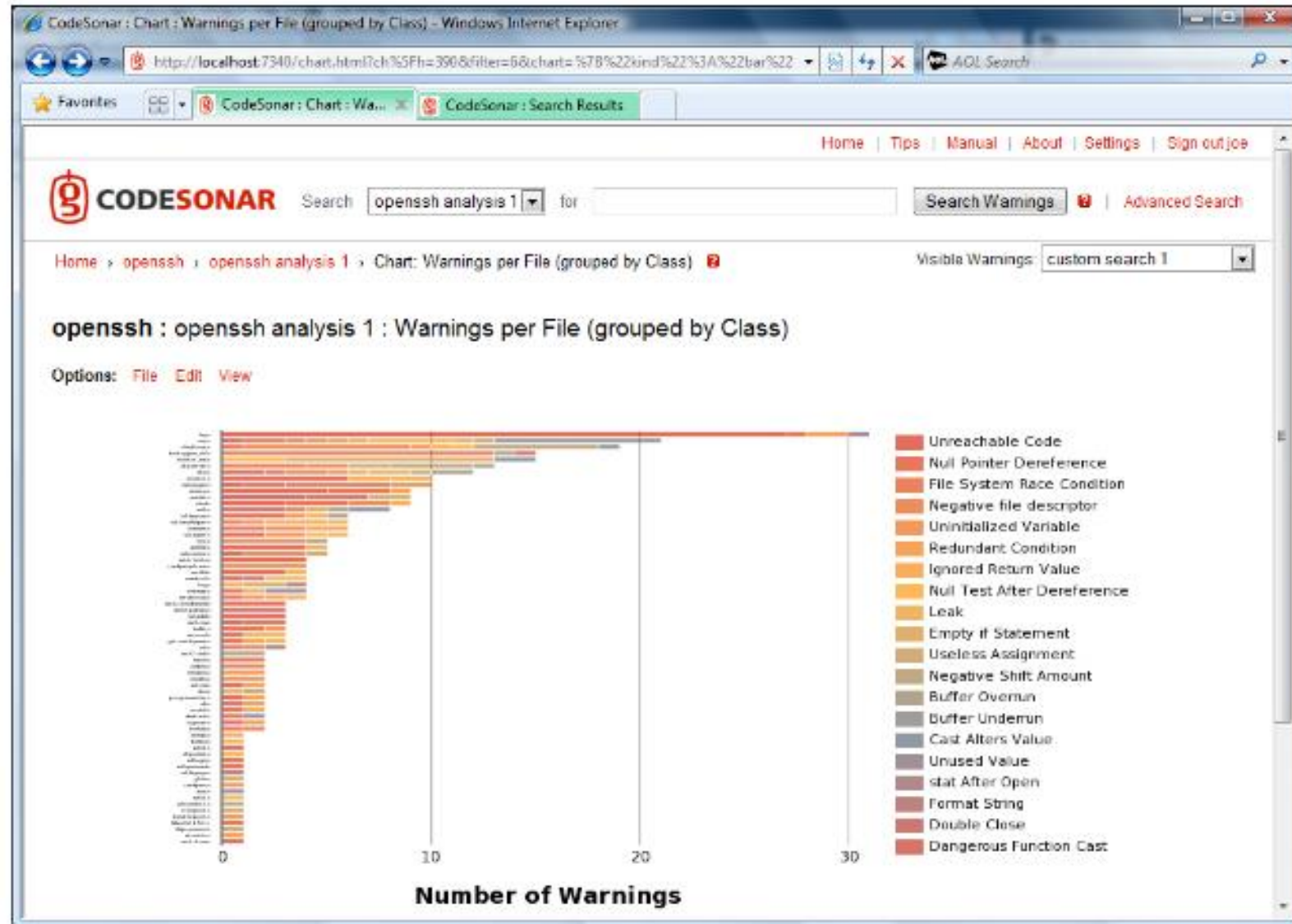


Static analysis tools (D)



CodeSonar chart summarizing warning locations and classes for a project.

Static analysis tools (D)



CodeSonar chart summarizing warning locations and classes for a project.

Static analysis tools (D)

Coverity® Integrity Manager Admin User Sign out | Preferences | Help | About Enter CID(s) Find

Dashboard **Projects** Configuration Administration

Projects > Filtered View > Chrome Web Browser Defects Source Metrics Reports Dashboard

Filters Defects Files

USE_AFTER_FREE
New / Unassigned / Other

12306: Use after free
USE_AFTER_FREE
New / Unassigned / Other

12305: Use after free
USE_AFTER_FREE
New / Unassigned / Other

12304: Use after free
USE_AFTER_FREE
New / Unassigned / Other

11742: Uninitialized pointer read
UNINIT
New / Unassigned / Other

11739: Uninitialized pointer read
UNINIT
New / Unassigned / Other

11726: Uninitialized pointer read
UNINIT
New / Unassigned / Other

11725: Uninitialized pointer read
UNINIT
New / Unassigned / Other

11629: String not null terminated
STRING_NULL
New / Unassigned / Other

11628: String not null terminated
STRING_NULL
New / Unassigned / Other

11490: Negative array index read
REVERSE_NEGATIVE
New / Unassigned / Other

Next 50 defects

Done

`/chromiumtrunk/home/chrome-svn/tarball/chromium/src/third_party/sqlite/src/ext/fts`

At conditional (4): "i < pQuery->nTerms" taking the true branch.
At conditional (6): "i < pQuery->nTerms" taking the true branch.
At conditional (8): "i < pQuery->nTerms" taking the true branch.
3886 for(i = 0; i < pQuery->nTerms; i=iNext){
At conditional (5): "(aTerm + i).isNot" taking the true branch.
At conditional (7): "(aTerm + i).isNot" taking the true branch.
At conditional (9): "(aTerm + i).isNot" taking the false branch.
3887 if(aTerm[i].isNot){
3888 /* Handle all NOT terms in a separate pass */
3889 nNot++;
3890 iNext = i + aTerm[i].nPhrase+1;
3891 continue;
3892 }
3893 iNext = i + aTerm[i].nPhrase + 1;
3894 rc = docListOfTerm(v, aTerm[i].iColumn, &aTerm[i], &right);
At conditional (10): "rc" taking the true branch.
3895 if(rc){
At conditional (11): "i != nNot" taking the true branch.
▶ Using uninitialized value "left.pData" when calling "dataBufferDestroy". [show details]
3896 if(i!=nNot) dataBufferDestroy(&left);
3897 queryClear(pQuery);
3898 return rc;
3899 }
3900 while(iNext < pQuery->nTerms && aTerm[iNext].isOr){
3901 rc = docListOfTerm(v, aTerm[iNext].iColumn, &aTerm[iNext], &or);
3902 iNext += aTerm[iNext].nPhrase + 1;
3903 if(rc){
▶ Using uninitialized value "left.pData" when calling "dataBufferDestroy". [show details]
3904 if(i!=nNot) dataBufferDestroy(&left);
3905 dataBufferDestroy(&right);
3906 queryClear(pQuery);
3907 return rc;
3908 }
3909 dataBufferInit(&new, 0);
3910 rc = docListOrMerge(right.pData, right.nData, or.pData, or.nData, &new);
3911 dataBufferDestroy(&right);
3912 dataBufferDestroy(&or);
3913 if(rc!=SQLITE_OK){
▶ Using uninitialized value "left.pData" when calling "dataBufferDestroy". [show details]
3914 if(i!=nNot) dataBufferDestroy(&left);
3915 queryClear(pQuery);
3916 dataBufferDestroy(&new);

11742 Uninitialized pointer read
In fulltextQuery(): Reads an uninitialized pointer or its target (CWE-457)

Classification: **Unclassified**
Severity: **Unspecified**
Action: **Undecided**
Owner: **Unassigned**
Ext. Reference:

Comment:

Apply + Next Apply Export Advanced...

Occurrences History Information

In Chrome Web Browser
Events contributing to defect:
var_decl fts2.c:3855
uninit_use_in_call fts2.c:3896
read_parm_fld fts2.c:517
uninit_use_in_call fts2.c:3904
uninit_use_in_call fts2.c:3914
uninit_use_in_call fts2.c:3949

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 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

**Modeling
tools (D)**

Validate models of the 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 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

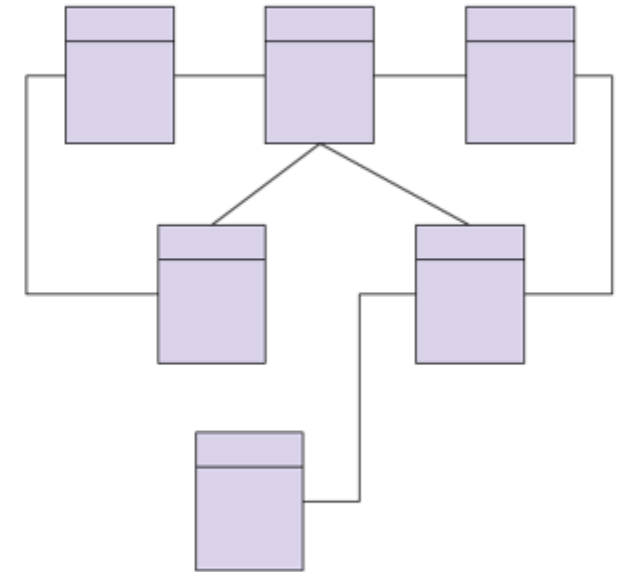
To better aid in designing the software

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**

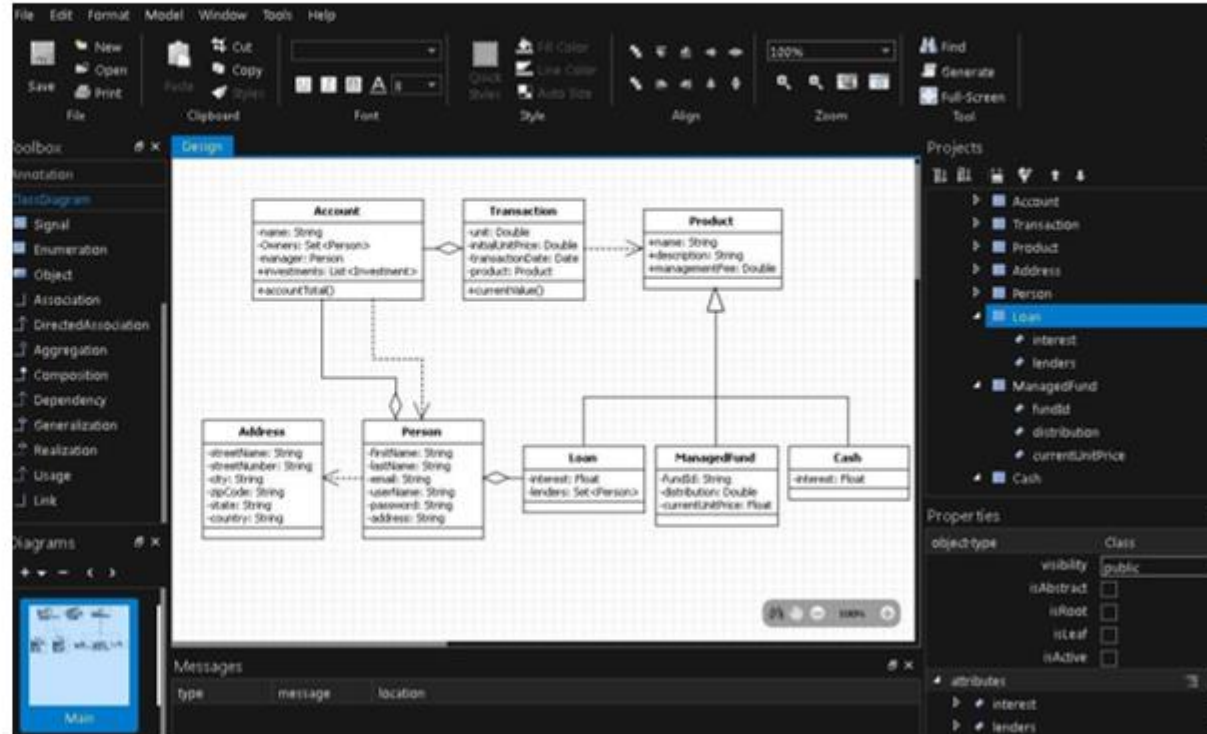
UML tool

Variety of **diagrams**

Class / Domain

Use case

Sequence



Modeling tools (D)

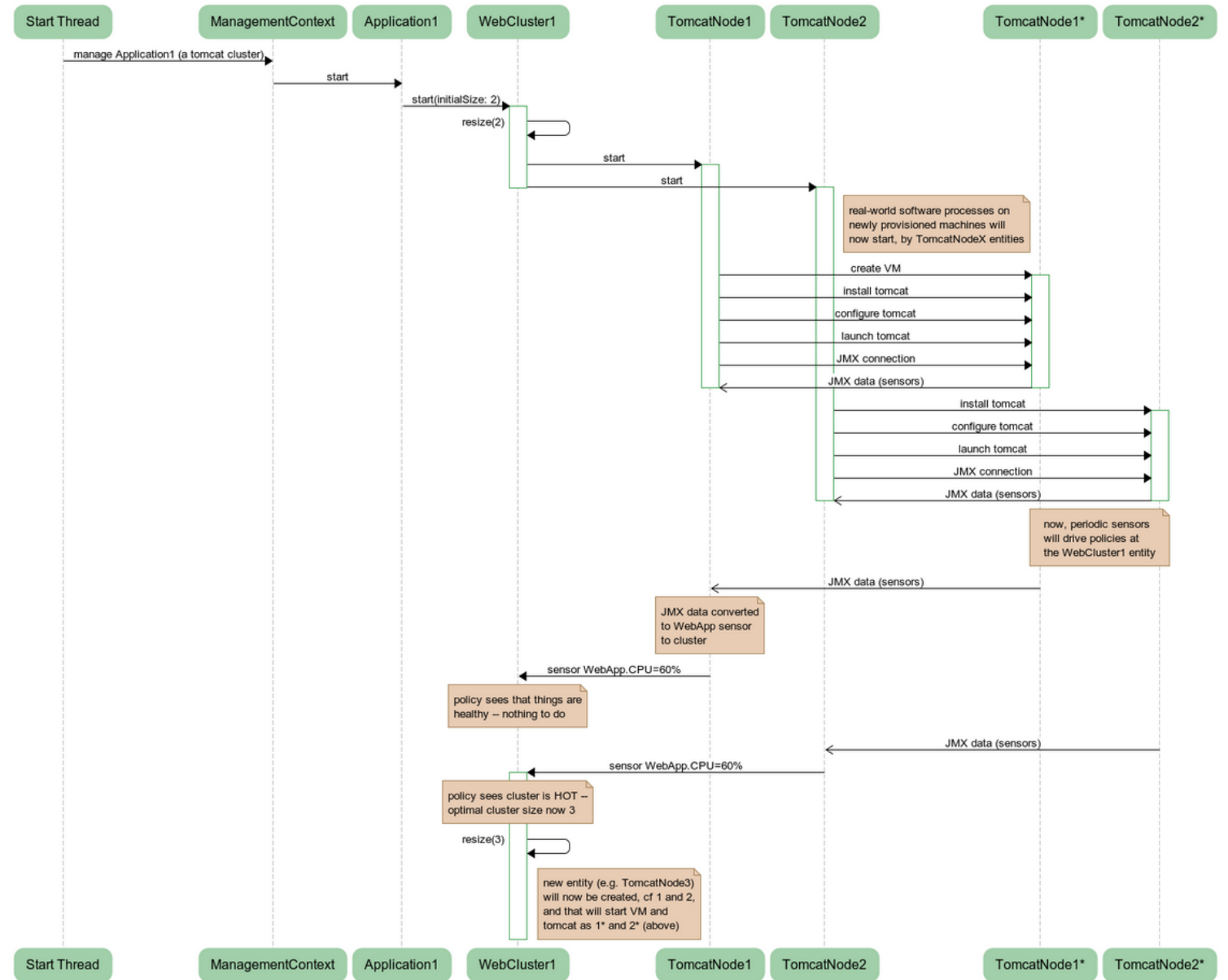
The screenshot displays the StarUML software interface for modeling UML 2.0. The main workspace shows a UML class diagram for the Model Management package. The diagram includes the following elements:

- Core::ModelElement**: A class with an association to **Core::Namespace** (multiplicity 1 to n) labeled `+importedElement` and an association to **Core::Generalizab** (multiplicity n to 1) labeled `+ownedElement`.
- Core::Namespace**: A class that inherits from **Core::Generalizab** and has an association to **Core::ModelElement** (multiplicity 0..1 to n) labeled `+namespace`.
- Core::Generalizab**: A class that is the superclass for **Core::Namespace**.
- ElementImport**: A class with an association to **Core::ModelElement** (multiplicity * to 1) labeled `+elementImport`. It has attributes: `Kind`, `boolean`, `package: Package`, and `importedElement: ModelElement`.

The interface also shows a **Model Explorer** on the right, displaying a tree view of the model structure, and a **Properties** window for the selected **[UMLClass] ElementImport** class. The **Output** console at the bottom shows the following messages:

```
[4:37:07 PM] The Profile "UMLStandard" is loaded successfully.  
[4:37:07 PM] The Approach "4p1ViewModel" is loaded successfully.  
[4:37:24 PM] C:\Program Files\StarUML\Samples\UML 2.0 Interchange Metamodel Abstract Syntax.uml
```

Modeling tools (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 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 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 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

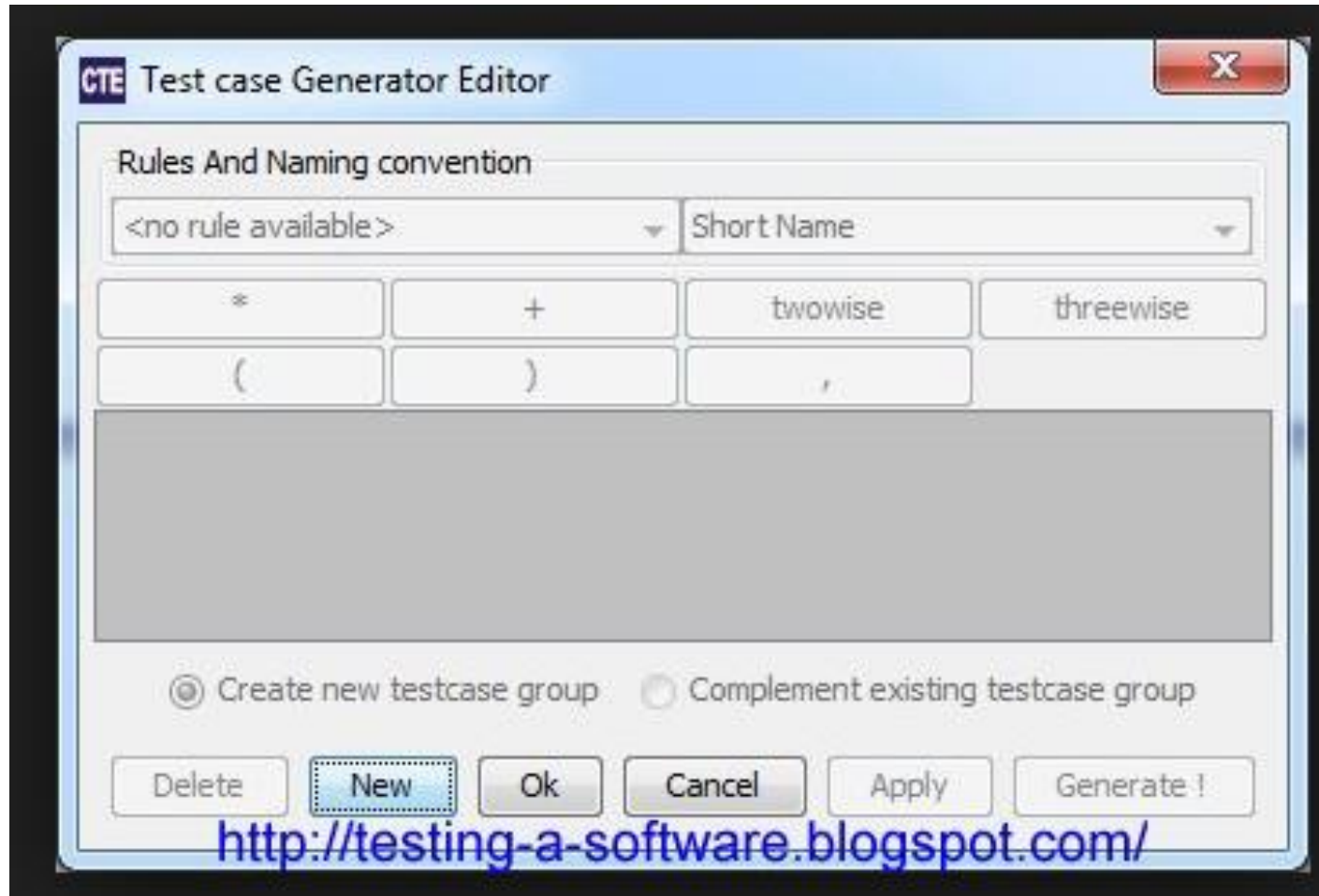
Test design tools

Generate test inputs or executable tests:
from requirements,
from a graphical user interface,
from design models (state, data or object)
or from code.

This type of tool may generate expected outcomes as well (i.e. may use a test oracle)

They can save valuable time and provide increased thoroughness of testing because of the completeness of the tests that the tool can generate.

Test design tools



Test design tools

#	Operating System	Language	Browser	#	Operating System	Language	Browser
1	Windows 8 64bit	French	Chrome latest	28	Ubuntu 12.10	German	Opera latest
2	Windows 7 Ultimate	Italian	Internet Explorer 8	29	Ubuntu 12.10	Spanish	Chrome latest
3	Windows XP home	French	Internet Explorer 8	30	Ubuntu 12.10	Chinese Simpl.	Opera latest
4	Ubuntu 12.10	French	Chrome latest	31	Ubuntu 12.10	Chinese Trad.	Chrome latest
5	Windows 8 64bit	German	Internet Explorer 10	32	Ubuntu 12.10	Japanese	Firefox latest
6	Windows 8 64bit	Spanish	Safari latest	33	Ubuntu 12.10	Korean	Firefox latest
7	OS X 10.8	Chinese Simpl.	Chrome latest	34	OS X 10.8	French	Safari latest
8	Windows 8 64bit	Chinese Trad.	Firefox latest	35	OS X 10.8	Italian	Firefox latest
9	Windows 8 64bit	Japanese	Opera latest	36	OS X 10.8	German	Opera latest
10	Windows 8 64bit	Korean	Internet Explorer 10	37	OS X 10.8	Spanish	Opera latest
11	Windows 8 64bit	Italian	Internet Explorer 10	38	OS X 10.8	Chinese Trad.	Safari latest
12	Windows 8 64bit	Chinese Simpl.	Internet Explorer 10	39	OS X 10.8	Japanese	Safari latest
13	Windows 7 Ultimate	French	Internet Explorer 10	40	OS X 10.8	Korean	Safari latest
14	Windows 7 Ultimate	German	Chrome latest	41	Windows XP home	French	Firefox latest
15	Windows 7 Ultimate	Spanish	Safari latest	42	Ubuntu 12.10	French	Opera latest
16	Windows 7 Ultimate	Chinese Simpl.	Firefox latest	43	Windows 7 Ultimate	Italian	Safari latest
17	Windows 7 Ultimate	Chinese Trad.	Opera latest	44	OS X 10.8	Italian	Opera latest
18	Windows 7 Ultimate	Japanese	Internet Explorer 8	45	Windows 7 Ultimate	German	Internet Explorer 8
19	Windows 7 Ultimate	Korean	Internet Explorer 8	46	Windows 8 64bit	German	Firefox latest
20	Windows XP home	Italian	Chrome latest	47	Windows 7 Ultimate	Spanish	Internet Explorer 8
21	Windows XP home	German	Safari latest	48	Windows 8 64bit	Spanish	Internet Explorer 10
22	Windows XP home	Spanish	Firefox latest	49	Windows XP home	Chinese Simpl.	Internet Explorer 8
23	Windows XP home	Chinese Simpl.	Opera latest	50	OS X 10.8	Chinese Simpl.	Safari latest
24	Windows XP home	Chinese Trad.	Internet Explorer 8	51	Windows 7 Ultimate	Chinese Trad.	Internet Explorer 10
25	Windows XP home	Japanese	Chrome latest	52	Windows 8 64bit	Japanese	Internet Explorer 10
26	Windows XP home	Korean	Chrome latest	53	Windows 8 64bit	Korean	Opera latest
27	Ubuntu 12.10	Italian	Firefox latest				

Tools support for **test 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 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 preparation tools

Manipulate databases or files to set up test data to be used during the execution of tests

Benefit: they ensure that live data in a test environment is made anonymous, for data protection.

Tools support for **test 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 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 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 specification
- 1.6 Tools for execution and logging
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

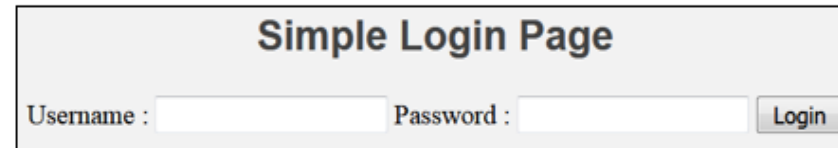
2. Effective use of test tools

- 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.

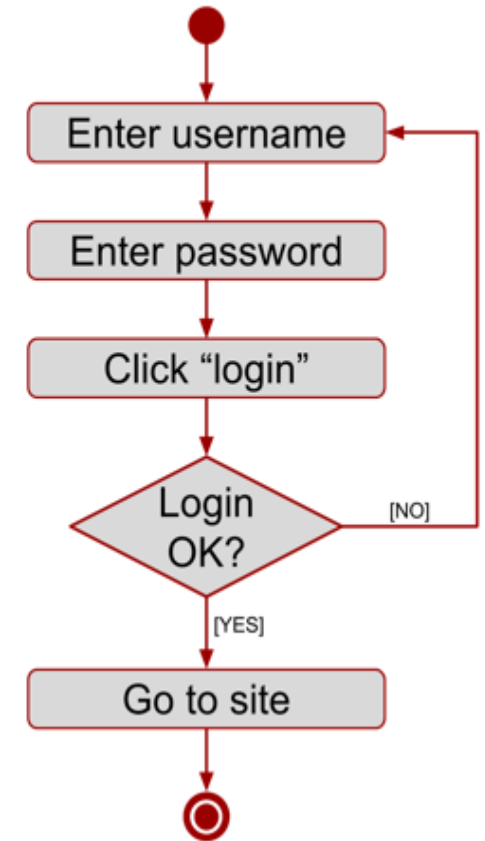
Data need for testing website

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 **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 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?

Test data for whitebox-testing

Concern: **Coverage**

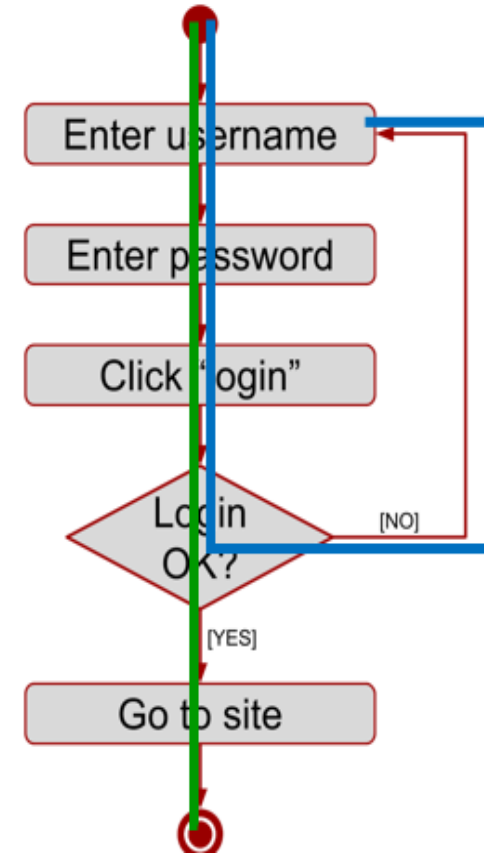
Ensure all **branches** are tested at least once

Generate **data** for this purpose

Example

Invalid combination of username and password

Valid combination of username and password



Tools support for **test 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 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?

Test data for blackbox-testing

No data

Valid / Invalid data sets

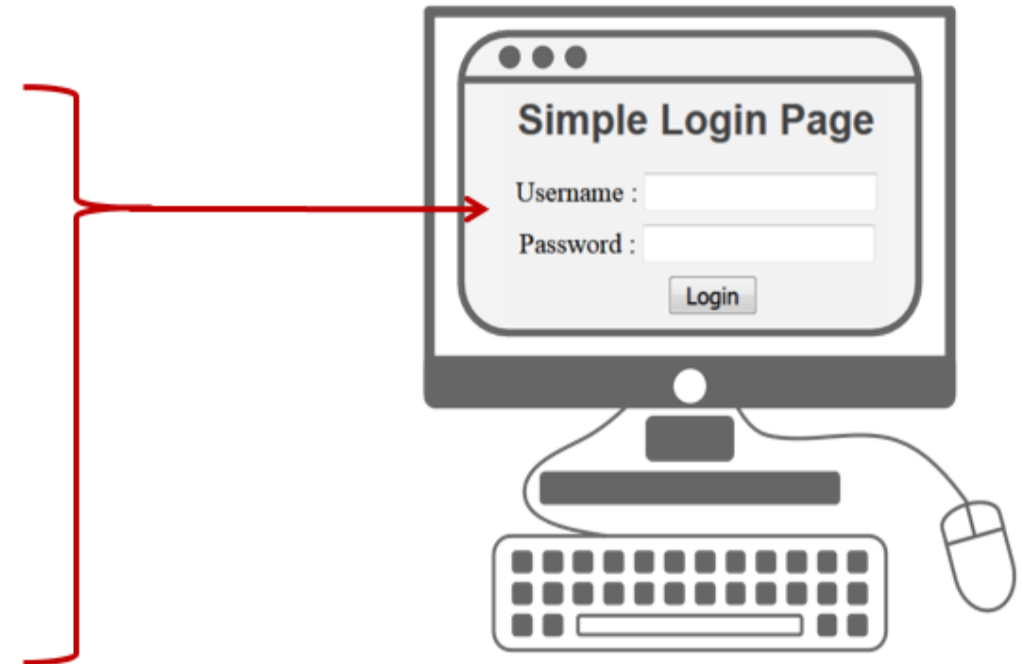
Illegal data sets

Equivalence and Boundary data sets

Decision table data sets

State transition data sets

Use case data sets



Tools support for **test 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 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?

Test data for **security** testing

Confidentiality

Test data to verify **correct encryption**

Integrity

Test data to verify **correct information** provided

Authentication and **authorisation**

Test data to verify correct identity management

Combinations of users, roles, operations



Tools support for **test 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 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?

Test data for **performance** testing

Real data

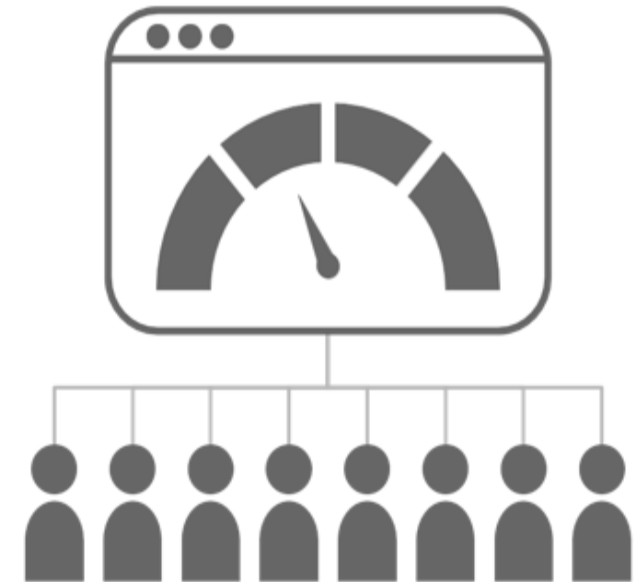
Test data obtained from **users**

Load

Large amounts of test data can be produced

Maintenance

Test data from the production environment



Test data preparation tools



DTM Data Generator

Realistic data for database testing purposes

[Features](#) • [Revision History](#) • [On-line Documentation](#) • [Press releases](#) • [Screenshots](#) • [Downloads](#) • [SDK](#) • [Order](#)

- DTM SQL editor
- DTM Schema Reporter
- DTM Migration Kit
- **DTM Data Generator**
- DTM Data Editor
- DTM DB Stress
- DTM Data Modeler
- DTM Data Scrubber
- DTM Data Comparer
- DTM Schema Comparer
- DTM Query Reporter
- DTM Schema Inspector
- DTM DB Event
- DTM XML Generator
- DTM Flat File Generator

> Product Line

> Downloads

> Articles

> Support and Contacts

> Site Search



Download
free
Demo



Order
Now !

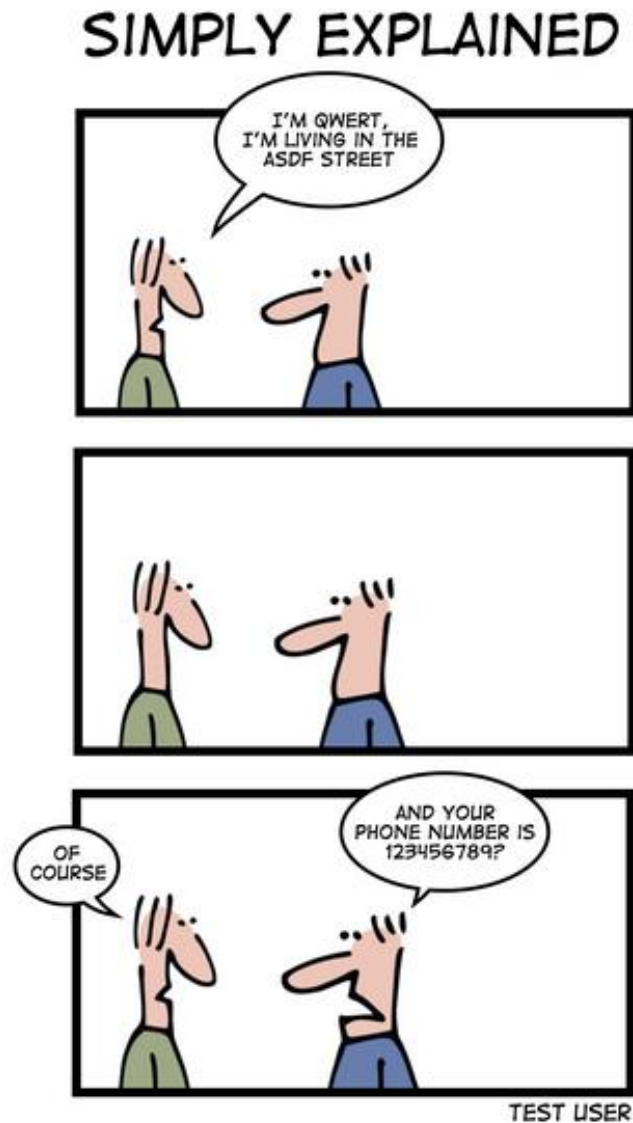
DTM Data Generator Screenshots

Main program window. Rule list

The screenshot shows the main program window of DTM Data Generator Enterprise. The window title is "DTM Data Generator Enterprise [D:\Projects And Files\NorthWind.dgp]". The menu bar includes File, Project, Rule, Tools, Plug-ins, and Help. The toolbar contains various icons for file operations and execution. The main area is titled "Data Generator Project (14 rules)" and contains a table of rule definitions. The table has columns for Rule definition, Generate, Active, and Note. To the right of the table are several buttons: Add Data Rule, Add File Rule, Add Tables Rule, Add Objects Rule, Add Clear Rule, Edit Rule, Remove Rule, Up, Down, Report..., Run Checked, and Run All. The status bar at the bottom indicates "Connected to SQL Server Native Client 10.0: //dbo/Northwind [Local Northwind DB]".

Rule definition	Generate	Active	Note
<input type="checkbox"/> Clear Rule, 14 table(s)	14	Yes	
<input type="checkbox"/> Generator for 'Categories', append	1000/500	Yes	
<input type="checkbox"/> Generator for 'CustomerDemographics', append	1000/500	Yes	
<input type="checkbox"/> Generator for 'Customers', replace	20/500	Yes	
<input type="checkbox"/> Generator for 'Employees', append	1000/500	Yes	
<input type="checkbox"/> Generator for 'Region', append	1000/500	Yes	
<input type="checkbox"/> Generator for 'Shippers', append	1000/500	Yes	
<input type="checkbox"/> Generator for 'Suppliers', append	1000/500	Yes	
<input type="checkbox"/> Generator for 'Territories', append	1000/500	Yes	
<input type="checkbox"/> Generator for 'CustomerCustomerDemo', append	1000/500	Yes	
<input type="checkbox"/> Generator for 'EmployeeTerritories', append	1000/500	Yes	
<input type="checkbox"/> Generator for 'Orders', append	1000/500	Yes	
<input type="checkbox"/> Generator for 'Products', append	1000/500	Yes	
<input type="checkbox"/> Generator for "Order Details", append	1000/500	Yes	

Test data preparation 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 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 & expected outcomes

The scripting language allows to manipulate the tests with little effort(i.e. repeat the test with other data)

Can also be used to record tests(capture & playback tools)

Tools support for **test execution & logging**

1. Types of test tools

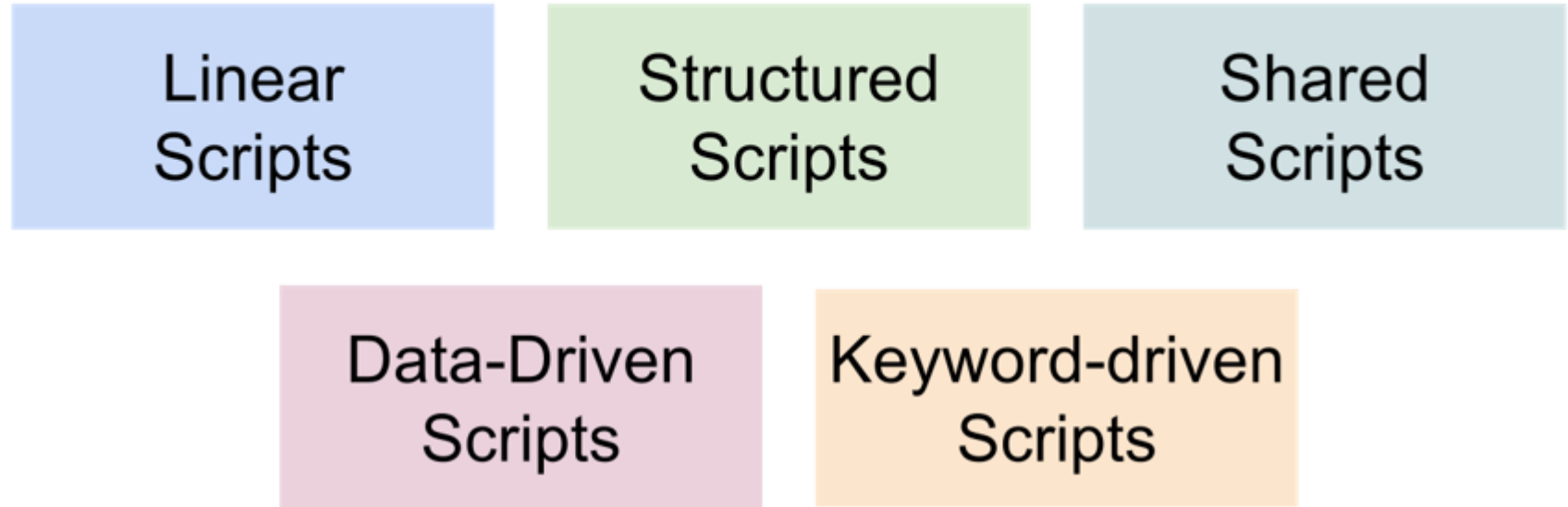
- 1.1 Tool support for testing
- 1.2 Test tool classification
- 1.3 Tools for test management
- 1.4 Tools for static testing
- 1.5 Tools for test specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

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 specification
- **1.6 Tools for execution and logging**
- 1.7 Tools for performance & monitoring
- 1.8 Tools for specific testing needs

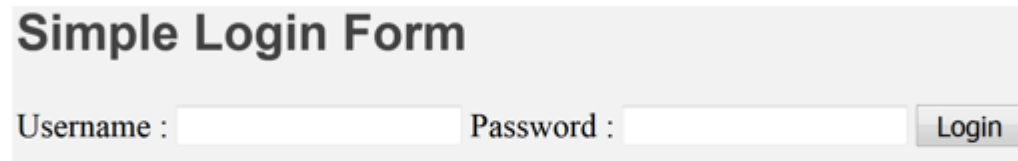
2. Effective use of test tools

- 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 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

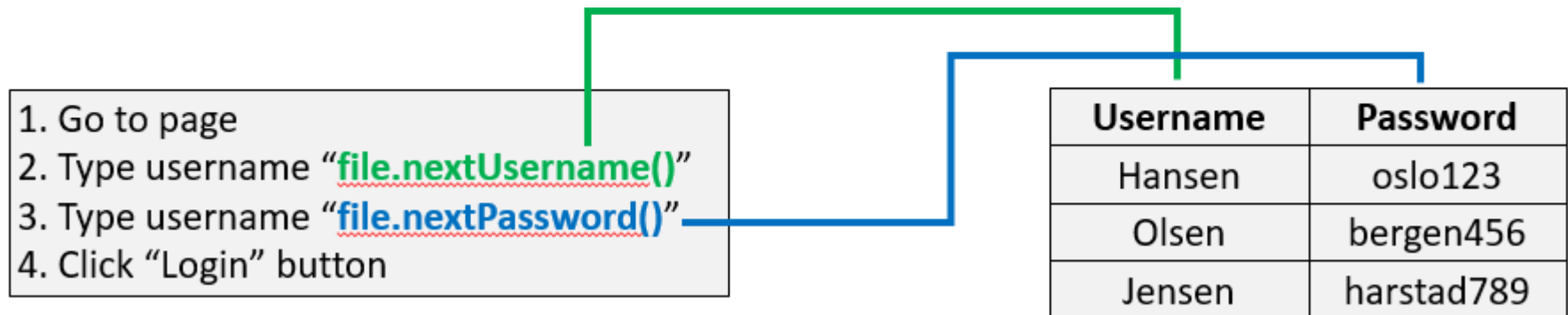
3. Introducing a test tool into an organization

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 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

Test execution tools

The screenshot displays the Eclipse IDE interface with the following components:

- Package Explorer:** Shows a project structure with folders like AjaxAuthentication, Database, Selenium, and Servers. The Selenium folder is expanded to show src, JUnit 4, and Referenced Libraries.
- Editor:** Displays the code for `JavadoainTest.java`. The code imports `org.junit.Test`, `com.thoughtworks.selenium.DefaultSelenium`, and `com.thoughtworks.selenium.Selenium`. It defines a `public class JavadoainTest` with a `@Test` annotated `public void jdtesting() throws InterruptedException` method. The method uses Selenium to start a browser, navigate to `http://google.com`, type into a search box, and click on a link.
- Task List:** Shows a search bar and a "Connect Mylyn" button.
- Outline:** Shows the class `JavadoainTest` and its method `jdtesting0 : void`.
- Console:** Shows the output of the test execution, indicating that the test terminated successfully.

```
import org.junit.Test;

import com.thoughtworks.selenium.DefaultSelenium;
import com.thoughtworks.selenium.Selenium;

public class JavadoainTest{

    @Test
    public void jdtesting() throws InterruptedException{

        Selenium selenium = new DefaultSelenium("localhost",4444,"firefox","http://google.com");
        selenium.start(); //starting selenium instance
        selenium.open("/");
        selenium.windowMaximize();
        selenium.windowFocus();
        selenium.type("//*[@id='gbqfq']", "javadoain.in");
        selenium.click("//*[@id='gbqfb']");
        Thread.sleep(20000);
        selenium.click("//*[@id='rso']/li[1]/div/h3/a/em");
        Thread.sleep(80000);
        selenium.close(); //to close the browser
        selenium.stop(); //to stop selenium instance

    }
}
```

<terminated> JavadoainTest [JUnit] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (Dec 27, 2012 10:58:44 PM)

Writable Smart Insert 27 : 2

Test execution tools

The screenshot displays the Microsoft Visual Studio interface. The main window shows the 'Test Results' pane, which indicates that a test run is completed with 17/17 tests passed. Below this, a table lists the individual test results, including the test name, project, error message, and duration.

Result	Test Name	Project	Error Message	Duration
Passed	LogOff_LogsOutAndRedirects	Big.Fat.Dish.Tests		00:00:00.0906167
Passed	Register_Post_ReturnsViewIfRegistrationFails	Big.Fat.Dish.Tests		00:00:00.0120230
Passed	Register_Post_ReturnsViewIfModelStateIsInvalid	Big.Fat.Dish.Tests		00:00:00.0007535
Passed	LogOn_Post_ReturnsViewIfModelStateIsInvalid	Big.Fat.Dish.Tests		00:00:00.0014254
Passed	LogOn_Get_ReturnsView	Big.Fat.Dish.Tests		00:00:00.0005725
Passed	ChangePassword_Post_ReturnsViewIfChangePasswordFails	Big.Fat.Dish.Tests		00:00:00.0105882
Passed	Register_Post_ReturnsRedirectOnSuccess	Big.Fat.Dish.Tests		00:00:00.0006481
Passed	Index	Big.Fat.Dish.Tests		00:00:00.0008837
Passed	LogOn_Post_ReturnsRedirectOnSuccess_WithReturnUrl	Big.Fat.Dish.Tests		00:00:00.0017549
Passed	LogOn_Post_ReturnsRedirectOnSuccess_WithoutReturnUrl	Big.Fat.Dish.Tests		00:00:00.0005188
Passed	ChangePassword_Post_ReturnsViewIfModelStateIsInvalid	Big.Fat.Dish.Tests		00:00:00.0004918
Passed	ChangePasswordSuccess_ReturnsView	Big.Fat.Dish.Tests		00:00:00.0004222
Passed	Register_Get_ReturnsView	Big.Fat.Dish.Tests		00:00:00.0005678
Passed	ChangePassword_Get_ReturnsView	Big.Fat.Dish.Tests		00:00:00.0005879
Passed	LogOn_Post_ReturnsViewIfValidateUserFails	Big.Fat.Dish.Tests		00:00:00.0005958
Passed	ChangePassword_Post_ReturnsRedirectOnSuccess	Big.Fat.Dish.Tests		00:00:00.0007185
Passed	About	Big.Fat.Dish.Tests		00:00:00.0004423

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 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 / unit test framework tools (D)

Facilitate the test of components of a system - simulating the environment in which that test object will run.

They may be called unit test tools when they have a particular focus on the component test level.

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 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 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

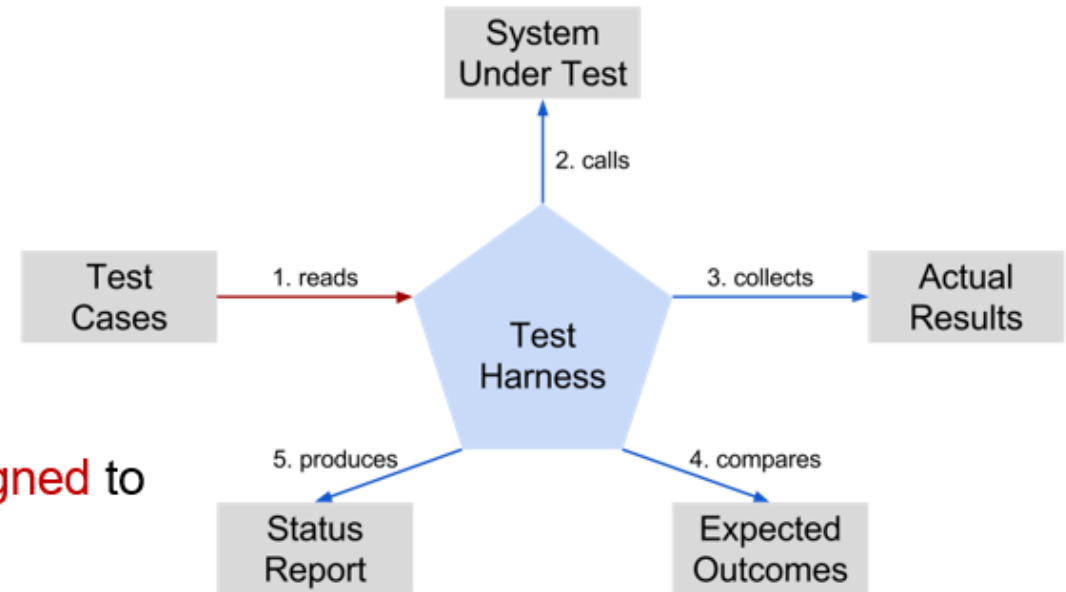
Test **harness** tools (D)

“**Enabler**” that does all the work of:

- (i) **Executing** the tests, using
- (ii) A test **library**, and
- (iii) Generates **reports**

Requires that the test **scripts** are **designed** to

- (iv) **Handle** different **data**, and
- (v) Test **scenarios**



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 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*

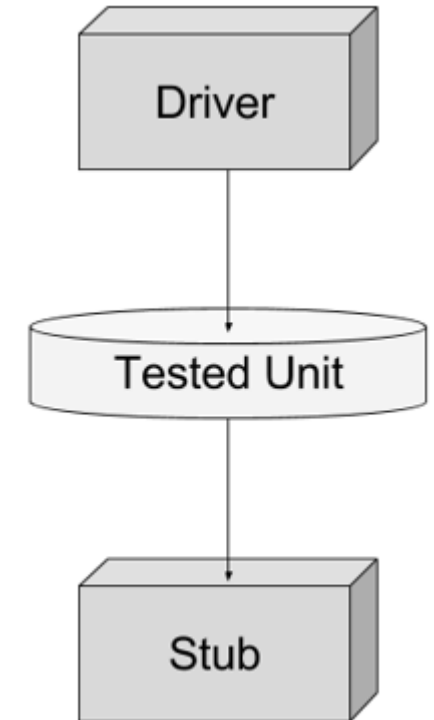
Stubs

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 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

Test harness/unit test framework tools (D)

```
[Test]
public void RhinoMocksDemo()
{
    MockRepository mocks = new MockRepository();

    // create some mocks
    IEmailBuilder emailBuilder = mocks.CreateMock<IEmailBuilder>();
    IEmailSender emailSender = mocks.CreateMock<IEmailSender>();

    Email email = new Email();

    // set expectations
    Expect.Call(emailBuilder.Create()).Return(email);
    emailSender.Send(email);

    // execute the test
    mocks.ReplayAll();
    Reporter reporter = new Reporter(emailSender, emailBuilder);
    reporter.SendSomeEmails();
    mocks.VerifyAll();
}
```

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 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

Determine differences between files, databases or test results

Test execution tools include dynamic comparators, but post-execution comparison may be done by a separate comparison tool.

A test comparator may use a test oracle, especially if it is automated.

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 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 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?

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 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.

Post-execution comparison

Done by a **separate**, standalone **tool** (not test execution tool)

>> **Test comparator** / test comparison tool

Best for comparing **large amounts** of **data**

Example

Comparing the contents of an entire file

Does the produced file match the expected contents of that file?

Comparing a large set of records from a database to the expected contents

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. ?

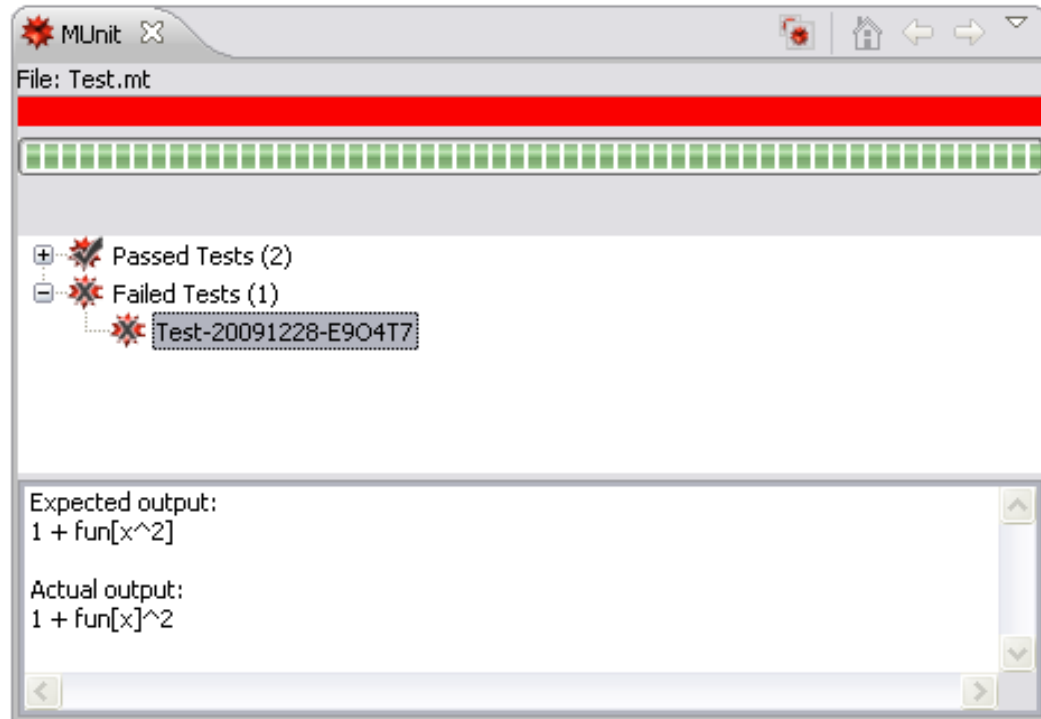
Test comparators

Mathematica Development User Guide > Tasks > Wolfram MUnit Tester

The Test Comparator

The test comparator is a useful feature for seeing exactly what the error is in a test failure.

First, you should run a [test file](#) or [test suite](#). Then select one of the test failures: this should show the expected and actual results in the test report. A sample is shown in the following.



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 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

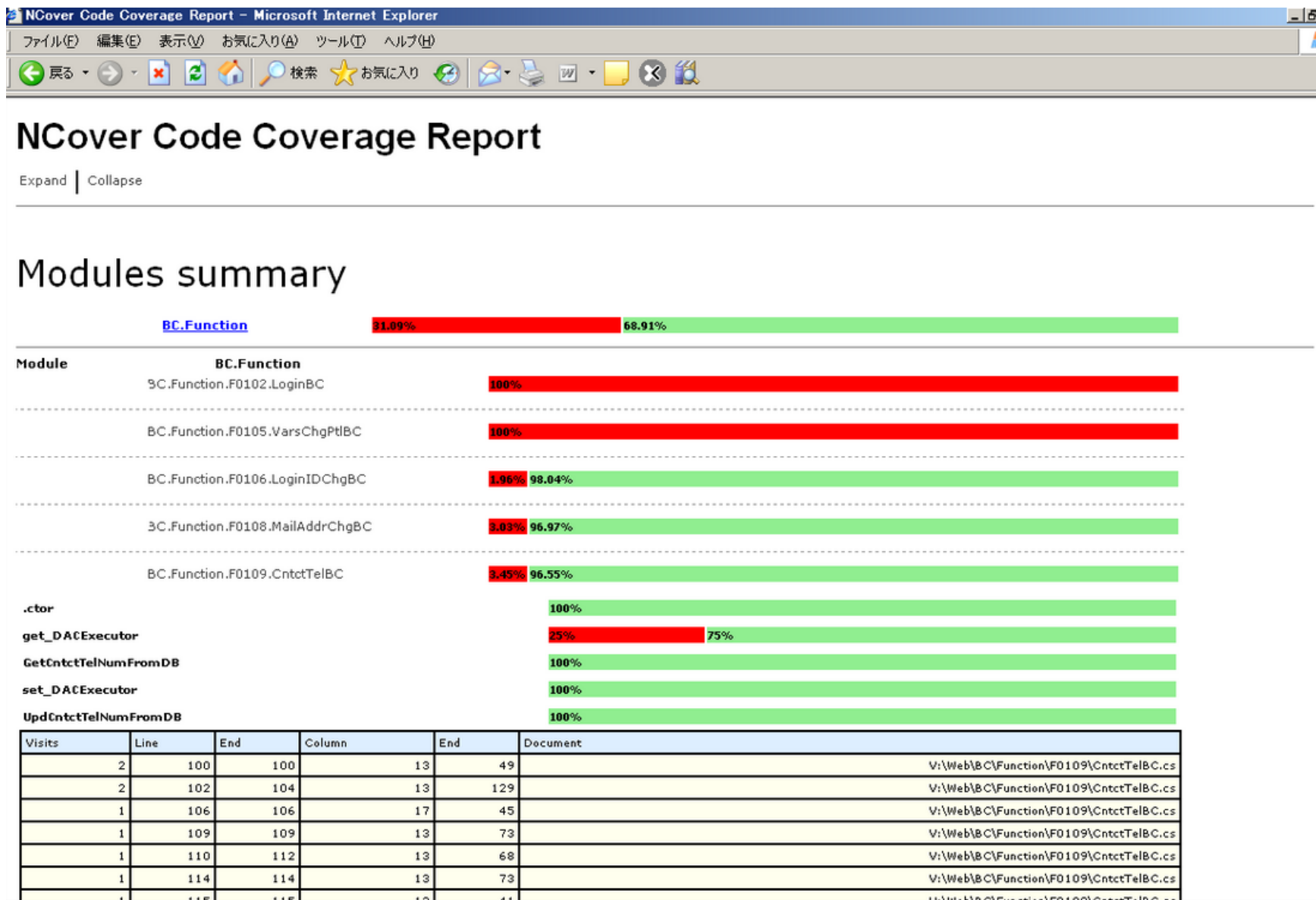
3. Introducing a test tool into an organization

Coverage measurement tools (D)

Can be intrusive or non intrusive (depends on the measurement technique used)

Measure the percentage of specific types of code structure that have been exercised

Coverage measurement tools (D)



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 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

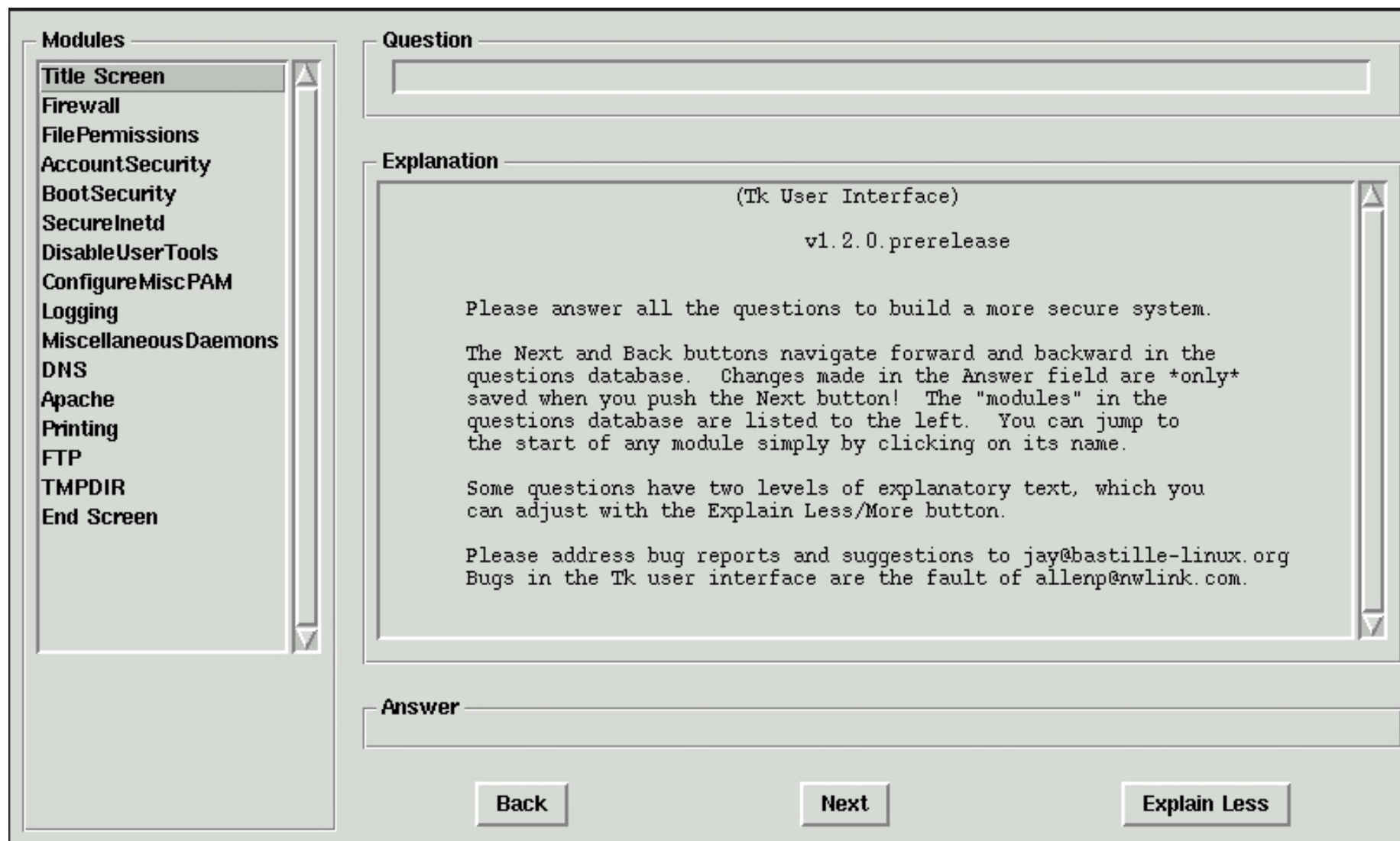
- 2.1 Potential benefits and risks of tools
- 2.2 Special considerations for tools

3. Introducing a test tool into an organization

Security tools

Search for specific vulnerabilities of the system

Security 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 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

Performance testing/ load testing/ stress testing tools

Monitor and report on how a system behaves under a variety of simulated usage conditions.

They simulate a load on:
- an **application**, a **database**, or a **system environment**.

They are often based on automated repetitive execution of tests, controlled by parameters.

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 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

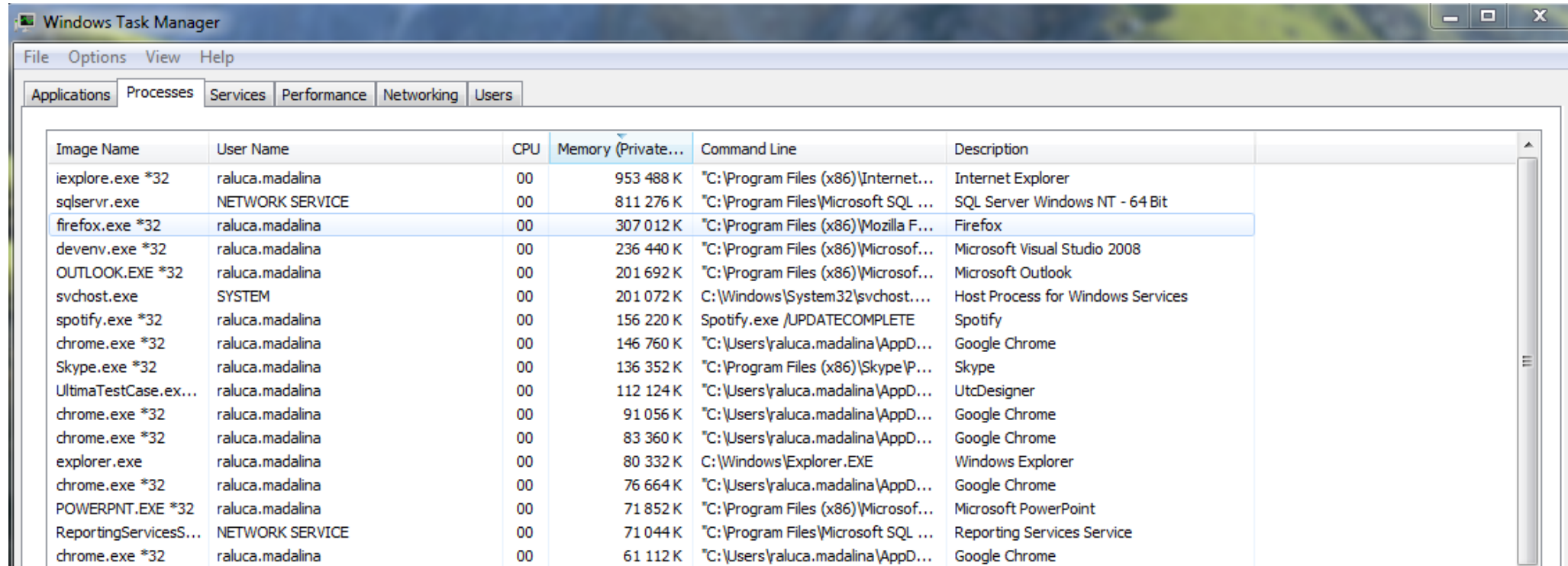
3. Introducing a test tool into an organization

Dynamic analysis tools (D)

Find defects that appear only when software is executing (i.e. memory leaks)

They are typically used in component and component integration testing

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 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 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

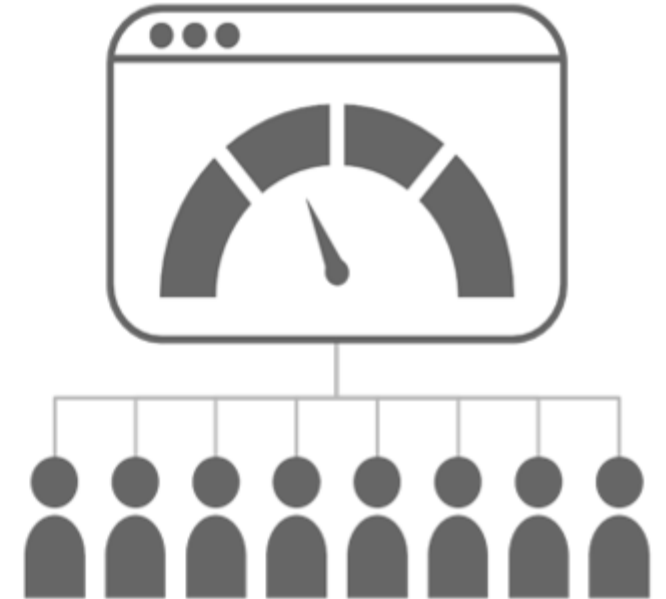
Test data obtained from **users**

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 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 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

Performance testing/load testing/stress testing tools

HP LoadRunner Controller - Scenario1 - [Run]

File View Scenario Monitors Results Diagnostics Tools Help

Scenario Groups

Group Name	Down	Pending	Init	Ready	Run	Rendez	Passed	Failed	Error	Gradual Exiting	Exiting	Stopped
1	0	0	0	0	5	0	0	0	0	4	0	1
HeroldYPWPWith					5					4		1

Scenario Status: **Running**

Running Vusers: 9
 Elapsed Time: 00:02:35 (hh:mm:ss)
 Hits/Second: 7.62 (last 60 sec)
 Passed Transactions: 56
 Failed Transactions: 0
 Errors: 0

Available Graphs

- Running Vusers
- User Defined Data Points
- Error Statistics
- Vusers with Errors
- Transaction Graphs
 - Trans Response Time
 - Trans/Sec (Passed)
 - Trans/Sec (Failed, Stopped)
 - Total Trans/Sec (Passed)
- Web Resource Graphs
 - Hits per Second
 - Throughput
 - HTTP Responses per Second
 - Pages Downloaded per Second
 - Retries per Second
 - Connections
 - Connections per Second
 - SSL per Second
- System Resource Graphs
 - Windows Resources
 - UNIX Resources

Running Vusers - whole scenario

Trans Response Time - whole scenario

User Defined Data Points - whole scenario

Total Trans/Sec (Passed) - whole scenario

Hits per Second - whole scenario

Windows Resources - Last 60 sec

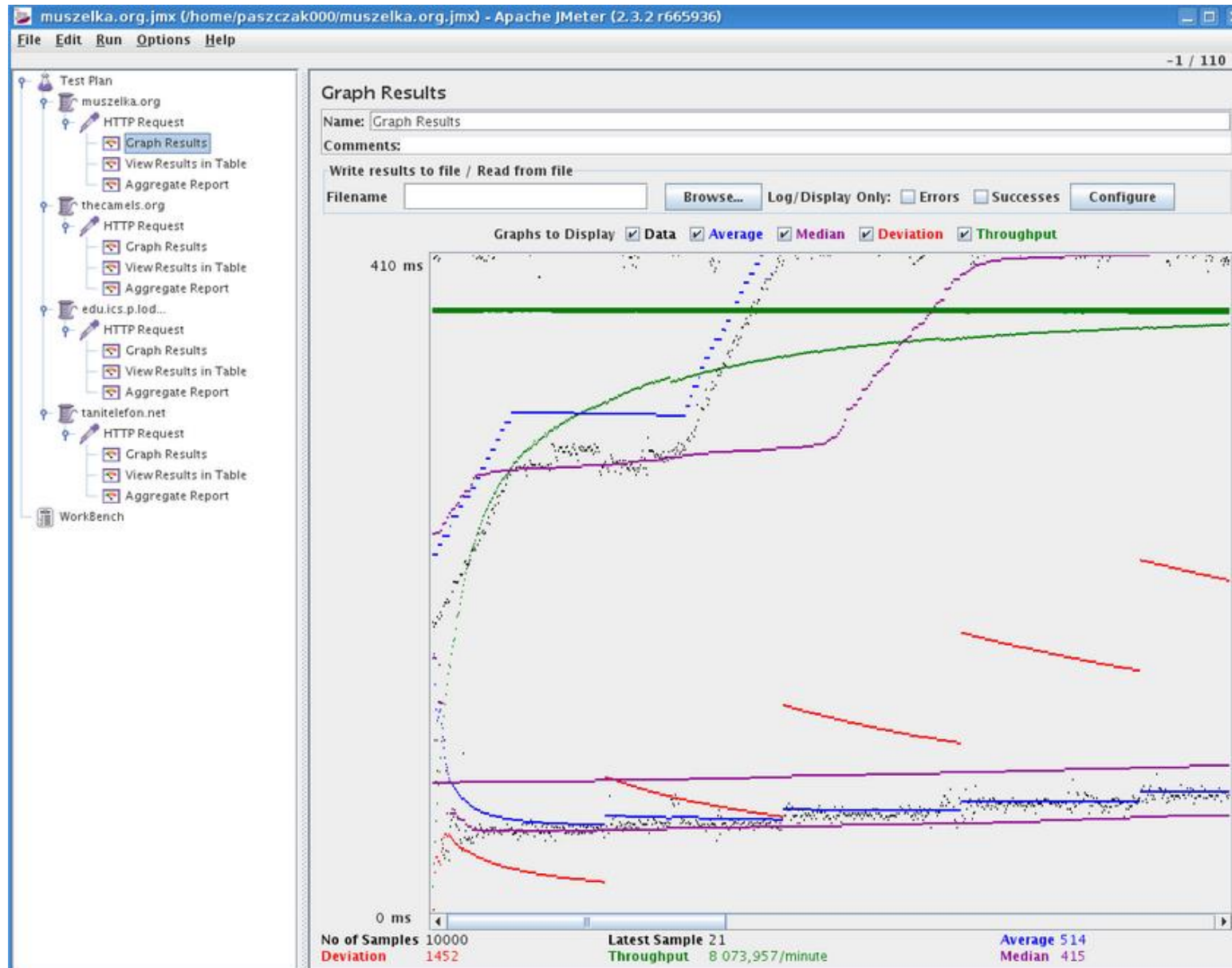
Throughput - whole scenario

Error Statistics - whole scenario

Color	Scale	Status	Max	Min	Avg	Std	Last
Blue	1	Running	10,000	0,000	7.471	N/A	10,000
Brown	1	Ready	0,000	0,000	0,000	N/A	0,000
Yellow	1	Finished	0,000	0,000	0,000	N/A	0,000
Red	1	Error	0,000	0,000	0,000	N/A	0,000

Design Run Diagnostics for J2EE/.NET

Performance testing/load testing/stress testing 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 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

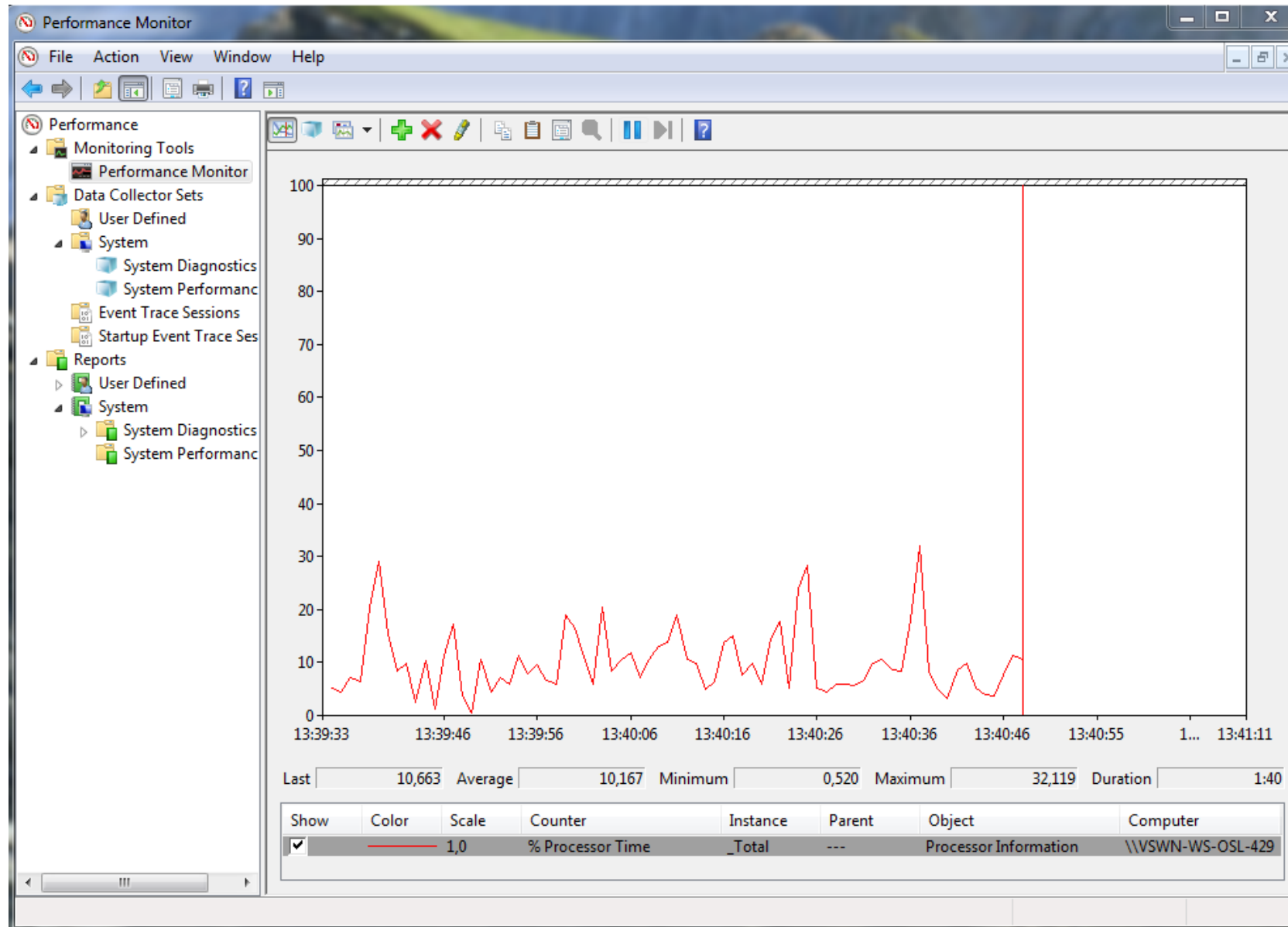
3. Introducing a test tool into an organization

Monitoring tools

Are not strictly testing tools but provide information that can be used for testing purposes.

Analyze, verify and report on usage of specific system resources, and give warnings of possible service problems.

Monitoring tools



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 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 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 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 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 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 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

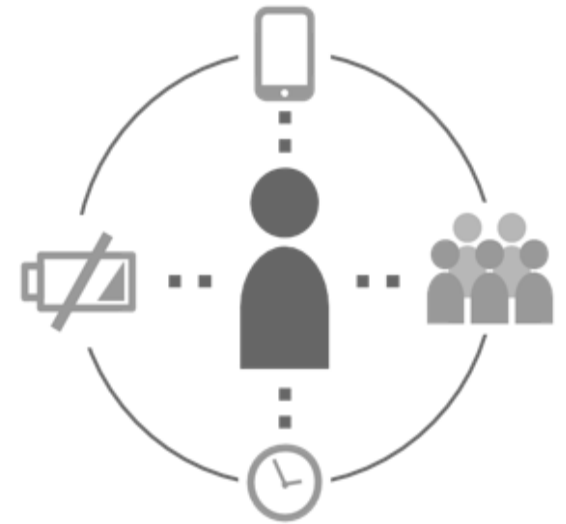
Doing more than one task simultaneously

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 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 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**

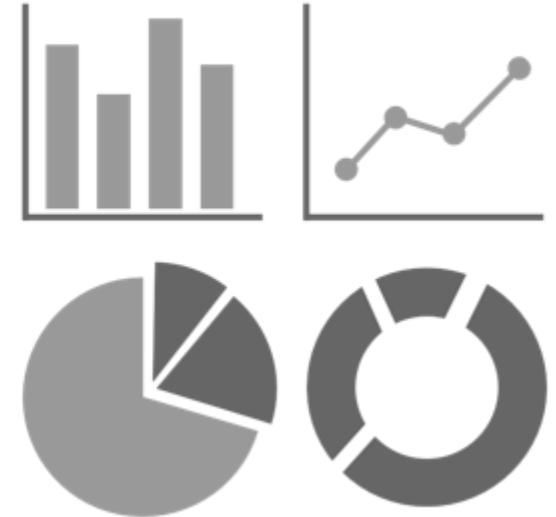
Chart, graphs > Long list of numbers

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 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 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 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 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
- **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 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 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 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 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 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

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**.

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 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