



# Testing automation

Automated test life-cycle  
methodology

Linda Roze

# What is automated testing?

◆ Test automation is the use of software to control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions

/Wikipedia/

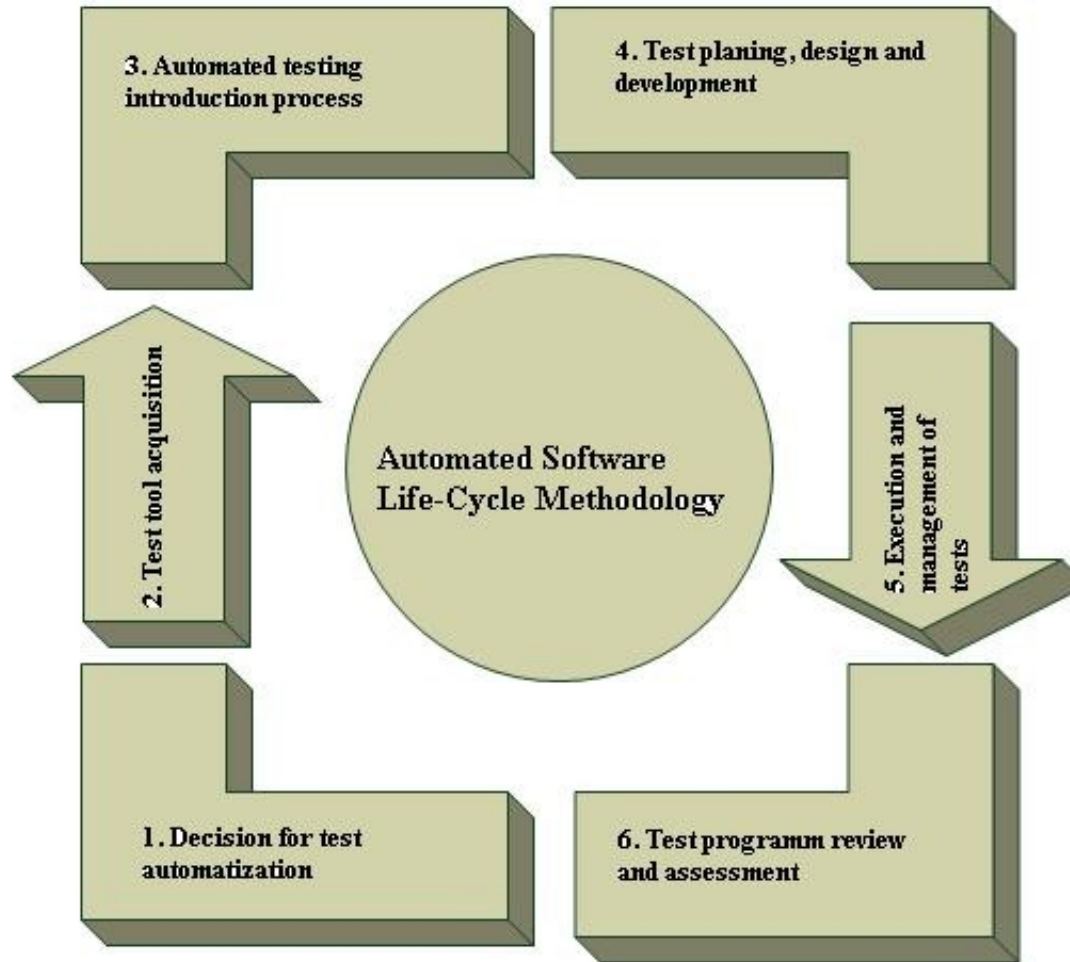
◆ Testing which is performed, to a greater or lesser extent, by a computer, rather than manually.

/Joe Strazzere/

# The definition of ATLM

- ◆ The automated software testing life cycle methodology
- ◆ Introduced by automated testing specialists Elfriede Dustin, Jeff Rashka un John Paul
- ◆ Structured approach for successful implementation of automated testing
- ◆ Multistage process consisting of six components, that supports the detailed and interrelated activities for test automatization

# ATLM diagram



# Decision for test automation

- ◆ Overcoming false expectations for test automation
- ◆ Understanding the benefits of test automation
- ◆ Acquiring management support
- ◆ Development of automated test tool proposal
  - Estimated improvement opportunities
  - Criteria for selecting the correct tool
  - Tool cost range
  - Necessary time to introduce tool
  - Tool expertise and training cost
  - Tool evaluation domain and implementation process

# Test tool acquisition

- ◆ Review organization's engineering environment
- ◆ Review types of test tools available
- ◆ Research for test tool candidates and rate them
- ◆ Define evaluation domain
- ◆ Evaluate top 2 or 3 highest rated test tools
- ◆ Review the outcomes of test tool evaluation
- ◆ Tool purchase and pilot project development

# Automated testing introduction process

- ◆ Organization's test process analysis for better demonstration of the benefits gained from test automatization
- ◆ Improvement of test process
- ◆ Test tool consideration for specific project
  - What are the software's requirements?
  - What is the application under test?
  - Is the software compatible with chosen test tool?
  - Is there enough time for test automatization in project schedule?
- ◆ Demonstration of test tool for project participants
- ◆ Definition of the roles and responsibilities

# Test planning, design and development

## ◆ Development of test plan

- Test requirement analysis
- Definition of test scope, test team and its responsibility, testing schedule
- Definition of testing risks and limitations

## ◆ Development of test design model and architecture

- Design-based test architecture (groups test procedures with the system application design components)
- Technique-based test architecture (groups test procedures with the various test techniques)

## ◆ Development of test procedure design and guidelines

## ◆ Development of test procedures

# Execution and management of tests

- ◆ Test execution according to the test plan
- ◆ Documentation of the test metrics (test coverage, progress metrics, quality metrics)
- ◆ Evaluation of the test outcome
  - False negative outcome – change in the application, test setup or user error, test procedure error
  - False positive outcome – test procedure executed successfully but problem in the application under test actually exists

# Test programm review and assessment

- ◆ Review of testing performance and collected testing metrics
- ◆ Comparing actually performed testing activities and used resources with initially planned
- ◆ Identification of successful activities and not so successful activities
- ◆ Definition of recommendations and corrective actions for test process improvement
- ◆ Documentation of lessons learned and improvements to be implemented

# What was the problem?

- ◆ No previous experience in test automatization
- ◆ Heard about different test tools in previous test conferences
- ◆ But with what to start test automatization?
- ◆ What should I look for?
- ◆ What is important in test automatization?



# ATLM in practice

- ◆ ATLM covers extensive information not only about test automatization but also about automatization of testing process overall
- ◆ It generates too much documentation what sometimes can take too much effort. Decide what you need.
- ◆ Not everything is useful in any situation. Take only what you need.
- ◆ It is not possible to automatize everything in one short run. Set your priorities and achieve them step by step.

# How ATLM helped?

- ◆ Before the start of test automatization, understand what it can do
- ◆ Be familiar with organization's needs and its engineering environment before you define criteria for evaluation of the test tools
- ◆ Make initial research of available test tools and rate them
- ◆ If test team till now has performed only manual testing, then be sure there is a developer available for help
- ◆ Evaluate top 2 or 3 highest rated test tools. Note your comments. Make comments on basic steps to create test.

# TestComplete

The screenshot displays the TestComplete application window. The title bar reads "TestComplete - C:\Documents and Settings\Winda\My Documents\TestComplete 6 Projects\ParskatuProjekts\_1\ParskatuProjekts\_1.pjs". The interface includes a menu bar (File, Edit, View, Script, Tools, Help), a toolbar, and a "Project Workspace" section with "Object Browser" and "Project Explorer". The "Project Explorer" shows a tree view of the project "ParskatuProjekts\_1" with sub-items like "CreateReport", "EditReport", "LogIn", "SendReport", "SlodzesTests", "ParskatuProjekts\_1 Analysis", and "ParskatuProjekts\_1 Logs". The "Workspace" area shows a tree view of the current test "CreateReport" with sub-items "ProjectTestItem1" and "Test Log [Unit1\Main]". The "Test Log" window is open, displaying a table of test events:

Type	Message	Priority	Time	H...	Link
	The 'e-Pärskats (v2.0) - Microsoft Internet Explorer' window was activated.	Normal	14:35...		
	The window was clicked with the left mouse button.	Normal	14:35...		
	Selected 'Ievedums-1B' from the combo box.	Normal	14:35...		
	Selected 'maijs' from the combo box.	Normal	14:35...		
	The window was clicked with the left mouse button.	Normal	14:35...		
	The window was clicked with the left mouse button.	Normal	14:35...		
	Keyboard input.	Normal	14:35...		

Below the test log, there is an "Information" section showing "Errors: 0" and "Warnings: 0", along with "Start Time: 14:35 2009.02.24." and "End Time: 14:35 2009.02.24.". A "Remarks" section contains a screenshot of a web page with the following text: "Sys[\"Process\"](\"IEXPLORE\")[\"Page\"](\"http://guppy.isoft.lv/eSurvey/UI/RespQuestList.aspx\")[\"Panel\"](\"popupPage\")[\"Panel\"](1)[\"Table\"](0)[\"Cell\"](0, 0)[\"Table\"](0)[\"Cell\"](0, 0)[\"Frame\"](\"PopupPageFrame\")[\"Panel\"](\"container\")[\"Form\"](\"as pnetForm\")[\"Panel\"](\"tab\")[\"Table\"](0)[\"Cell\"](0, 1)[\"Select\"](\"ctl00\_ContentMain\_uxQuest\")". A "Picture" window shows a screenshot of the web page. At the bottom, the "Call Stack" window is open, showing "Call Stack" and "process not accessible". The status bar at the bottom includes "Project Explorer", "Code Explorer", "Bookmarks", "Search/Replace Results", "Call Stack", "Watch List", "Locals", and "Breakpoints".

# Visual Studio Team System 2008 Test Edition

The screenshot displays the Visual Studio Team System 2008 Test Edition interface. The main window shows a project named "Parskats" with a solution explorer on the left. The solution explorer is expanded to show a "Scenarios" folder, which contains a "Scenario1" folder. Inside "Scenario1", there is a "Test Mix" folder containing three test steps: "[44%] Login", "[37%] CreateSurvey", and "[19%] EditSurvey". Below "Test Mix" is a "Browser Mix" folder containing "[76%] Internet Explorer 7.0" and "[24%] Firefox 2.0". Further down is a "Network Mix" folder containing "[34%] LAN", "[33%] Dial-up 56k", and "[33%] Cable/DSL 768k". At the bottom of the solution explorer is a "Step Load Pattern" folder. Below the solution explorer is a "Counter Sets" folder, a "Run Settings" folder, and a "Run Settings1 [Active]" folder. The "Run Settings1" folder contains "Counter Set Mappings", "Controller Computer", "LoadTest", "Controller", "Agent Computers", and "Agent".

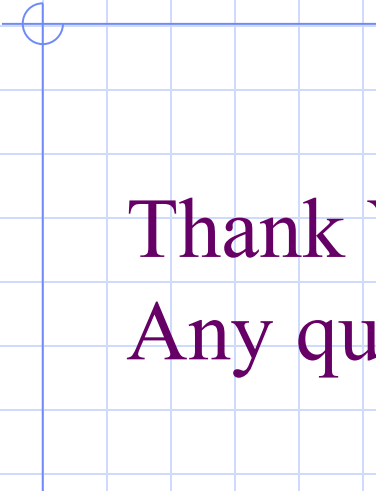
The right-hand side of the interface shows the "Properties" window for the selected "Step Load Pattern". The properties are:

- Microsoft.VisualStudio.TestTools.WebStress.Step
- Load Pattern
  - Pattern
  - Step
- Parameters
  - Initial User Count: 10
  - Maximum User Count: 203
  - Step Duration (second): 15
  - Step Ramp Time (second): 0
  - Step User Count: 15

The bottom of the interface shows the "Test Results" window. The test run is completed, with the following results:

Result	Test Name	Project	Error Message
Passed	CreateSurvey	Parskats	

The "Test Results" window also shows a summary: "Test run completed Results: 1/1 passed; Item(s) checked: 0".



**Thank You for attention!**  
**Any questions?**