TRUONG DAI HOC KHOA HOC TU NHIEN

KHTN

TP. HO CHI MINH

UNIVERSITY OF SCIENCE
HO CHI MINH CITY

Introduction to Test Automation

Nguyen V. Vu

Topics covered

- Overview of test automation
- Typical test automation process
- Testing approaches
- Automation tools

Test Automation

- Test automation refers to the use of software tools to execute tests

- Automated testing tools can enter data, run tests, compare results, and report test results

- Automated testing vs. manual testing
  - Manual testing: tests are performed by humans
  - Automated testing: tests are performed by computers

Why Test Automation?

- Manual testing is time and cost consuming
- Automation testing shorten tests and project duration
- Difficult to do manual testing in some situations
  - Multi lingual sites
  - Performance test
  - Security test
- Automation helps increase test coverage
- Manual testing can become tedious and error prone

Benefits of Test Automation - 1

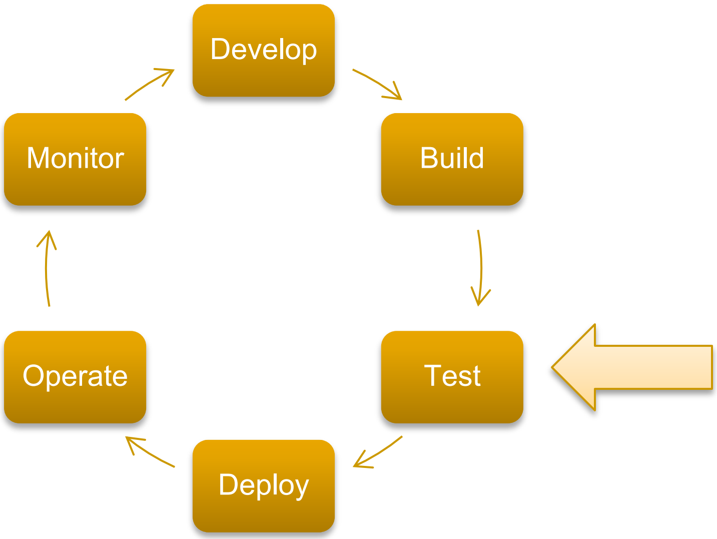
- Saves Time and Cost
  - Higher ROI (Return on Investment)
- Faster than the manual testing
  - Which one?
- Early time to market
  - Better speed in executing tests
- Reusable testing
  - Which one?

Benefits of Test Automation - 2

- Wider test coverage of application features
  - Test coverage?
  - Why?
- Reliable in results
  - Why?
- Improves accuracy
- Test more frequently and thoroughly
  - Why?

Test automation is essential for DevOps

- DevOps – a practice for making steps from Development to Operation easy and quick
- DevOps reduces software lifecycle time significantly
- It becomes a trend today



Must be automation tests

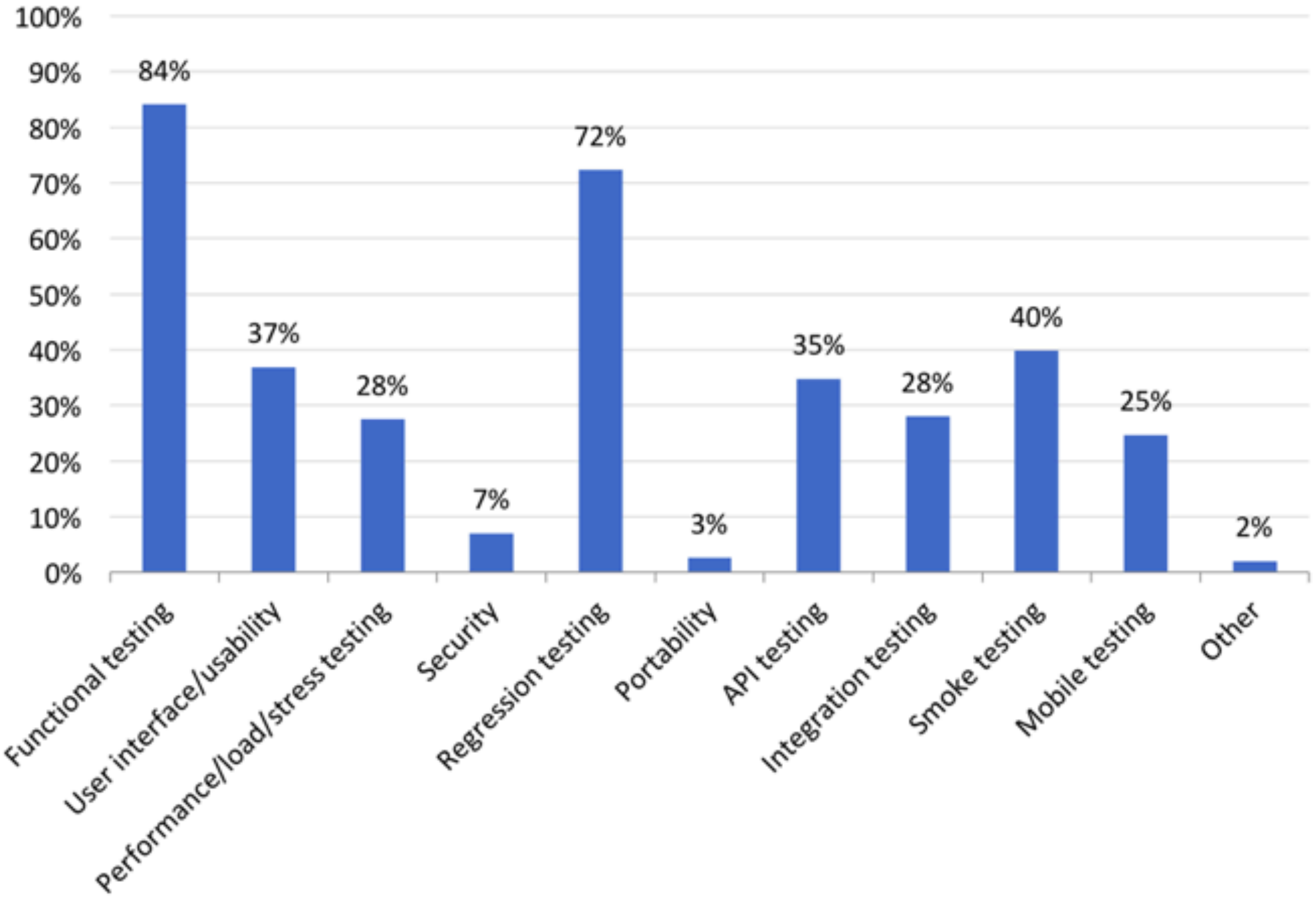
When Test Automation Works Best?

- Test automation works best for some testing
  - High risk, business critical test cases
  - Test cases that are executed repeatedly
    - Such as regression test
  - Test cases that are very tedious or difficult to perform manually
  - Time consuming test cases
  - Performance tests
  - Load tests
  - Security tests

When Test Automation Works Best?

- Generally, REGRESSION testing
- Regression testing
  - Testing functionality that has been tested before in earlier iterations

Testing types using automation

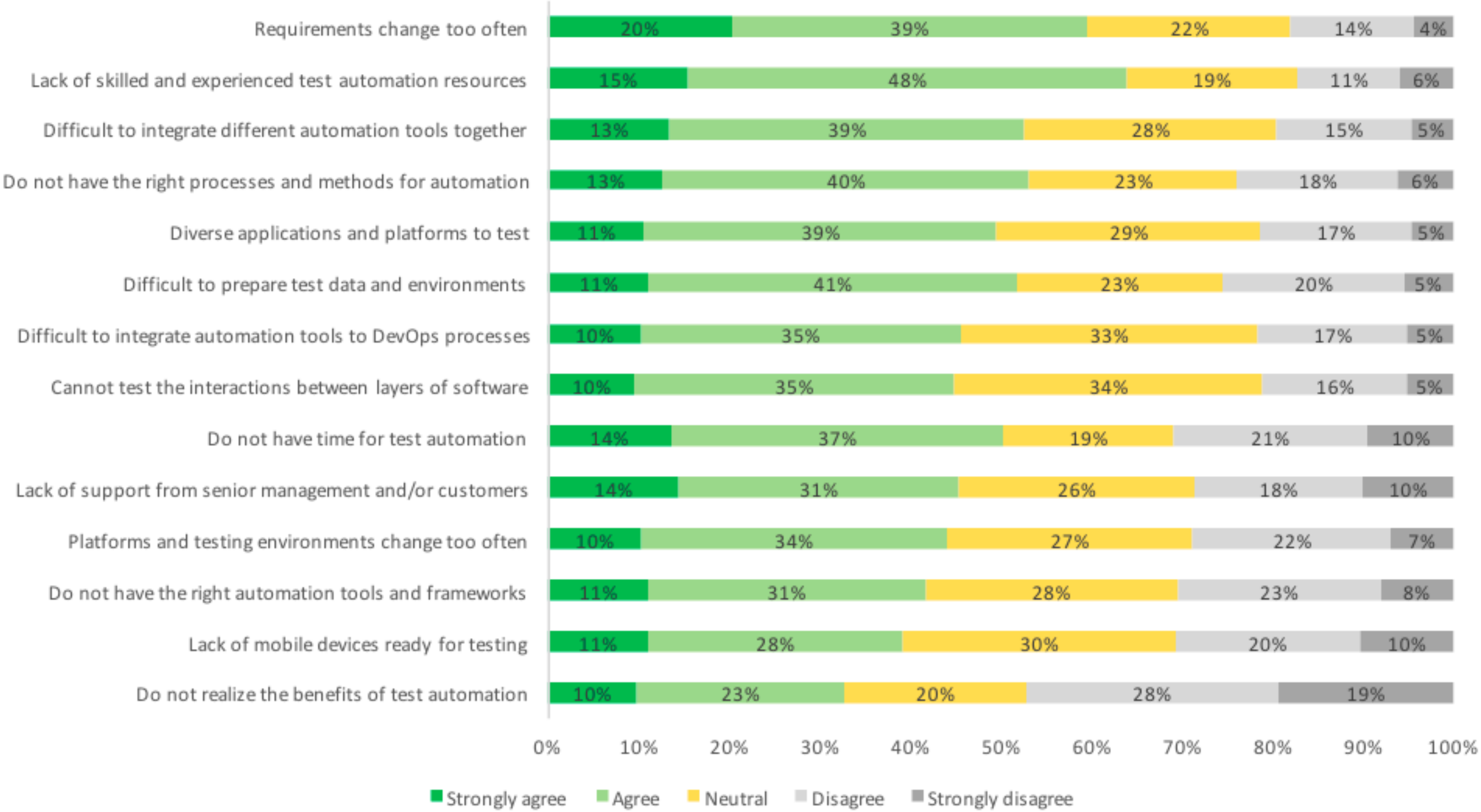


Source: “The most striking problems in test automation: A survey”, 2018. Katalon.com

When Test Automation Not Suitable?

- Newly designed test cases
  - Test cases should be manually tested at least once
- Test cases for frequently changed requirements
- Ad-hoc test cases
- UI/UX testing
  - Dependent on human judgment and experience

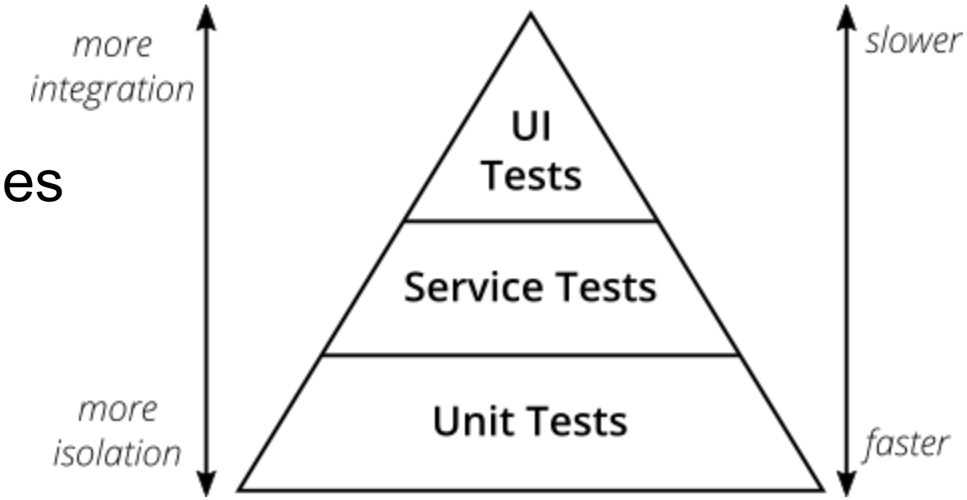
Challenges of test automation



Source: “The most striking problems in test automation: A survey”, 2018. Katalon.com

Levels of test automation

- Unit testing
  - Methods, functions, classes
- Integration testing
  - Integration multiple parts
- System testing
  - Focusing on UI, user’s features



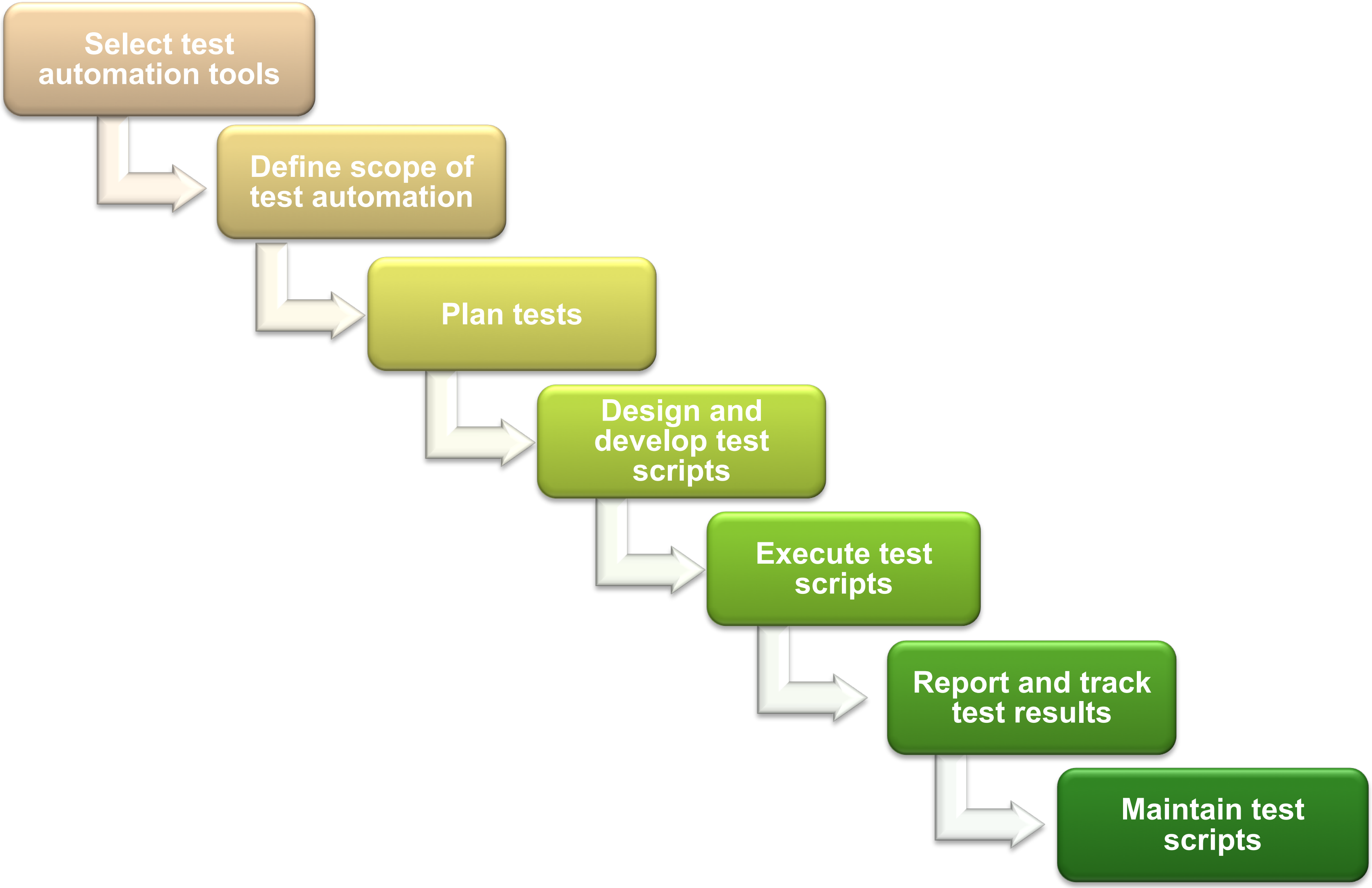
The test Pyramid

Source: martinfowler.com

Topics covered

- Overview of test automation
- Typical test automation process
- Testing approaches
- Automation tools

Typical Test Automation Process

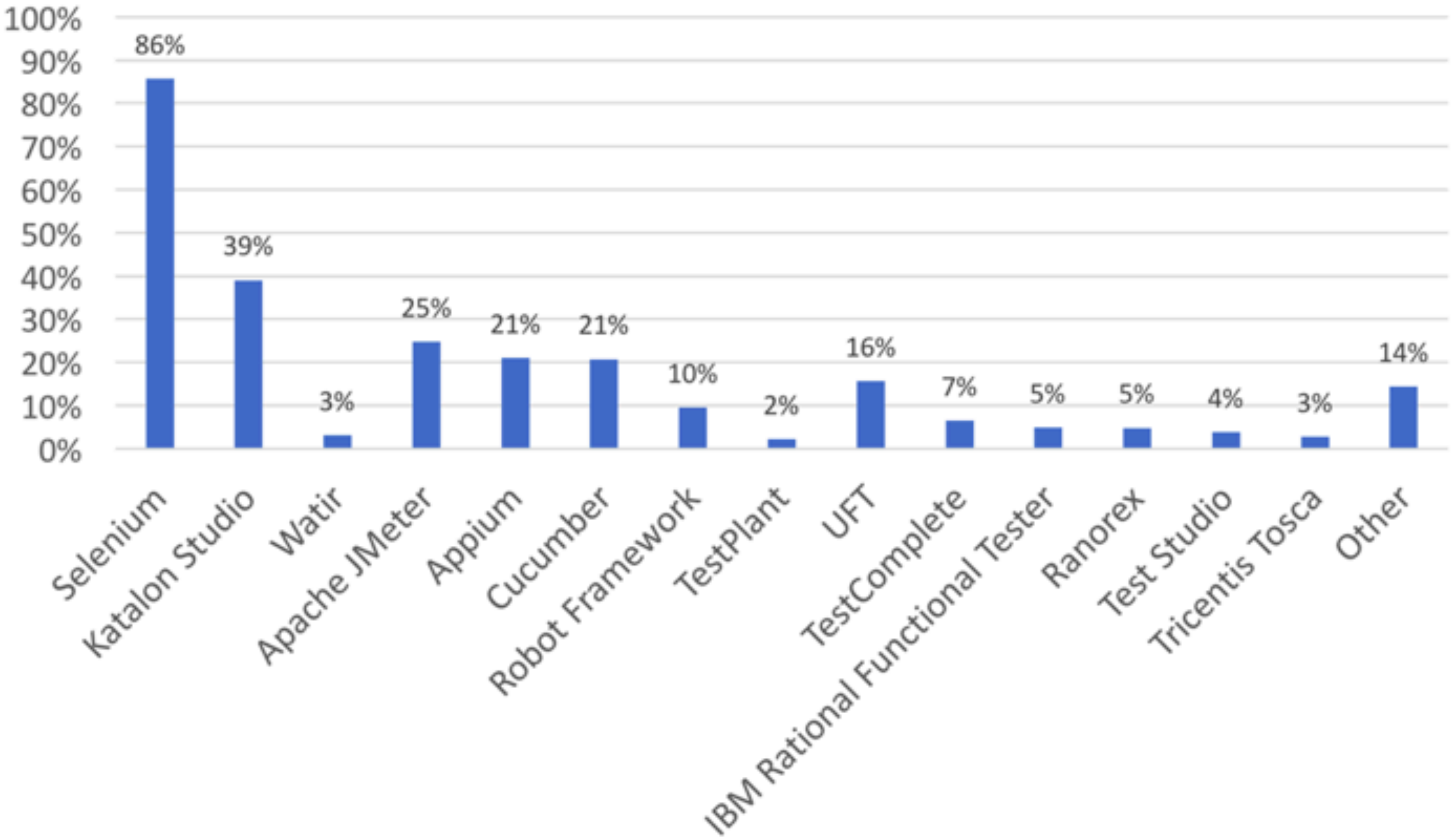


Select test automation tools
Define scope of test automation
Plan tests
Design and develop test scripts
Execute test scripts
Report and track test results
Maintain test scripts

Select Automation Tools

- Selecting tools suitable for applications under test (AUT) is very challenging
- Types of tool
  - Commercial: some are powerful but expensive
  - Open source: free but limited functionality
- Criteria to evaluate
  - Budget
  - Easy of use
  - Scripting languages
  - Platform (Windows, Unix, Mac OS, etc)
  - Training
  - Team experience

Most popular tools



Source: “The most striking problems in test automation: A survey”, 2018. Katalon.com

Automation Tool Evaluation

Criteria	Maveryx	Selenium	Cucumber	TestComplete	Ranorex	UFT (QTP)
Open source	Yes	Yes	Yes	No	No	No
Platform	Windows, Linux and Mac	Cross-platform	Cross-platform	Windows (mainly), Android, iOS	Windows (mainly), Android, iOS	Windows (mainly)
AUT programming languages	Java	Web-based languages	Ruby, Java, .NET, Flex or web applications	Many	Many	Many
Scripting languages	Java	Many (Java, C#, Perl, Python, etc.)	Ruby, Java, C#	VBScript, Jscript, DelphiScript, C++Script, C#Script	C#, VB.NET	VBScript
Support	7 (very active)	9	7	9	8	9
Usability	8 (easy to use)	7	6	9	8	9
Script maintainability	6 (smart object detection)	3 (tool provides little support)	1 (almost no support by tool)	7	6	7 (smart object detection and correction)
Required programming skills	7 (high level)	6 (support with record/ playback)	5	7 (strong record/playback capabilities)	6	8
Automated testing approaches	Data-driven, keyword-driven	Record/ playback, keyword-driven, data-driven	Structured, keyword-driven, data-driven	Record/ playback, keyword-driven, data-driven	Record/ playback, keyword-driven, data-driven	Record/ playback, keyword-driven, data-driven
Cost	Free (community edition), from €940/license (professional edition)	Free	Free	From $999/license	From €609/ license, one year maintenance	From $8000/ license

Source: https://www.katalon.com/resources-center/blog/a-structured-evaluation-for-selecting-a-right-automated-testing-tool/

Define Scope of Test Automation

- Which areas in AUT to be automated
- Which areas to be tested manually

- Areas to be considered
  - Business important features
  - Scenarios which have large amount of data
  - Common functionalities across applications
  - Reused business components
  - Complexity of test cases
  - Test cases for cross browser testing

Plan tests

- Define test plans and strategies for testing
  - Tools to be used
  - Test approaches
    - Functional, non-functional, usability, performance, security, testing, ...
    - Automation testing approaches
    - Unit testing, system testing, integration testing, acceptance testing
  - Schedule and timeframe
  - Staffing
  - Strategies for testing
    - Manual testing vs. automation testing, areas to test, test environments, etc.

Design and Develop Test Scripts

- Design test cases
- Design test data
- Design and develop test framework and test scripts
- Evaluate test scripts

- Similar to programming

- Test framework: a set of rules for automation
- Test framework consists of software systems, function libraries, reusable modules, data source, etc.

Execute Tests, Report and Track Test Results

- Run test scripts on AUT
- Test execution is usually controlled by the automation tool
- Test results are compared with expected results
- Test results are reported
- Defects are captured and entered to defect management systems

Maintain Test Scripts

- Test scripts have to be maintained often as software changes
  - New functions are added, updated
  - Requirements are changed
- As developers change source code, test scripts may not work anymore
  - Parameters are changed
  - GUI objects are changed
  - Outputs are changed
- Maintaining test scripts can be very time consuming

Topics covered

- Overview of test automation
- Typical test automaion process
- Automation testing approaches
- Automation tools

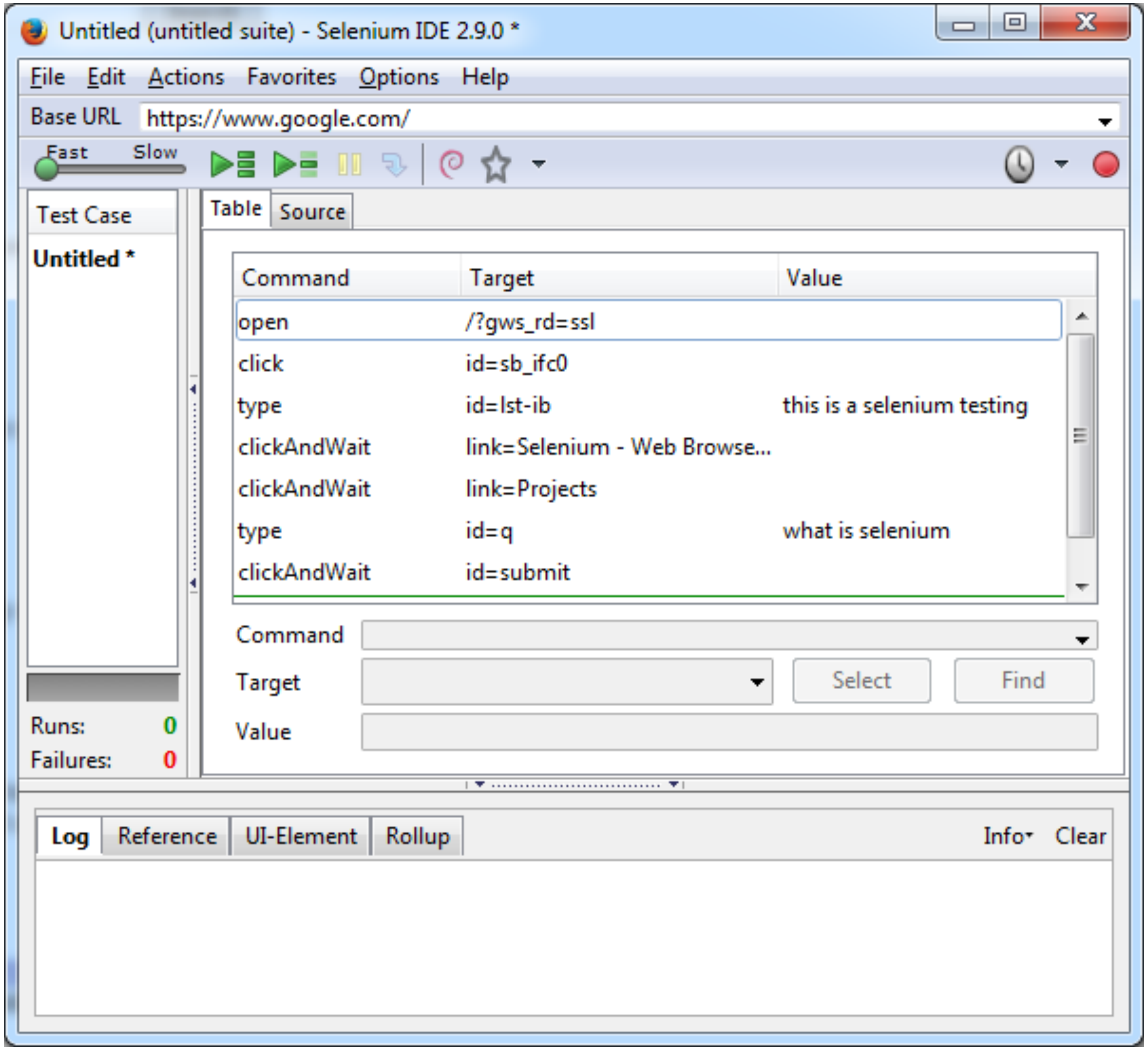
Scripting Approaches

- Record and playback
- Linear scripting
- Modular scripting
- Data-driven testing
- Keyword-driven testing

Record and Playback

- Tools record users’ actions performed on AUT
- Often, scripts are generated after recording
- Tools playback what is recorded
- Popular feature in many automation tools

Selenium’s Record and Playback



Untitled (untitled suite) - Selenium IDE 2.9.0 *
File Edit Actions Favorites Options Help
Fast Slow
Test Case
Untitled *
Table Source
Select
Find
Log Reference UI-Element Rollup Info▾ Clear

Base URL https://www.google.com/

Command	Target	Value
open	/?gws_rd=ssl	
click	id=sb_ifc0	
type	id=lst-ib	this is a selenium testing
clickAndWait	link=Selenium - Web Browse...	
clickAndWait	link=Projects	
type	id=q	what is selenium
clickAndWait	id=submit	

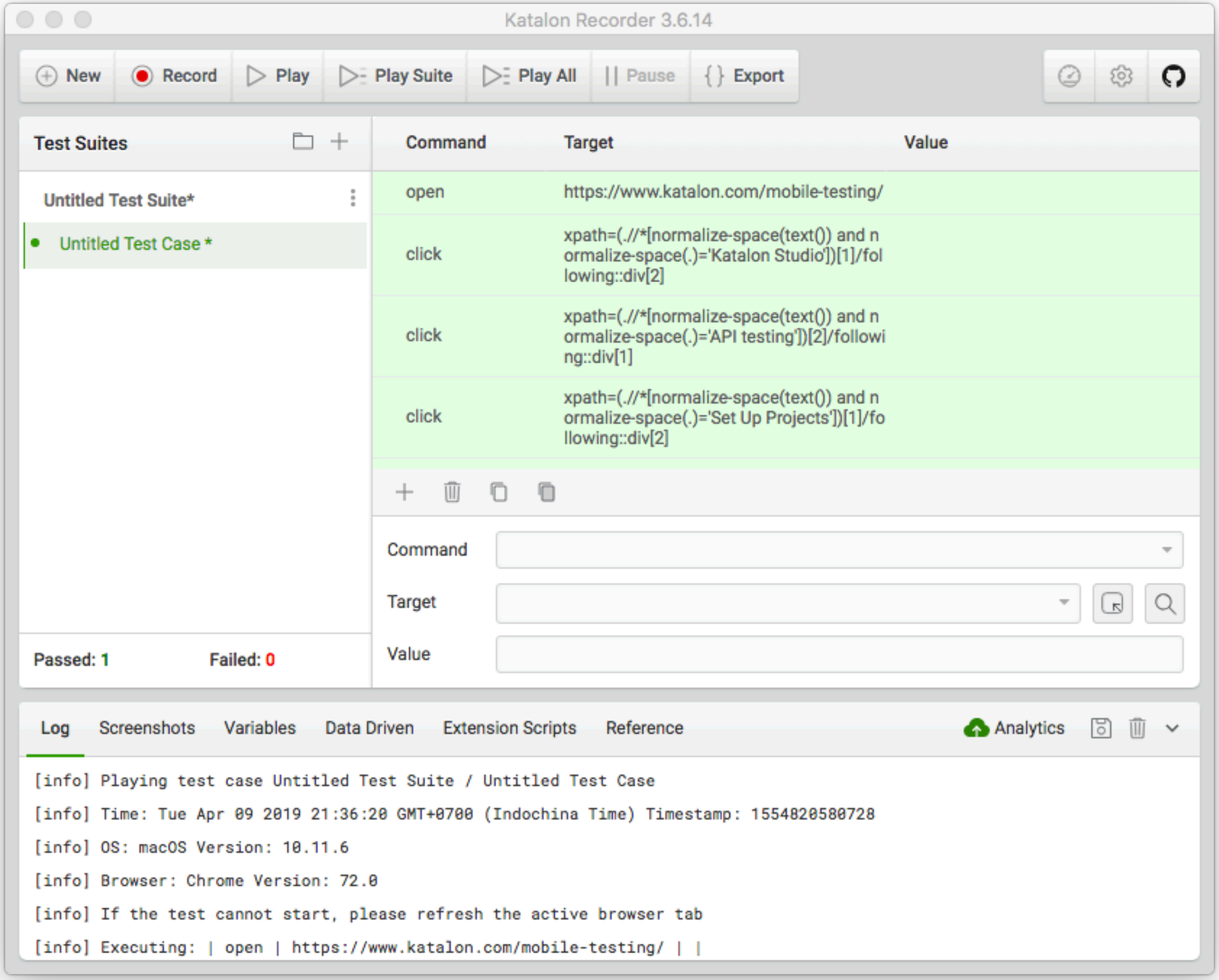
Command

Target

Value

Runs: 0
Failures: 0

Katalon Recorder



Command	Target	Value
open	https://www.katalon.com/mobile-testing/	
click	xpath=(.//*[normalize-space(text()) and normalize-space(.)='Katalon Studio'])[1]/following::div[2]	
click	xpath=(.//*[normalize-space(text()) and normalize-space(.)='API testing'])[2]/following::div[1]	
click	xpath=(.//*[normalize-space(text()) and normalize-space(.)='Set Up Projects'])[1]/following::div[2]	

Passed: 1 Failed: 0

```
[info] Playing test case Untitled Test Suite / Untitled Test Case
[info] Time: Tue Apr 09 2019 21:36:20 GMT+0700 (Indochina Time) Timestamp: 1554820580728
[info] OS: macOS Version: 10.11.6
[info] Browser: Chrome Version: 72.0
[info] If the test cannot start, please refresh the active browser tab
[info] Executing: | open | https://www.katalon.com/mobile-testing/ | |
```

Record and Playback (cont'd)

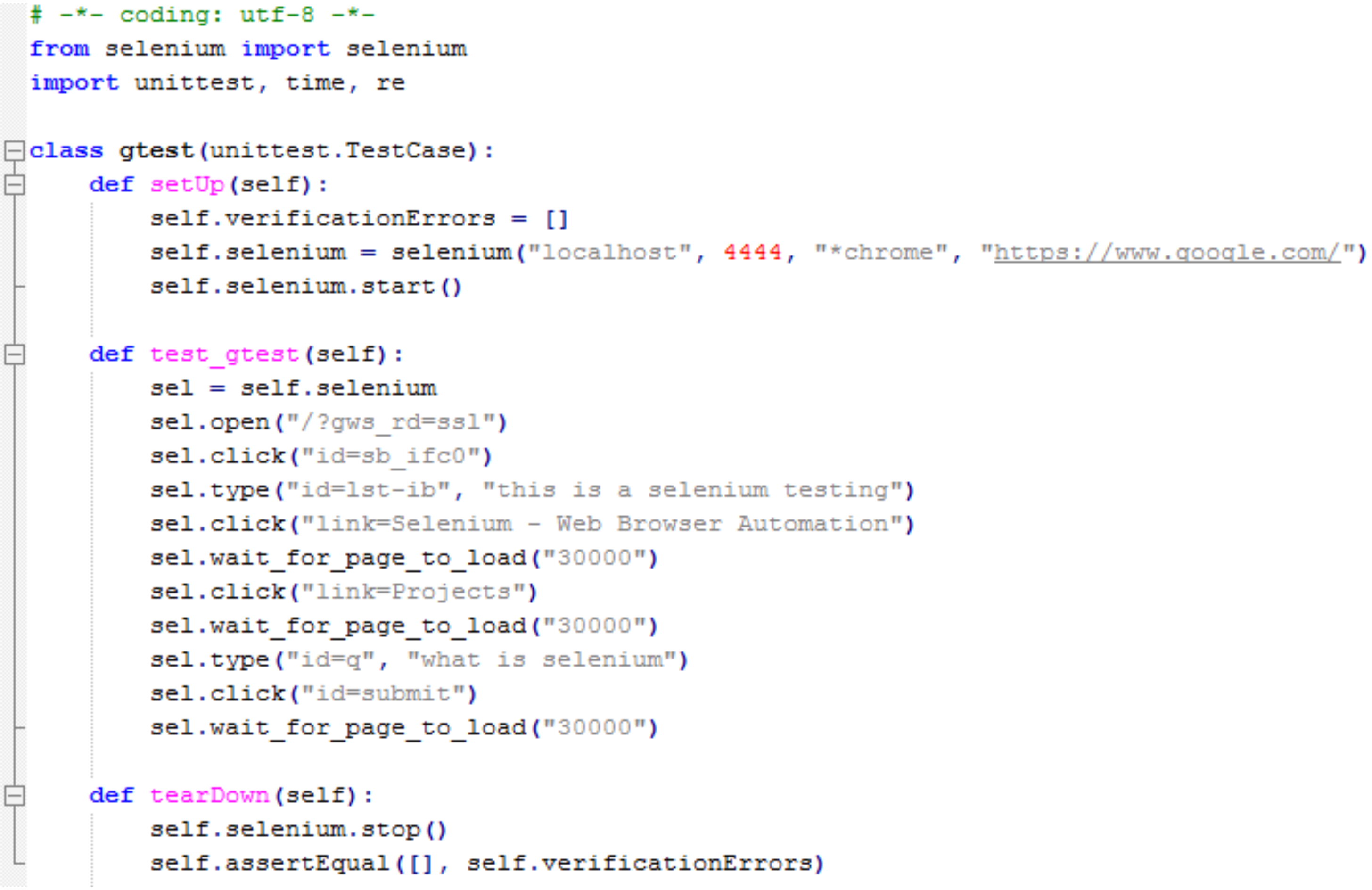
- Advantages
  - Easy to use
  - No programming skills needed
  - Good for learning
- Problems
  - Scripts can only be created when AUT is ready
  - Does not test anything. Checkpoints are needed to test
  - Only suitable with UI testing
  - Small changes in UI can cause tests to stop
  - Hard to manage and maintain
    - Lots of test scripts are created
- Not a good approach for advanced testing

Linear Scripting

- Scripts are created to test AUT
- Use some programming languages
- Scripts are also created with Record and Playback
- A test project can have many test suites
  - A test suite has one or more test cases

- Good for simple test cases
- Difficult for large automation

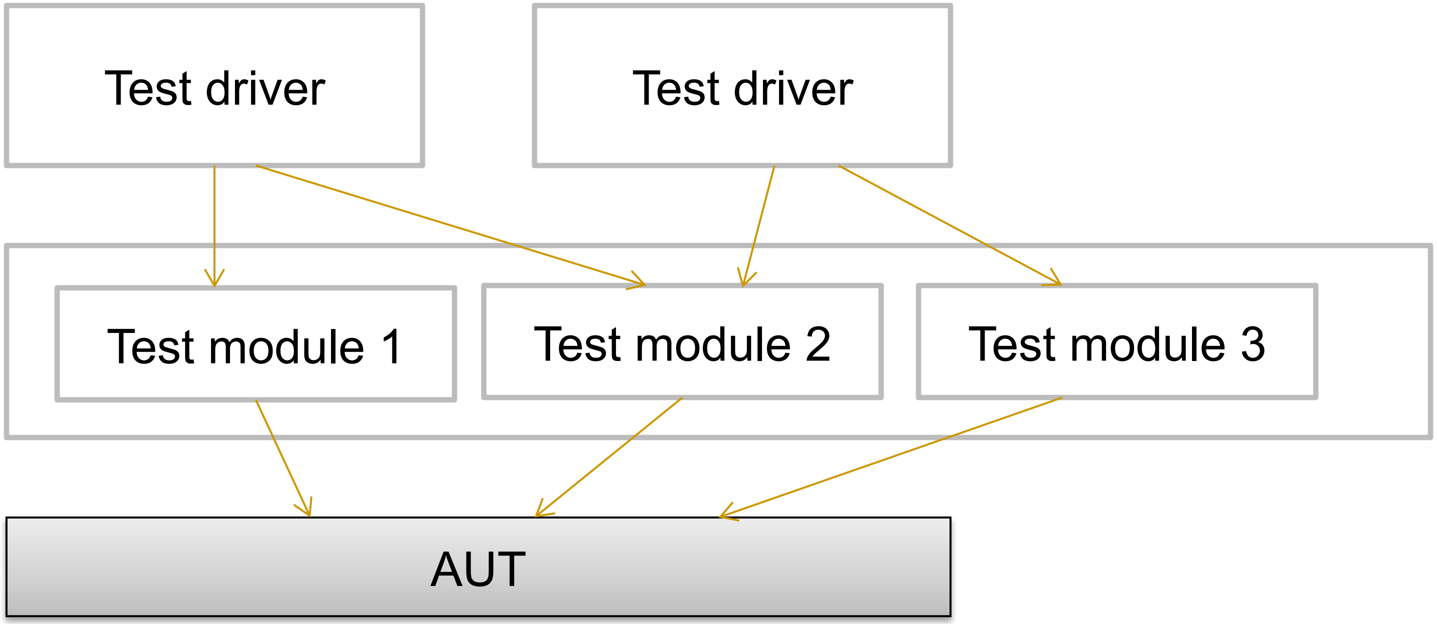
Linear Scripting (cont’d)



```
# -*- coding: utf-8 -*-
from selenium import selenium
import unittest, time, re

class gtest(unittest.TestCase):
    def setUp(self):
        self.verificationErrors = []
        self.selenium = selenium("localhost", 4444, "*chrome", "https://www.google.com/")
        self.selenium.start()

    def test_gtest(self):
        sel = self.selenium
        sel.open("/?gws_rd=ssl")
        sel.click("id=sb_ifc0")
        sel.type("id=lst-ib", "this is a selenium testing")
        sel.click("link=Selenium - Web Browser Automation")
        sel.wait_for_page_to_load("30000")
        sel.click("link=Projects")
        sel.wait_for_page_to_load("30000")
        sel.type("id=q", "what is selenium")
        sel.click("id=submit")
        sel.wait_for_page_to_load("30000")

    def tearDown(self):
        self.selenium.stop()
        self.assertEqual([], self.verificationErrors)
```

Modular Scripting

- Place test scripts into functions or modules
- Use drivers to call these functions or modules to execute scripts on AUT



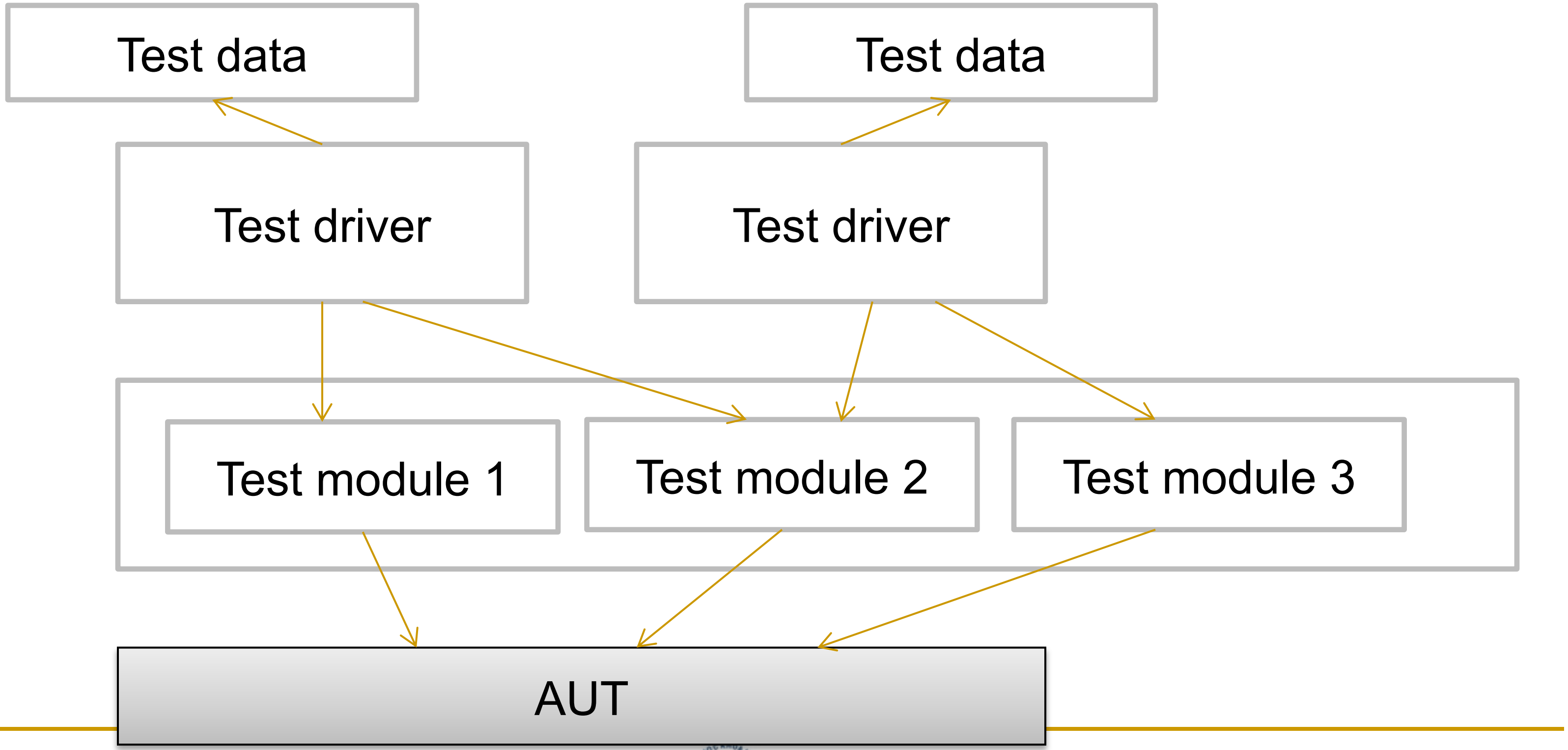
Modular Scripting (cont'd)

- Advantages: just like procedural programming
  - Reuse of scripts
  - Driver scripts are simple
  - Easy to maintain than linear scripting
- Disavantages
  - Requiring building libraries/functions initially
  - Test data is embedded directly in scripts
    - Not flexible in changing data

- Requiring programming skills
- Working with simple as well as large tests

Data-Driven Scripting/Framework

- Data is separated from test scripts
- Test scripts read data automatically from files
- Can test multiple data items, different inputs



Data-Driven Scripting (cont’d)

- Test driver scripts can use multiple data items
  - E.g., testing different username and password combinations
  - Flexible in changing inputs
- Programming skills not needed when updating data
- Good for separating roles
  - Developers responsibile for scripting
  - Testers responsible for test data

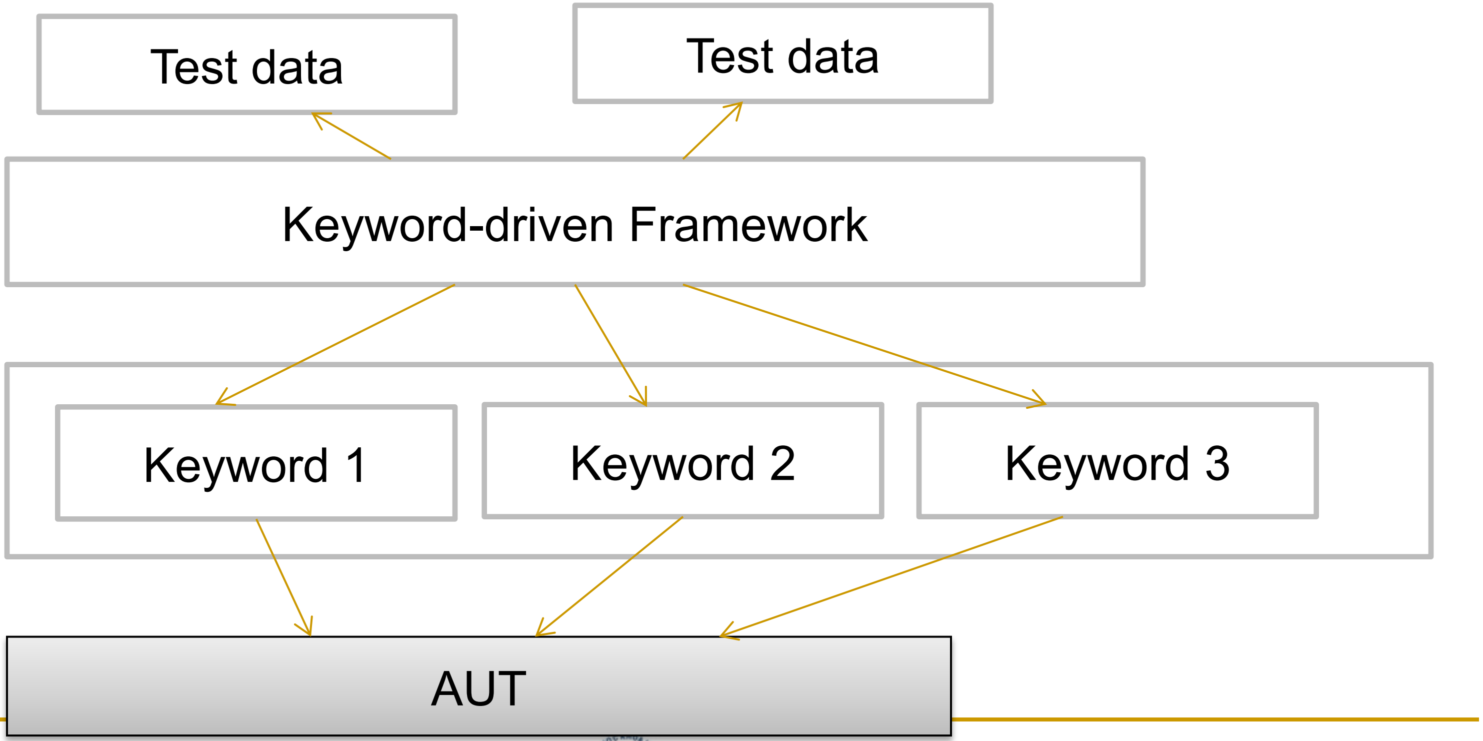
Data-Driven Scripting (cont'd)

- Disavantages
  - Initial effort is required to create framework (parser, libraries, etc)
    - Just like building a new tool
  - New tests need new driver scripts
  - Require programming skills to create new driver scripts

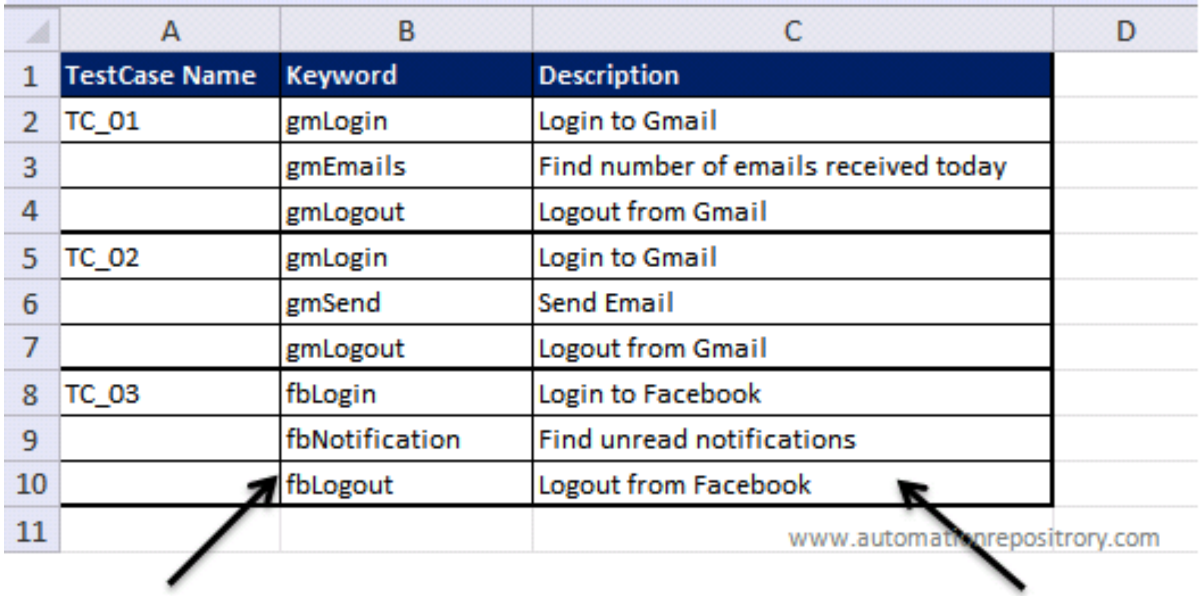
- Good solution for large scale automation

Keyword-Driven Testing

- Separate test scripts from
  - test data
  - directives (keywords) on how to use data
- Keywords and data drive test execution



Keyword-Driven Testing (cont’d)



	A	B	C	D
1	TestCase Name	Keyword	Description	
2	TC_01	gmLogin	Login to Gmail	
3		gmEmails	Find number of emails received today	
4		gmLogout	Logout from Gmail	
5	TC_02	gmLogin	Login to Gmail	
6		gmSend	Send Email	
7		gmLogout	Logout from Gmail	
8	TC_03	fbLogin	Login to Facebook	
9		fbNotification	Find unread notifications	
10		fbLogout	Logout from Facebook	
11				

www.automationrepositrory.com

Each keyword associated with the test case is mentioned.

For ease of understanding, proper description is provided with each keyword

Keyword-Driven Testing (cont'd)

- Advantages
  - All tests can be handled by one framework
  - Tests can be created from keywords
  - Non-programmers can create tests
  - Separate test data and scripts
- Disadvantages
  - Effort required to build framework upfront
  - Programming skills needed to build framework
- Good solution supported by many commercial tools

Combination of the approaches

- Advanced tools today allow to combine multiple approaches
- Tools integrate recorder, scripts, keywords, data
- Tools
  - Katalon Studio
  - TestComplete
  - UFT
  - Ranorex

Topics covered

- Overview of test automation
- Typical test automaion process
- Testing approaches
- Automation tools

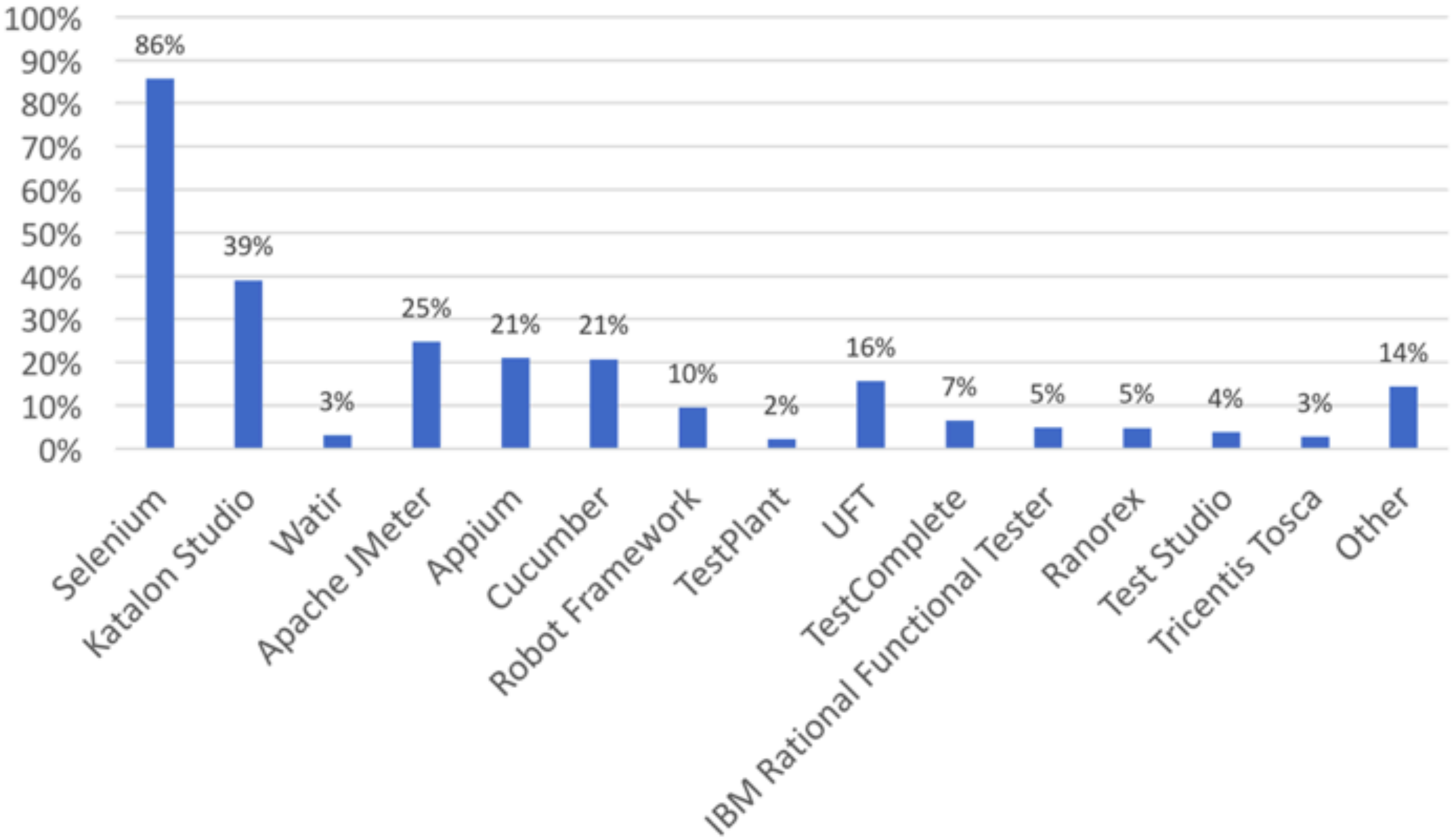
Test Automation Tools

- Hundreds of test automation tools available in market
- Commercial tools
  - Usually powerful
  - Well supported
  - Rich feature set
  - Some are expensive, unluckily
- Open source
  - Free, of course
  - Many are good to use, others not that good
  - Support uncertain

Test Automation Tools (cont'd)

- Popular commercial tools
  - HP QTP/UFT
  - TestComplete
  - Ranorex
  - Rational Robot
  - Rational Functional Tester
  - eggPlant
- Popular open source and/or free tools
  - Selenium
  - Katalon Studio
  - Cucumber
  - Maveryx

Most popular tools



Source: “The most striking problems in test automation: A survey”, 2018. Katalon.com

Want to Work in Test Automation?

- Programming skills are a plus
  - Scripting languages: Python, Ruby, Java, C#, etc.
- Regular expressions
- SQL
- Learning to use test automation tools
  - Selenium
  - Katalon Studio

Conclusions

- Test automation is now a trend in software development
  - Customers require to apply test automation
  - Demand is high
- Test automation is essential for DevOps
- It does not replace manual testing