CERTIFIED TEST AUTOMATION PROFESSIONAL (CTAP) COURSEWARE

Rob Flier

Van Haren PUBLISHING

CTAP FOUNDATION Certified by Van Haren | certN

Certified Test Automation Professional (CTAP)
Courseware

Colophon

Title:	Certified Test Automation Professional (CTAP) Courseware
Authors:	Rob Flier
Publisher:	Van Haren Publishing, ‘s-Hertogenbosch
ISBN Hard Copy:	978 94 018 1024 1
Edition:	First edition, first print, April, 2023
Design:	Van Haren Publishing, ‘s-Hertogenbosch
Copyright:	© Van Haren Publishing 2023

For further information about Van Haren Publishing please e-mail us at: info@vanharen.net or visit our website: www.vanharen.net

Publisher about the Courseware

The Courseware was created by experts from the industry who served as the author(s) for this publication. The input for the material is based on existing publications and the experience and expertise of the author(s). The material has been revised by trainers who also have experience working with the material. Close attention was also paid to the key learning points to ensure what needs to be mastered.

The objective of the courseware is to provide maximum support to the trainer and to the student, during his or her training. The material has a modular structure and according to the author(s) has the highest success rate should the student opt for examination. The Courseware is also accredited for this reason, wherever applicable.

In order to satisfy the requirements for accreditation the material must meet certain quality standards. The structure, the use of certain terms, diagrams and references are all part of this accreditation. Additionally, the material must be made available to each student in order to obtain full accreditation. To optimally support the trainer and the participant of the training assignments, practice exams and results are provided with the material.

Direct reference to advised literature is also regularly covered in the sheets so that students can find additional information concerning a particular topic. The decision to leave out notes pages from the Courseware was to encourage students to take notes throughout the material.

Although the courseware is complete, the possibility that the trainer deviates from the structure of the sheets or chooses to not refer to all the sheets or commands does exist. The student always has the possibility to cover these topics and go through them on their own time. It is recommended to follow the structure of the courseware and publications for maximum exam preparation.

The courseware and the recommended literature are the perfect combination to learn and understand the theory.

-- Van Haren Publishing

Other publications by Van Haren Publishing

Van Haren Publishing (VHP) specializes in titles on Best Practices, methods and standards within four domains:

- IT and IT Management
- Architecture (Enterprise and IT)
- Business Management and
- Project Management

Van Haren Publishing is also publishing on behalf of leading organizations and companies: ASLBiSL Foundation, BRMI, CA, Centre Henri Tudor, Gaming Works, IACCM, IAOP, IFDC, Innovation Value Institute, IPMA-NL, ITSqc, NAF, KNVI, PMI-NL, PON, The Open Group, The SOX Institute.

Topics are (per domain):

IT and IT Management

ABC of ICT
ASL®
CATS CM®
CMMI®
COBIT®
e-CF
ISO/IEC 20000
ISO/IEC 27001/27002
ISPL
IT4IT®
IT-CMF™
IT Service CMM
ITIL®
MOF
MSF
SABSA
SAF
SIAM™
TRIM
VeriSM™

Enterprise Architecture

ArchiMate®
GEA®
Novius Architectuur
Methode
TOGAF®

Business Management

BABOK® Guide
BiSL® and BiSL® Next
BRMBOK™
BTF
EFQM
eSCM
IACCM
ISA-95
ISO 9000/9001
OPBOK
SixSigma
SOX
SqEME®

Project Management

A4-Projectmanagement
DSDM/Atern
ICB / NCB
ISO 21500
MINCE®
M_o_R®
MSP®
P3O®
PMBOK® Guide
Praxis®
PRINCE2®

For the latest information on VHP publications, visit our website: www.vanharen.net.

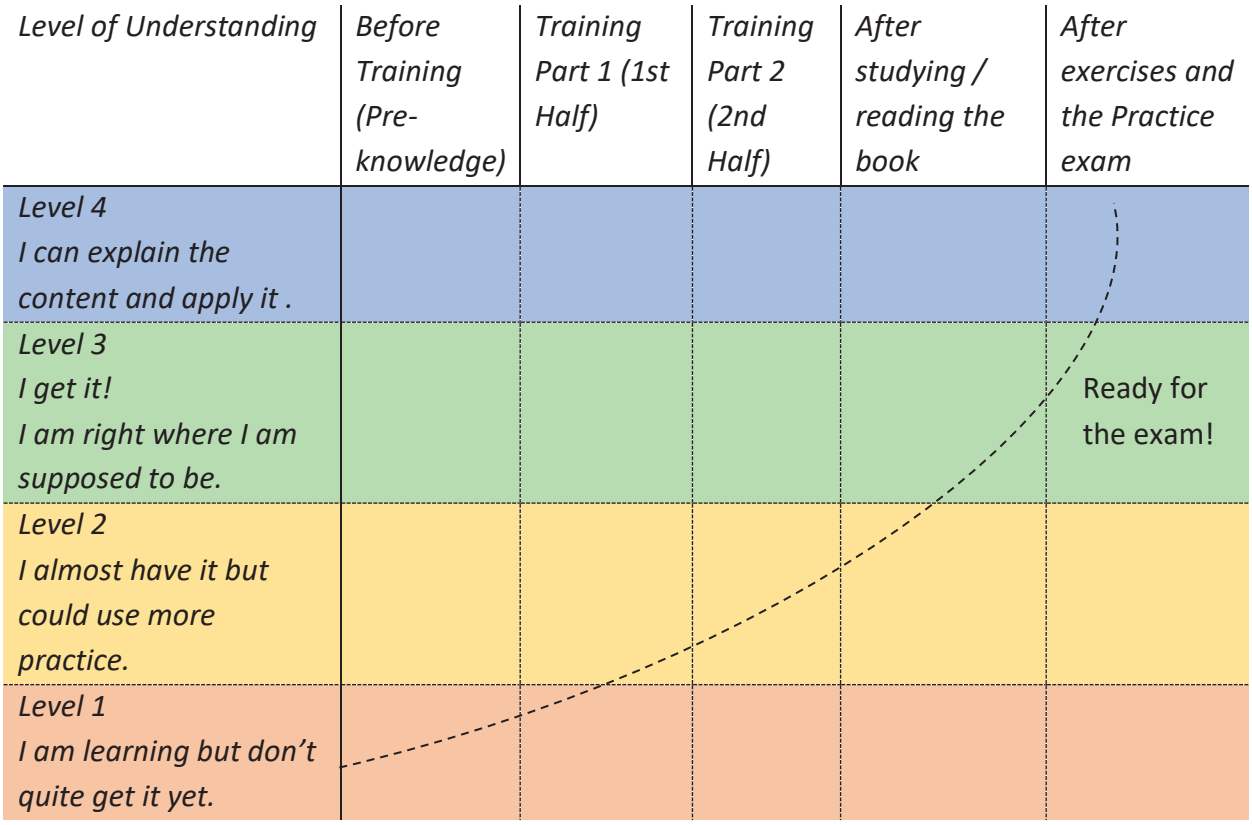
Table of content

Self-Reflection of understanding Diagram

'*What you do not measure, you cannot control*.'' – Tom Peters

Fill in this diagram to self-evaluate your understanding of the material. This is an evaluation of how well you know the material and how well you understand it. In order to pass the exam successfully you should be aiming to reach the higher end of Level 3. If you really want to become a pro, then you should be aiming for Level 4. Your overall level of understanding will naturally follow the learning curve. So, it's important to keep track of where you are at each point of the training and address any areas of difficulty.

Based on where you are within the Self-Reflection of Understanding diagram you can evaluate the progress of your own training.



(Self-Reflection of Understanding Diagram)

Write down the problem areas that you are still having difficulty with so that you can consolidate them yourself, or with your trainer. After you have had a look at these, then you should evaluate to see if you now have a better understanding of where you actually are on the learning curve.

Troubleshooting

Problem areas:	*Topic:*
Part 1	
Part 2	
You have gone through the book and studied.	
You have answered the questions and done the practice exam.	

Agenda

Day 1

Testtool overview
- What is test automation (a goal in itself)?

Working with tools for the
- Graphical User Interface
- Programming Interface

Testtool selection
- Tool adoption
- Tool selection

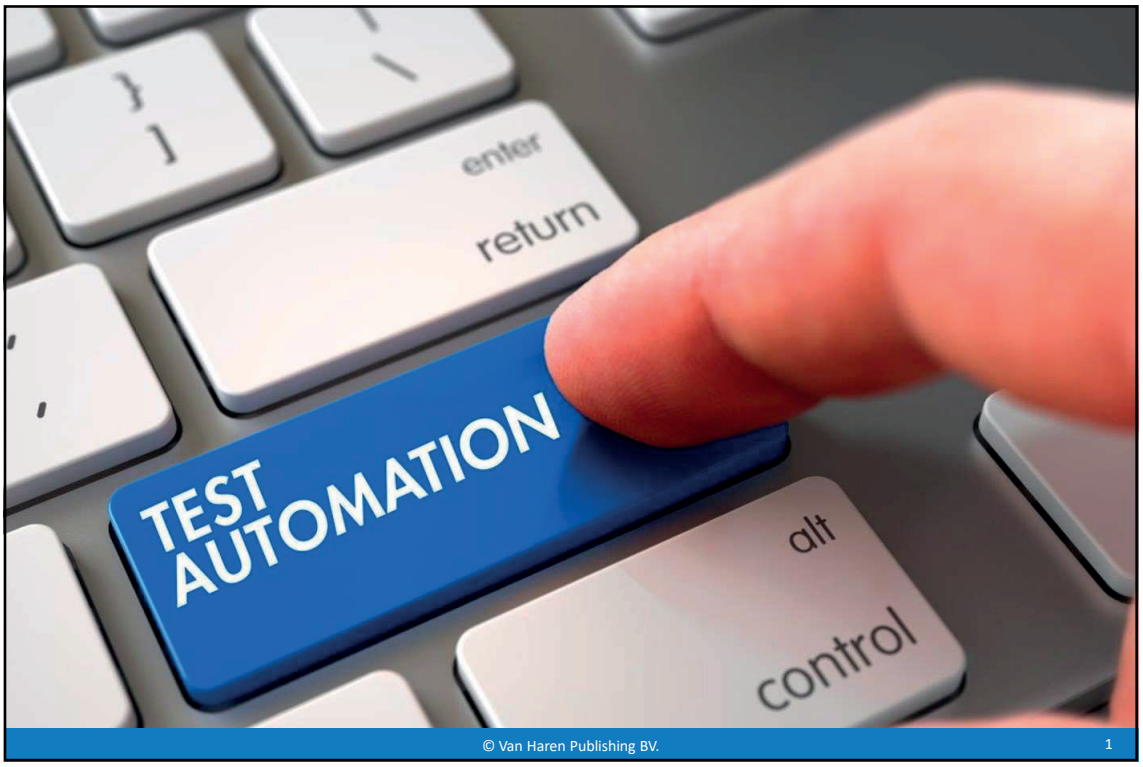
Working with tools for the Graphical User Interface
- Robot Framework

Day 2

Test roles

Continuous testing

Working with
- Robot Framework
- Cypress
- Browserstack

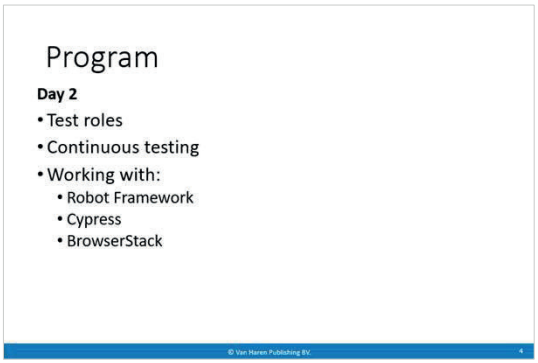


Topics

- What is test automation (is test automation a goal in itself)?
- Which test tool suits best?
- Test tool selection (considerations before you select a test tool)
- Various demos and exercises

Program

- **Day 1**
- Test tool overview
- Working with:
  - Graphical user interface
  - Programming interface
- Test tool selection
  - Tool adoption
  - Tool selection
- Working with:
  - Graphical user interface
  - Robot Framework



Program

Day 2

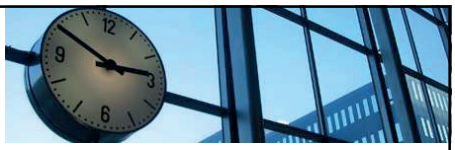
- Test roles
- Continuous testing
- Working with:
  - Robot Framework
  - Cypress
  - BrowserStack

The group

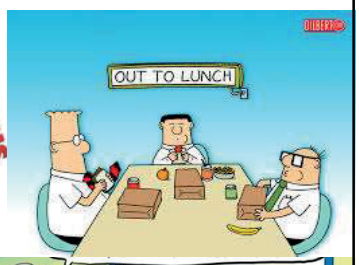
- Be open
- Be critical
- Be involved

 5

Announcements



- Time
- Breaks/lunch
- Disruptions: mobile phones
- Questions asked?
- Materials



 6

Materials

- Courseware
- Exercises
- Tools:
  - Selenium IDE plug-in
  - Postman
  - Robot Framework
  - Cypress

Introduction – let's get acquainted

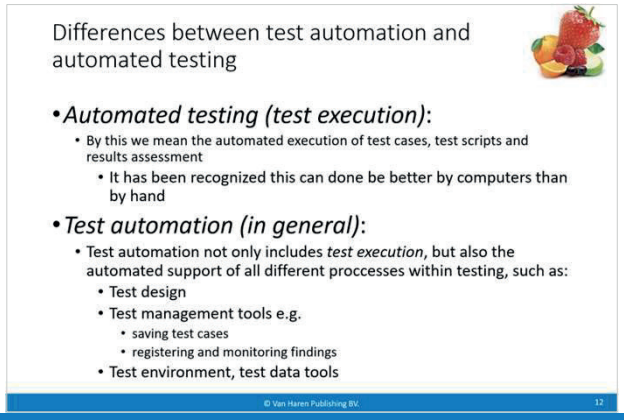
- Which tools do you use?
- Nothing yet?
  Choose the tool that best fits your wishes
- There are tools available for every budget
- It is important to determine in advance what you are going to test

Test tool overview

In the last couple of years the number of test tools has increased enormously

What is test automation?

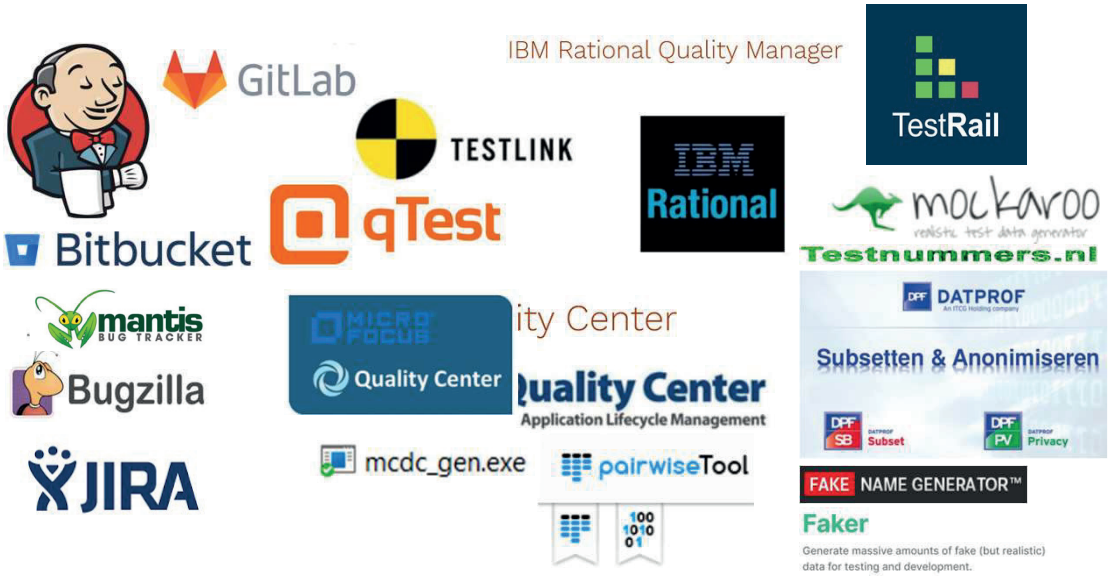
Test automation is that part of the test where you can use tools



Differences between test automation and automated testing

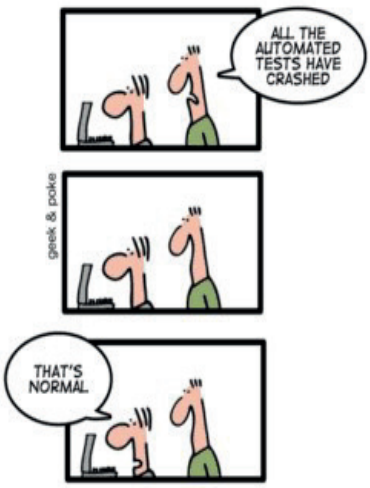
- *Automated testing (test execution)*:
  - By this we mean the automated execution of test cases, test scripts and results assessment
    - It has been recognized this can done be better by computers than by hand
- *Test automation (in general)*:
  - Test automation not only includes *test execution*, but also the automated support of all different proccesses within testing, such as:
    - Test design
    - Test management tools e.g.
      - saving test cases
      - registering and monitoring findings
    - Test environment, test data tools

Examples of test automation: test mgt tools



Examples of test automation: test execution tools

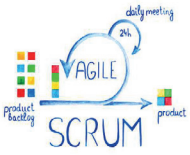
Why automated testing?



- Reasons:
  - *I don't want to do the same thing over and over*
  - *I want testing to run while I am asleep*
  - *I want regression testing*
  - *I want performance testing*
  - *Because it's fun!*

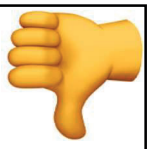
Automated testing becomes more important

- Due to the increasing need for shorter time-to-market
- By applying the Agile/Scrum method (incremental development of products)
- Through *DevOps* (Continuous Integration and Continuous Delivery)



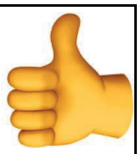
Why not start right away?

Counter arguments

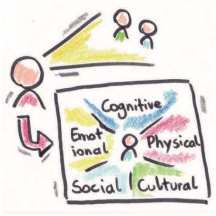


- Tooling requires the necessary expertise
- License costs
- Not every test script can be used multiple times
- Maintenance of test scripts
- Application is subject to too much change
- Creating a good test environment takes a lot of time and is expensive

Arguments in favor



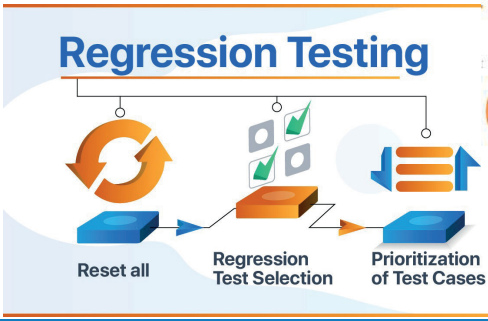
- Higher reliability by switching off the human factor
- Higher productivity allowing more test to be performed
- More job satisfaction due to automating routine activities

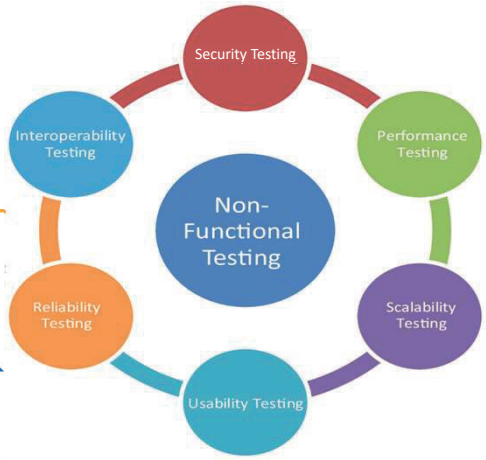


Arguments in favor

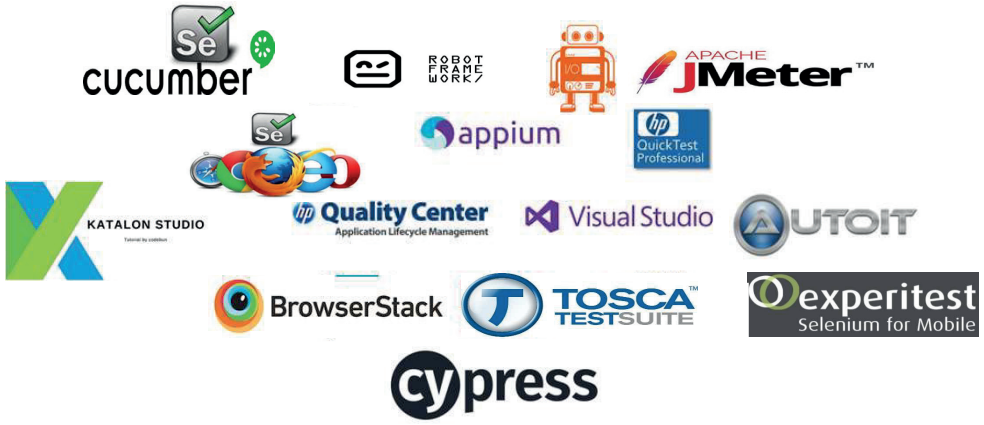
More testing options, for example:

performance- security- and *regression testing*





Working with tools for the graphical user interface

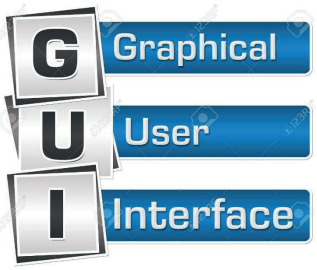


COURSEWARE

Test execution

Tools that can automate the test execution are available in several editions, for example:

- Via the graphical user interface
- Via a so-called program interface

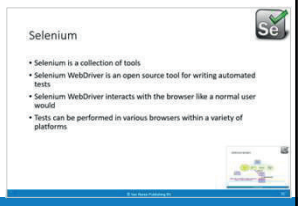


29

Through the graphical user interface (GUI)

- There are tools which work by recording actions so that they can be replayed
- These are also called *record & playback tools*
- (Open source) examples are:
  - Selenium IDE
  - Katalon Studio

Katalon

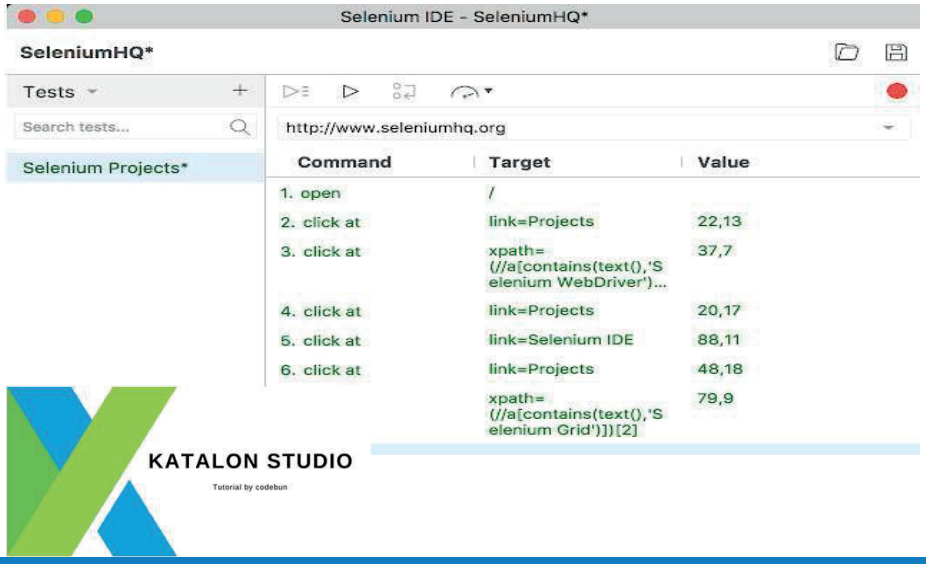


https://www.softwaretestinghelp.com/best-gui-testing-tools/

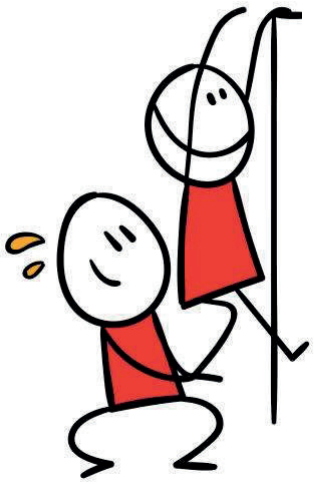
How do record and playback tools work?

- First start the test tool, then start the application to be tested and carry out some actions, such as entering a customer into the database
- Close the application and quit the test tool
- Then save the script generated by the test tool
- Now the test tool can play the script and the application to be tested starts; actions are performed to enter a customer in the database, the customer data is entered and the new customercard is saved

Selenium IDE & Katalon demo



How do record and playback tools work?



To work

Exercise:

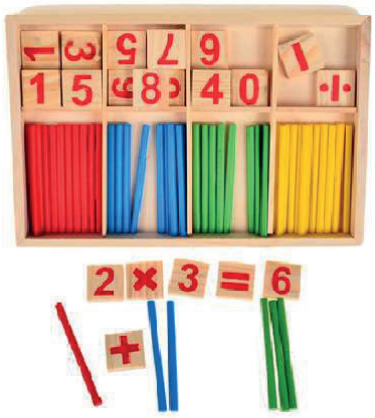
Open the plug-in in Google Chrome and create a new SAVINGS account. Then make a transfer and check the transfer on the overview page.

Userid: *john* Password: *demo*

Url: http://localhost:8080/parabank
Public url: https://parabank.parasoft.com/parabank/index.htm
Hint: Don't forget to log out

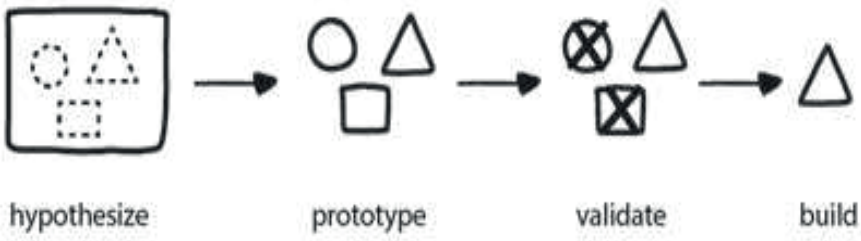
When might record and playback be a good option?

- As a learning tool:
  - It can be incredibly useful to record your steps and see what is generated
  - Reviewing these recorded steps can give you real examples of how the underlying automation framework can be leveraged from code



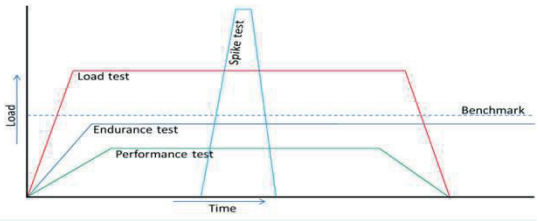
When might record & playback be a good option?

- As a simple proof of concept:
  - These tools can be used to see if an application can be automated
  - You can also use the generated scripts or steps to get a sense of where your pain points would be



When might record & playback be a good option?

- For load testing:
  - Many load testing tools do record & playback under the hood
  - Testers record test scenarios to capture the requests and responses between the client and server
  - At a very simple level they basically just capture your network traffic and play it back



And when not

- Limited test coverage:
  - Record and playback tools perform exactly the steps you have included, possibly including the typos or mis-clicks
  - That means these tests usually don't do much more than basic navigation tests
  - Navigation tests are important, but testing the navigation alone is not enough

Through the graphical user interface (GUI)

- There are also tools to fully automate the functional actions (*clicking a button on a website or opening an app on a phone*)
- Examples of keyword-driven test tools include:

WebDriver

WebdriverIO

appium

cypress

ROBOT FRAMEWORK

*… like a **mouse click,** selection of a **menu item**, **keystrokes**, **opening** or **closing** a window*

Working with tools through the programming interface

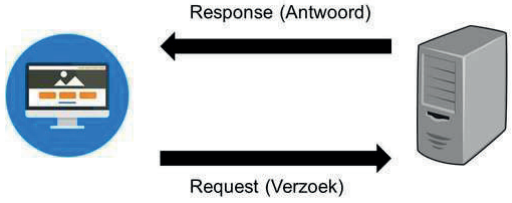
REST-assured
PARASOFT SOAtest
ROBOT FRAMEWORK
POSTMAN
SoapUI
cypress

COURSEWARE

Through a programming interface

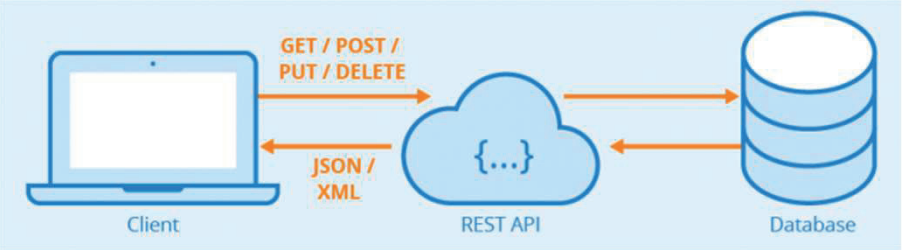
Tools that can automate the test execution are available in several variants, for example:

- Through the graphical user interface (GUI)
- Through a programming interface (API)



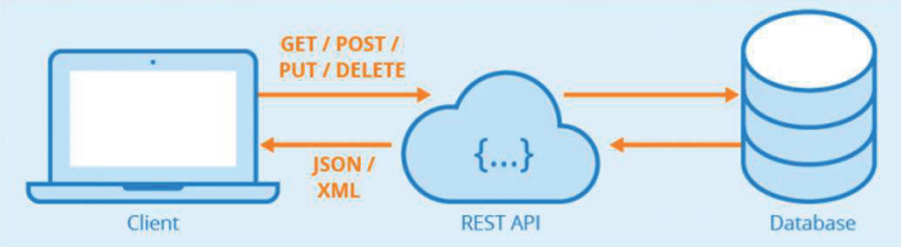
Through a programming interface

- In the context of APIs, the word ‘application’ refers to any software with a distinct function
- Interface can be thought of as a *contract* of service between two applications
- This *contract* defines how the two communicate with each other using requests and responses



Through a programming interface

- Examples of an *interface* are:
  - Application Programming Interface (API)
  - Messages in XML or JSON format (web service)



Through a programming interface

- The difference between a web service and an API is in the network
  - *A program interface* that communicates through the internet, based on the SOAP- or REST-protocol, is a *web service*
- *Web services* are made for online machine-to-machine communication
- In fact, it is only the front-end interfaces of websites and applications that reside on end-users' devices
- When it comes to the way the web is made, there are currently two competing approaches: *SOAP* and *REST*

Through a programming interface

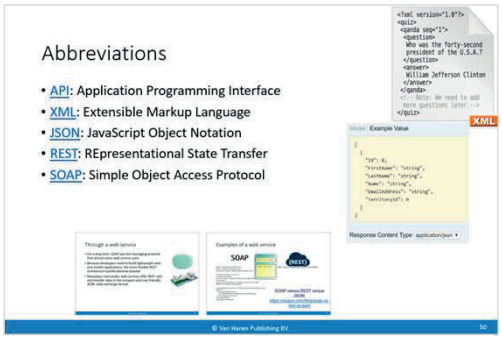
- Examples of tools for testing se*rvices*:

SoapUI

PARASOFT SOAtest

REST-assured

POSTMAN



Abbreviations

- API: Application Programming Interface
- XML: Extensible Markup Language
- JSON: JavaScript Object Notation
- REST: REpresentational State Transfer
- SOAP: Simple Object Access Protocol

```
<?xml version="1.0"?>
<quiz>
 <qanda seq="1">
  <question>
   Who was the forty-second
   president of the U.S.A.?
  </question>
  <answer>
   William Jefferson Clinton
  </answer>
 </qanda>
 <!-- Note: We need to add
  more questions later.-->
</quiz>
```

XML

Model | Example Value

```
[
  {
    "Id": 0,
    "FirstName": "string",
    "LastName": "string",
    "Name": "string",
    "EmailAddress": "string",
    "TerritoryId": 0
  }
]
```

Response Content Type application/json

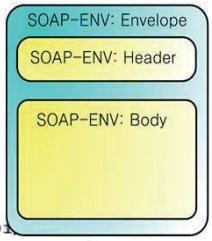
Through a web service

- *For a long time, SOAP* was the messaging protocol that almost every web service used
- Because developers need to build *lightweight* web- and mobile applications, the more flexible *REST*-architecture quickly became popular
- Nowadays most public web services offer REST calls and transfer data in the compact and user-friendly JSON- data exchange format

Examples of a web service

SOAP

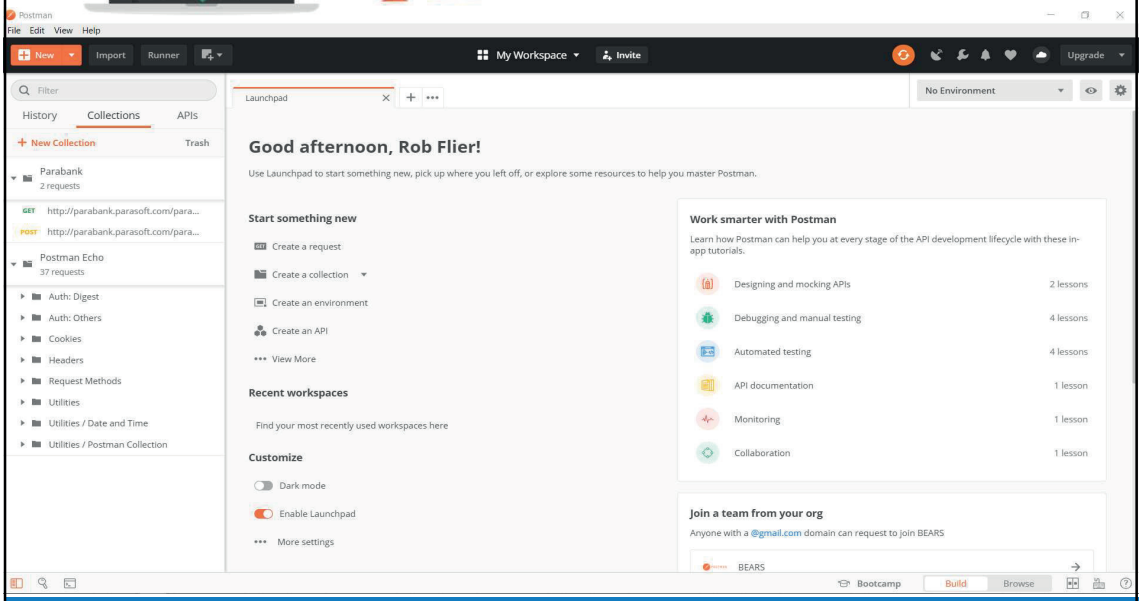
```
POST /GetStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001
envelope"
soap:encodingStyle="http://www.w3.org/
2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/
stock">
  <m:GetStockPrice>
    <m:StockName>IBM</m:StockName>
  </m:GetStockPrice>
</soap:Body>
</soap:Envelope>
```

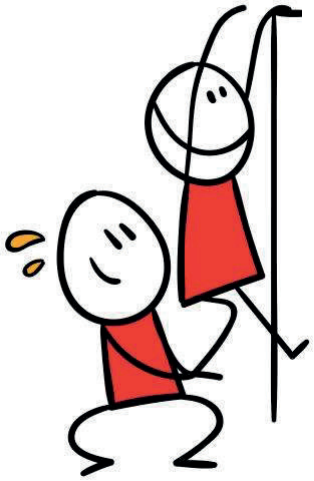


GET http://example.org/stock/IBM

SOAP versus REST versus JSON: https://raygun.com/blog/soap-vs-rest-vs-json/

POSTMAN





To work

Open Postman and execute the exercises below:

1. Request the transfer of an amount of money from one account to another account.
2. And if successful, make a request of all transactions that have been made today.
3. Finally, make a request to get a loan.

Examples of a web service

- http://api.zippopotam.us/us/90210
- Script:

```
<script type="text/javascript"> var client = new XMLHttpRequest(); client.open("GET", "http://api.zippopotam.us/us/90210", true); client.onreadystatechange = function() { if(client.readyState == 4) { alert(client.responseText); }; }; client.send(); </script>
```

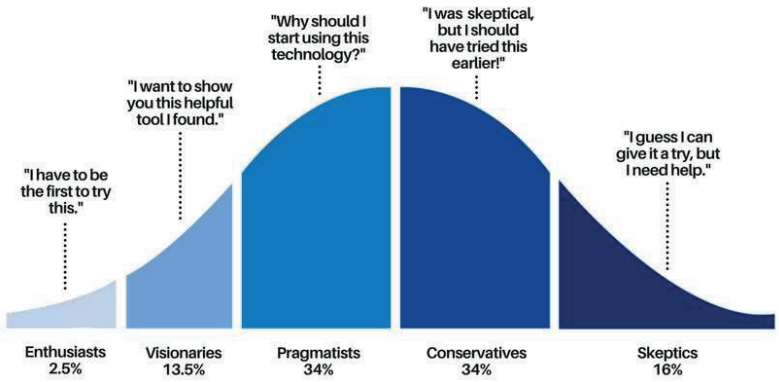
Bron: https://zippopotam.us/

Tutorials:

- https://www.guru99.com/postman-tutorial.html
- https://learning.getpostman.com/docs/postman/launching-postman/sending-the-first-request/

Test tool selection

Tool adoption & tool selection



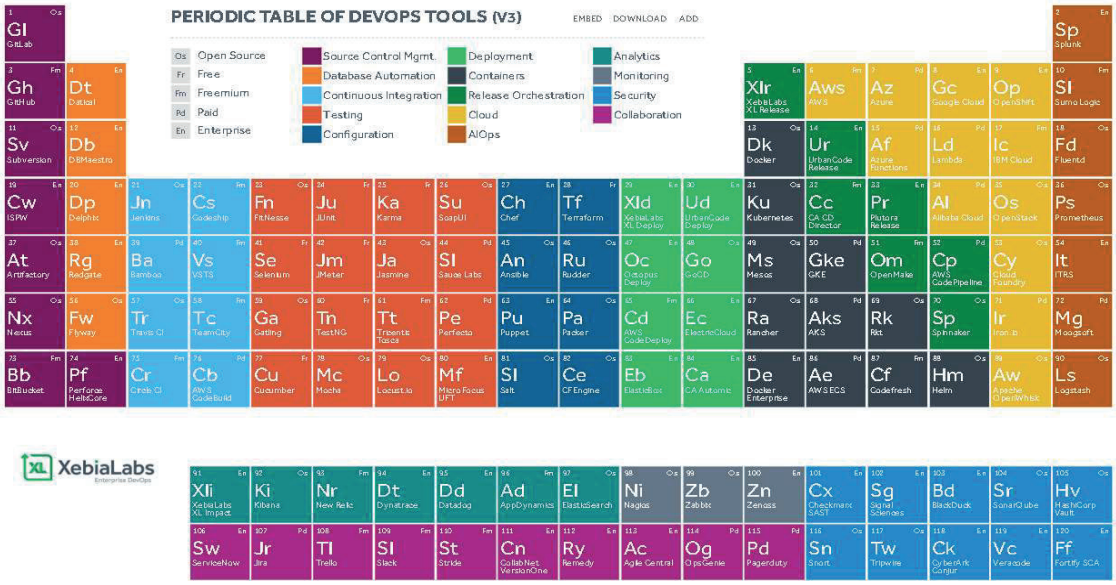
COURSEWARE

Test tool adoption

- The objectives of test automation and the technical environment are to determine which tools can be used
- A tool selection progress is, therefore, essential for selecting the right tool

COURSEWARE

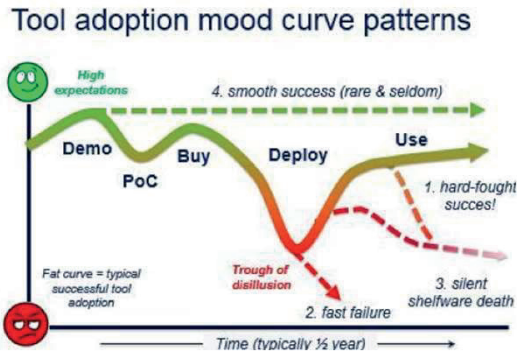
How do you find the right tool?



Tool adoption

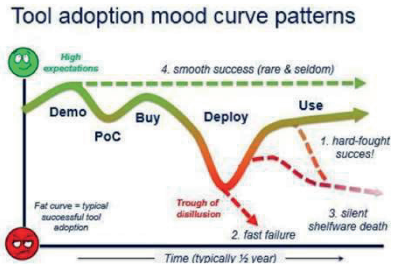
In practice, one of the following four patterns is almost always evident:

1. The hard-fought success
2. The rapid failure
3. The silent death
4. The smooth success



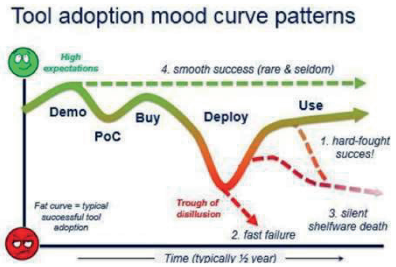
1. The hard-fought success

- Demos are always beautiful and make people happy
- A Proof-of-Concept is more realistic:
  - You actually use the tool in your own situation, with your application, your scripts and your data
  - There are also some disadvantages, so the mood drops a little
- Around the time of purchase everyone is positive again, then the real work begins



1. The hard-fought success

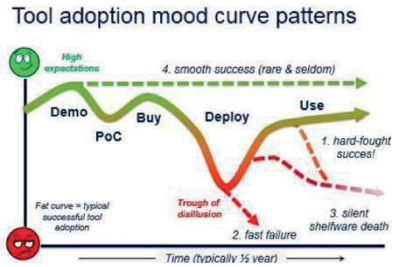
- Then comes the most dangerous moment: the valley of disillusion:
  - You will almost always have to cross this valley together
  - This is the moment when everyone threatens to lose heart:
    - it turns out to be more difficult than expected, we do not yet have the required knowledge and experience and even worse...
      - the tool contains serious bugs



Because, of course, test tools are also software and so they include errors

2. The rapid failure

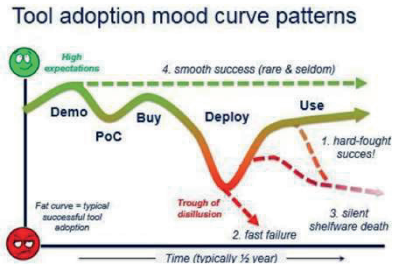
- What can also become clear in the valley of disillusion is that the choice of tool was an error
- In this case a rapid failure is the cheapest option



Crying and starting over

3. The silent dead

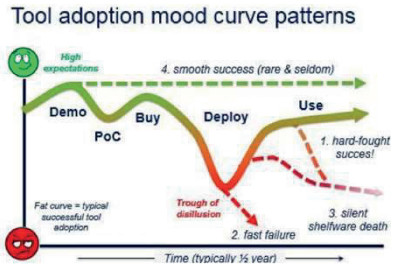
- They muddle on, things get better and a regression test set starts to work
- But maintaining the scripts proves problematic and the tool slowly disappears into the background



The silent dead is near!

4. The smooth success

- When counting your blessings, also take a look at the future stability and maintainability of the scripts
  - Because in time the problem will usually find itself there



- Knowledge and experience with tool selection and implementation is decisive in recognizing the patterns and making the right decisions at key moments in the process

Tool selection

KATALON STUDIO
REST-assured
APACHE JMeter™
WebDriver
ROBOT FRAMEWORK
POSTMAN
MOCHA
PARASOFT
SOAtest
appium
QuickTest Professional
Jasmine
mantis BUG TRACKER
Quality Center Application Lifecycle Management
Visual Studio
espresso
Bugzilla
Jest
Calaba.sh
SoapUI
Gatling LOAD TESTING
Atlassian JIRA
Silk Performer
SAUCELABS
BrowserStack
TOSCA TESTSUITE
experitest Selenium for Mobile
COURSEWARE