DevOps 2TIN

Chapter 7

Integrated Testing

DE HOGESCHOOL MET HET NETWERK

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be

Integrated testing

Soorten testen
Unittesten
Integratie testen
Functionele testen
Niet functionele testen
Test automatisatie
Testing in pipelines

HOGESCHOOL PXL

Wat is een Pipeline?

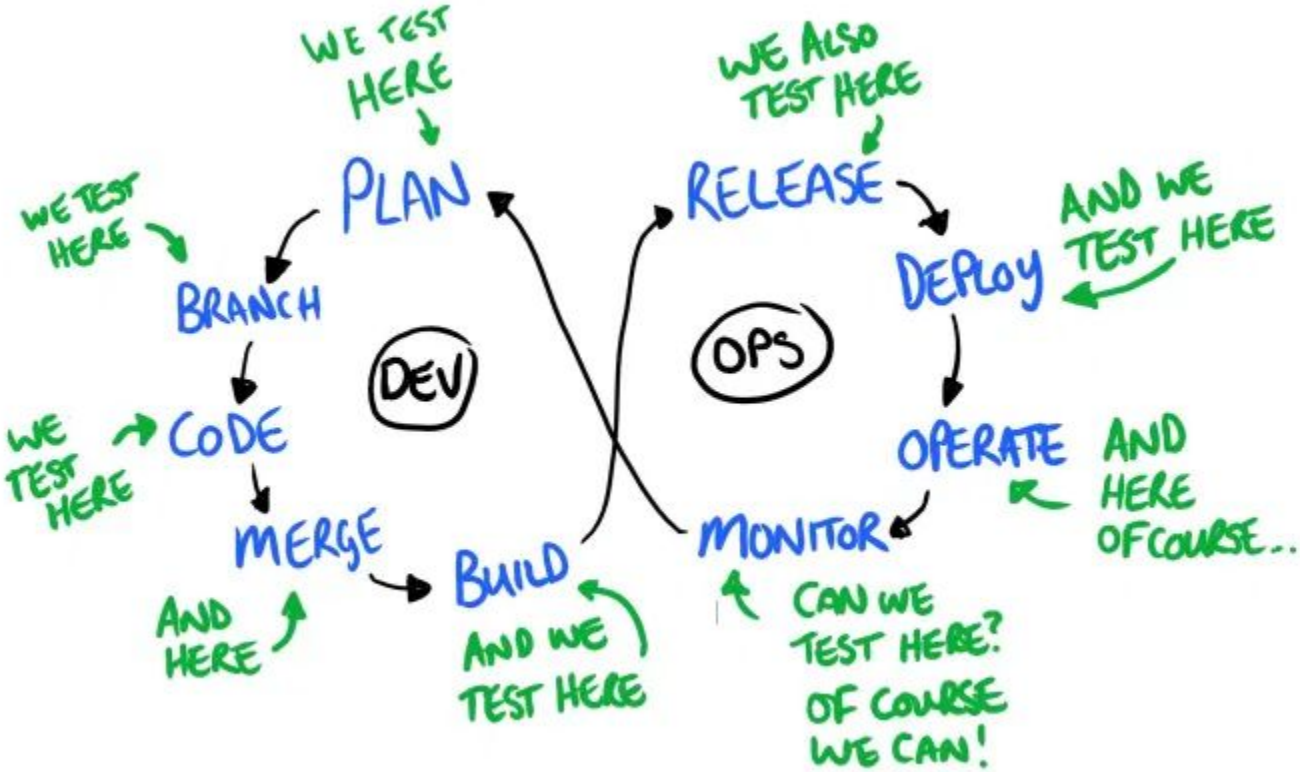
Definitie “deployment pipeline”:
(first defined by Jez Humble and David Farley in their book *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*)

It ensures that all code checked in to version control is automatically built and tested in a production-like environment.

Sleutelwoorden:

- Alle Code
- Versiebeheer
- Automatisch gebouwd
- **Automatisch getest**
- Productie-waardige omgeving

Integrated testing



RECAP - Testing in de 3 ways

Test **ALLES** vanaf het moment dat het kan!

Test **automatisch** om Continous Integration te bekomen

Definitie *continuous integration*:

- **A comprehensive and reliable set of automated tests that validate we are in a deployable state.**
- **A culture that "stops the entire production line" when our validation tests fail. (=digital andon cord)**

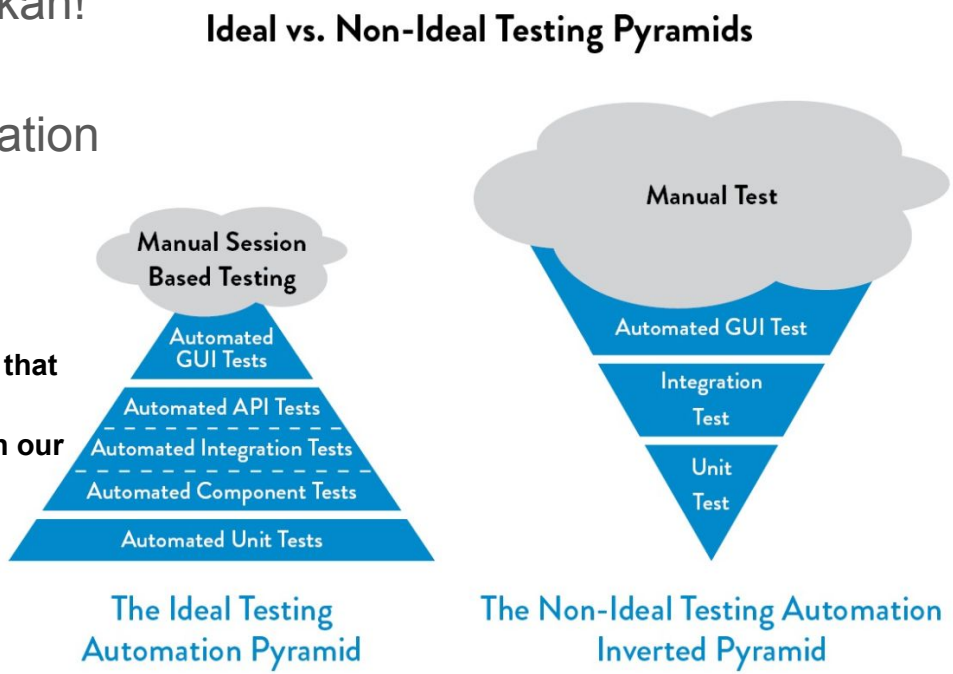


Figure 14: *The ideal and non-ideal automated testing pyramids (Source: Martin Fowler, "TestPyramid.")*

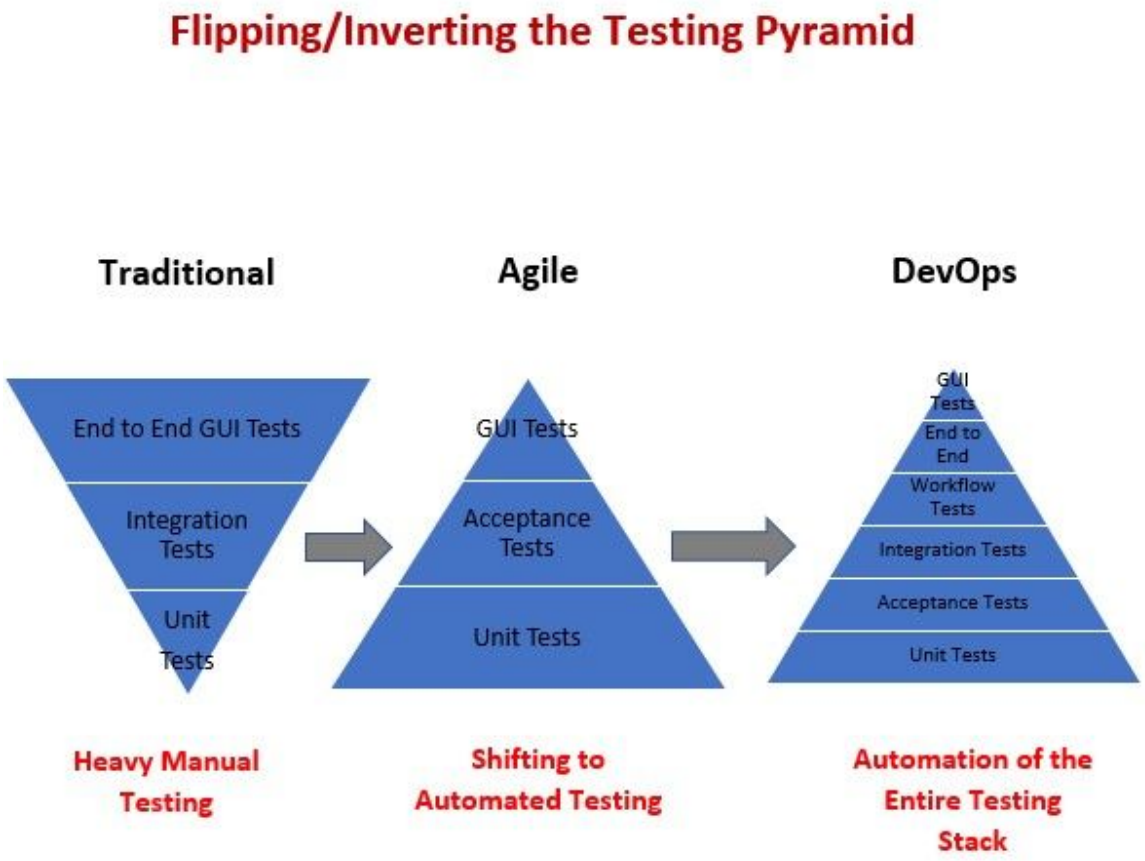
Maar hoe gaan we daarmee om?

Overal testen is goed in devops-> Zo snel mogelijk feedback!

Maar, testen kost tijd, dus hoe lossen we dat op?

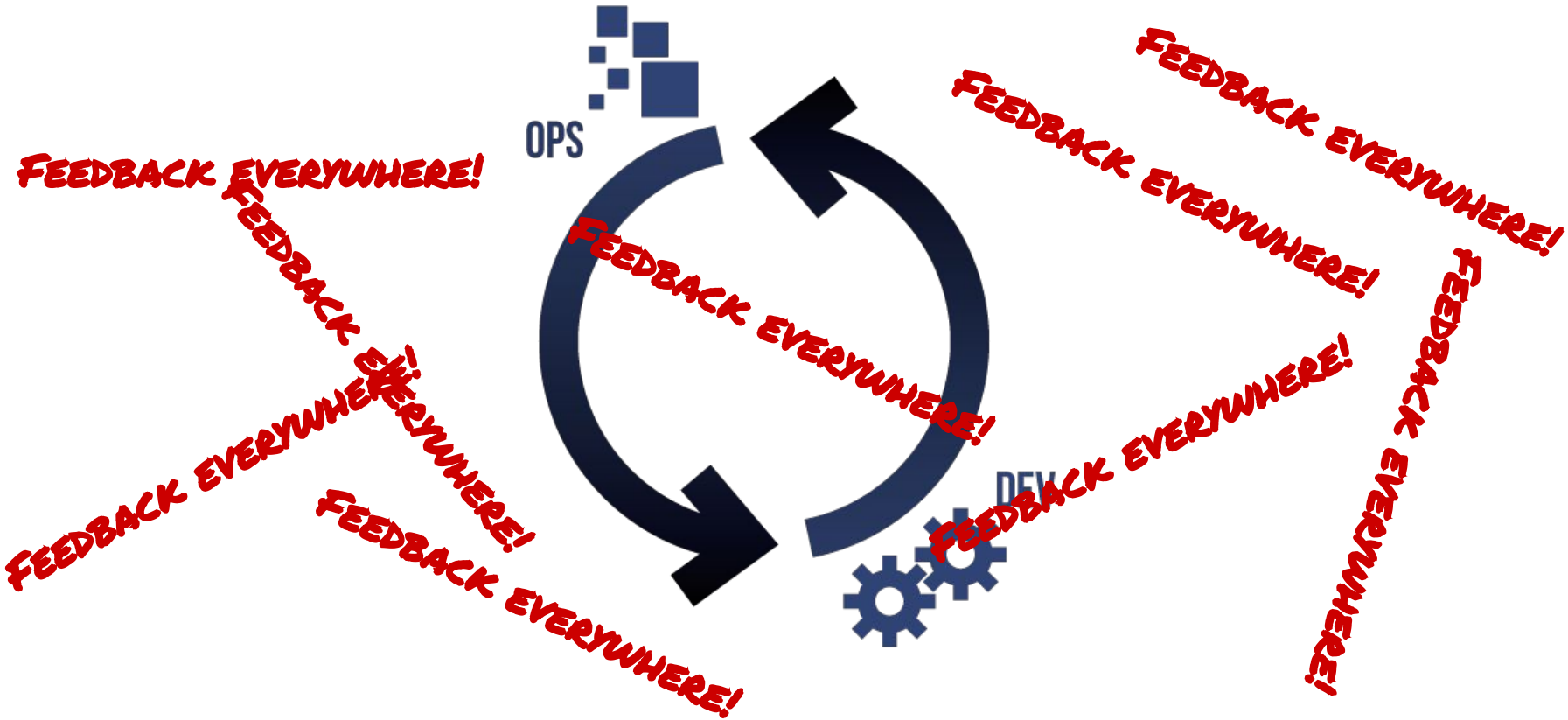
We LEREN en optimaliseren waar nodig.

Begin gerust manueel, maar evolueer!

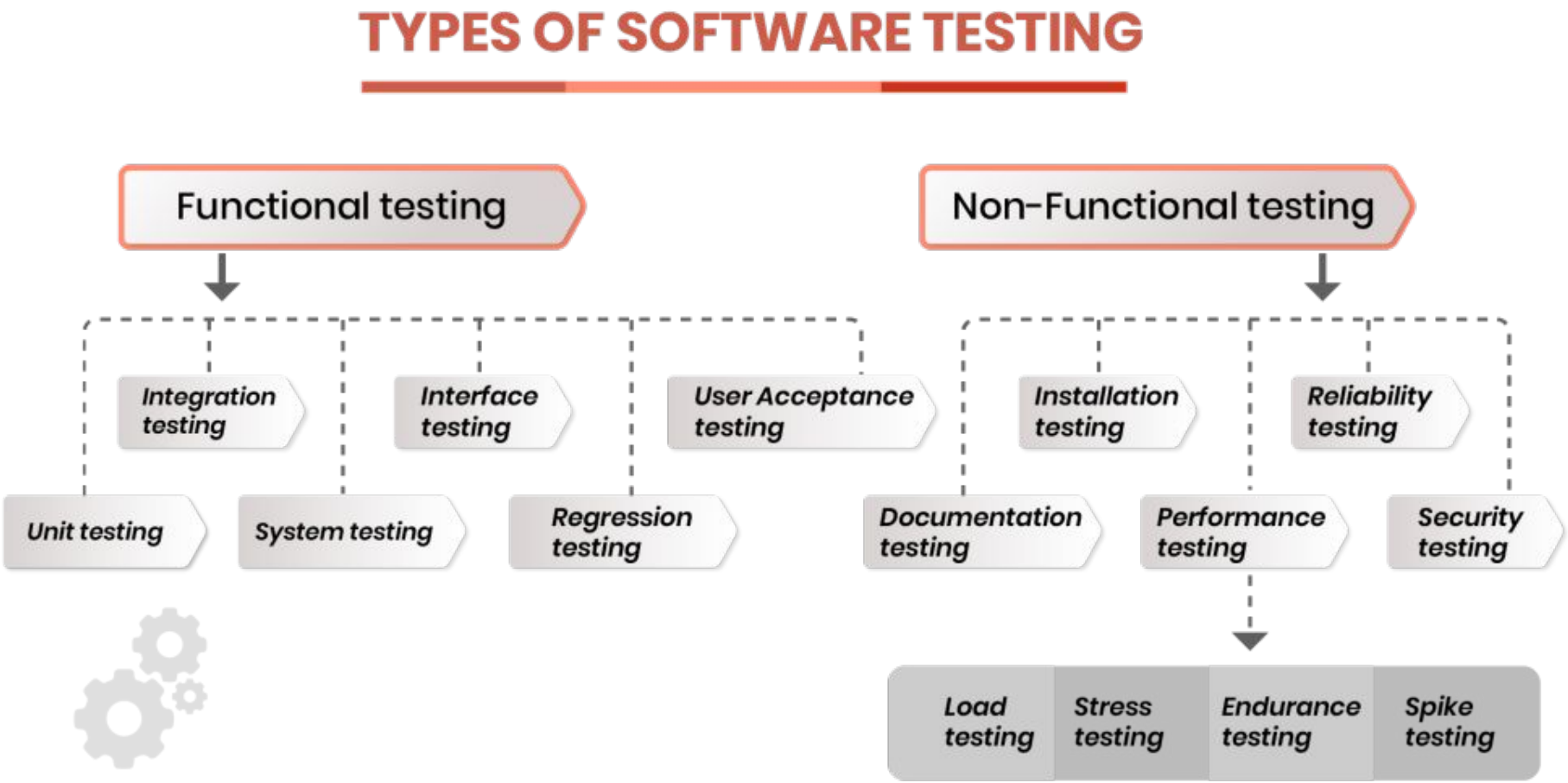


THE SECOND WAY

AMPLIFY FEEDBACK LOOPS



Soorten Testen



Soorten testen - Voorbeeld

Voor een functionele **smartphone** zijn de belangrijkste onderdelen die nodig zijn "batterij" en "simkaart".

- **Unit testing** - de batterij wordt gecontroleerd op levensduur, capaciteit en andere parameters. Simkaart wordt gecontroleerd op activering.

- **Integration Testing** - batterij en simkaart zijn geïntegreerd, dwz samengesteld om de mobiele telefoon te starten.

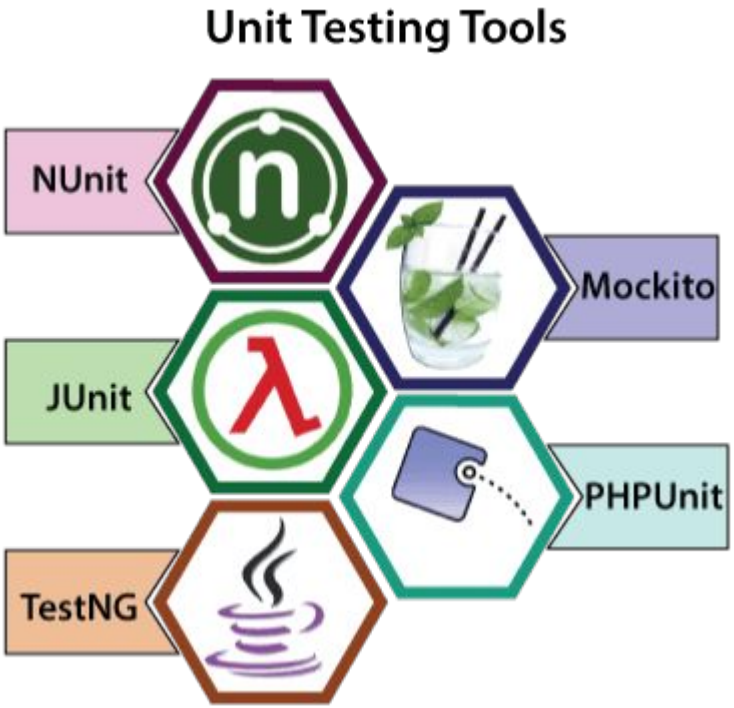
  Opgeslagen nummers op de simkaart kunnen uitgelezen worden door de telefoon.

  De geactiveerde simkaart maakt het mogelijk om met de telefoon een (nood)nummer te bellen.

- **Functioneel testen** - de functionaliteit van de mobiele telefoon wordt gecontroleerd op basis van zijn functies en ook op batterijgebruik en simkaartfaciliteiten.

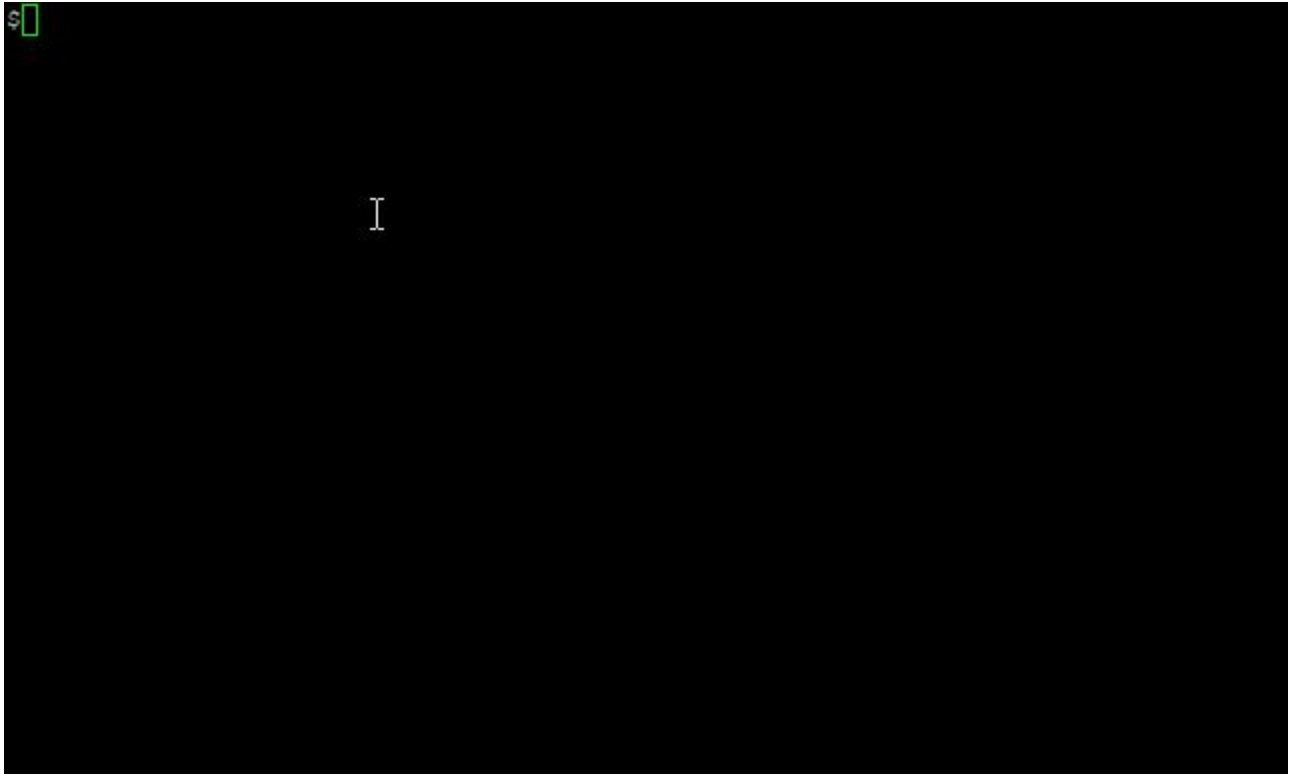
Unit testen

- Testen van software componenten los van Groter geheel
- Kwaliteitscontrole op code-niveau
- Bewijs dat code doet wat het zou moeten Doen



https://www.javatpoint.com/unit-testing-tools

Unit testen

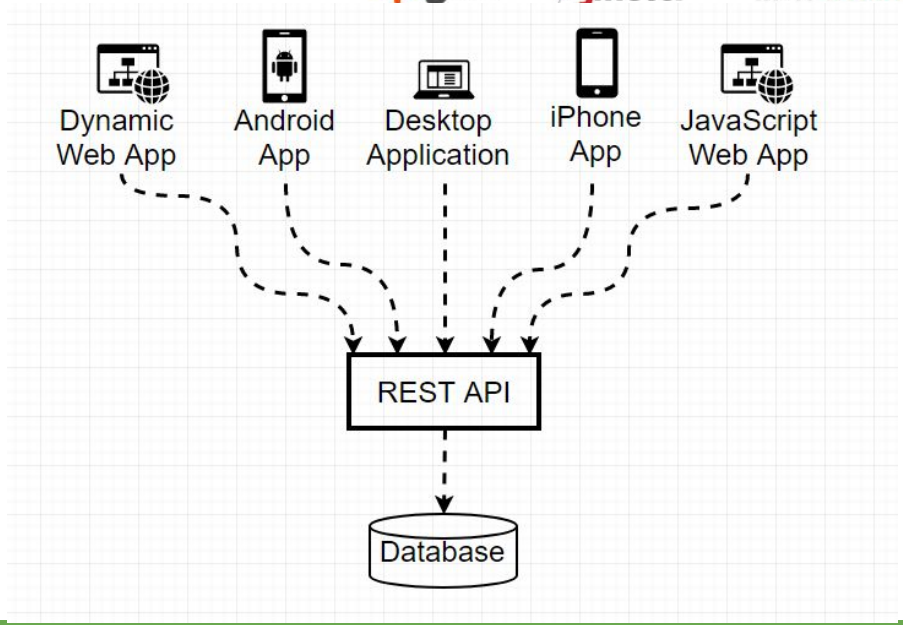


Integratie testen

API Testing Tools

Integraties met andere software

-> Meestal via API's



Integratie testen

Opbouw API (In java)

- Programma code
- Jetty webserver

```web-app
<web-app
  xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1">

  <servlet>
    <servlet-name>PersonServlet</servlet-name>
    <servlet-class>PersonServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>PersonServlet</servlet-name>
    <url-pattern>/people/*</url-pattern>
  </servlet-mapping>

</web-app>
```

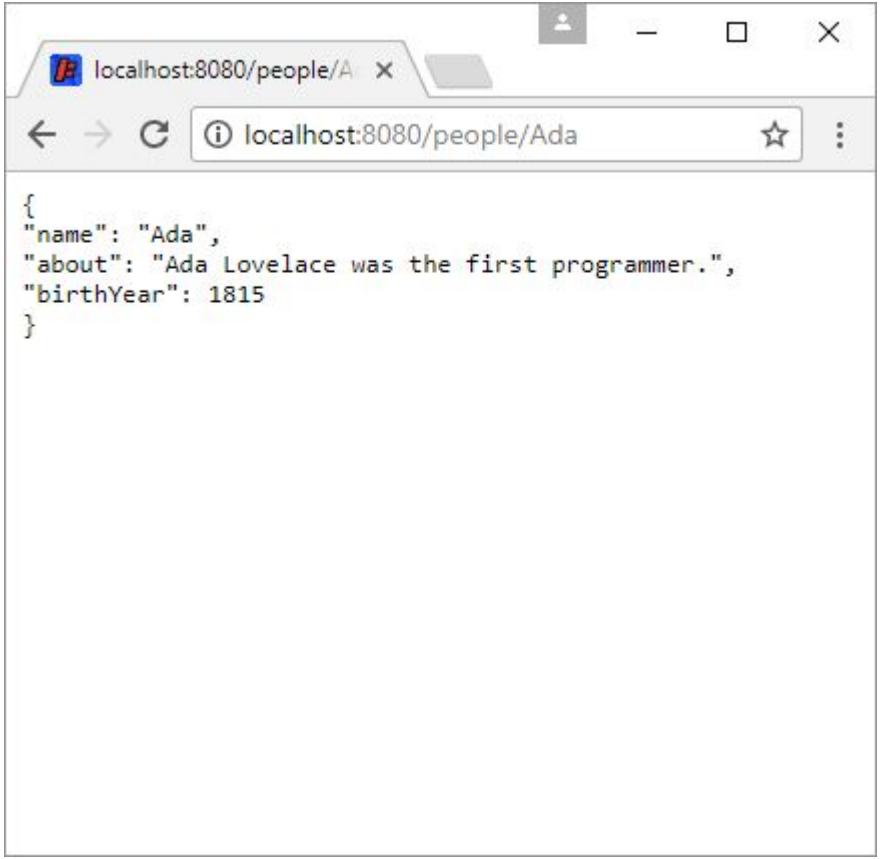
```java
import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.json.JSONObject;

public class PersonServlet extends HttpServlet {

  @Override
  public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {

    String requestUrl = request.getRequestURI();
    String name = requestUrl.substring("/people/".length());

    Person person = DataStore.getInstance().getPerson(name);

    if(person != null){
      String json = "{\n";
      json += "\"name\": " + JSONObject.quote(person.getName()) + ",\n";
      json += "\"about\": " + JSONObject.quote(person.getAbout()) + ",\n";
      json += "\"birthYear\": " + person.getBirthYear() + "\n";
      json += "}";
      response.getOutputStream().println(json);
    }
    else{
      //That person wasn't found, so return an empty JSON object. We could also return an error.
      response.getOutputStream().println("{}");
    }
  }



  @Override
  public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {

    String name = request.getParameter("name");
    String about = request.getParameter("about");
    int birthYear = Integer.parseInt(request.getParameter("birthYear"));

    DataStore.getInstance().putPerson(new Person(name, about, birthYear, password));
  }
}
```

Integratie testen

Output in webbrowser:

Hoe testen we dit?

- Manueel met Browser of wget (CLI)?
- Automatisch!



Integratie testen

- Postman test collection
  - Bundeling van requests op url’s met headers / content / body / ….
  - Automatische inventarisatie van de response
  - Mogelijk om de volledige suite in één keer te runnen
  - Eenvoudige import / export features

https://learning.postman.com/docs/getting-started/creating-the-first-collection/

Integratie testen

Postman

File Edit View Help

New Import Runner My Workspace Invite Upgrade

Filter

History Collections APIs

+ New Collection Trash

test
1 request

Launchpad | GET get | GET https://www.a... | GET http://localhost...

No Environment

Untitled Request BUILD

GET http://localhost:8080/people/ada Send Save

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION	••• Bulk Edit
Key	Value	Description	

Response


```
{
"name": "Ada",
"about": "Ada Lovelace was the first programmer.",
"birthYear": 1815
}
```


Find and Replace Console Bootcamp Build Browse

Integratie testen

Postman

File Edit View Help

New Import Runner My Workspace Invite Upgrade

Filter

History Collections APIs

+ New Collection Trash

test
2 requests

GET get

POST Devops-POST-TestCase1

Launchpad | GET get | GET https://w... | GET http://loc... | POST Devops...

No Environment

Devops-POST-TestCase1

Examples 0 BUILD

POST https://postman-echo.com/post?PXL=Digital&Course=DevOps Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
PXL	Digital	Departement
Course	DevOps	Vak
Key	Value	Description

Bulk Edit

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 398 ms Size: 993 B Save Response

Pretty Raw Preview Visualize JSON


```
1  {
2      "args": {
3          "PXL": "Digital",
4          "Course": "DevOps"
5      },
6      "data": "Deze data zit in de body van de boodschap die ik upload",
7      "files": {},
8      "form": {},
9      "headers": {
10         "x-forwarded-proto": "https",
11         "x-forwarded-port": "443",
12         "host": "postman-echo.com",
13         "x-amzn-trace-id": "Root=1-5fb433a2-4073656864c79bde37560044",
14         "content-length": "55",
```


Find and Replace Console Bootcamp Build Browse

Integratie testen

=> Suites zijn uitbreidbaar met test scripts

- Diepere controle op de gemaakte requests en responses
  - Welke statuscode krijg ik terug
  - Welke waardes krijg ik terug, zit waarde x daarin
  - Gebruik van variabelen doorheen de test suite
- Pass / fail net zoals bij andere soorten testen

https://learning.postman.com/docs/writing-scripts/test-scripts/

https://learning.postman.com/docs/writing-scripts/script-references/test-examples/

Integratie testen

Postman ECHO No Environment, just now

Run Summary | Run Again | New

All Tests | Passed (4) | Failed (1)

1

Iteration 1

GET GET Request postman-echo.com/get?record={{name_detail}} 200 OK 227 ms 860 B

- Pass Status code is 200
- Pass Status code is 200

POST POST Request postman-echo.com/post 200 OK 226 ms 880 B

- Pass Status code is 200
- Pass Status code is 200
- Fail Response time is less than 200ms | AssertionError: expected 226 to be below 200

Desktop Agent | Runner | Trash

Integratie testen - Postman test scripts

- Testen op response code

Intro to writing tests - with examples / Basic test syntax / **pm.test**

Save

GET https://postman-echo.com/get?foo1=bar1&foo2=bar2 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

```
1  pm.test("Status test", function () {
2      pm.response.to.have.status(200);
3  });
```

Test scripts are written in JavaScript, and are run after the response is received.

Learn more about tests scripts

SNIPPETS

Get an environment variable

Get a global variable

Get a variable

Set an environment variable

Set a global variable

Clear an environment variable

Integratie testen - Postman test scripts

- Testen op response code + body

POST | localhost:3000/api/artist

Params | Authorization | Headers (8) | Body | Pre-request Script | Tests | Settings

```
1  pm.test("response should be okay to process", function () {
2      pm.response.to.not.be.error;
3      pm.response.to.be.json;
4      pm.response.to.not.have.jsonBody("error");
5  });
```

Integratie testen - Postman test scripts

- Testen op inhoud response

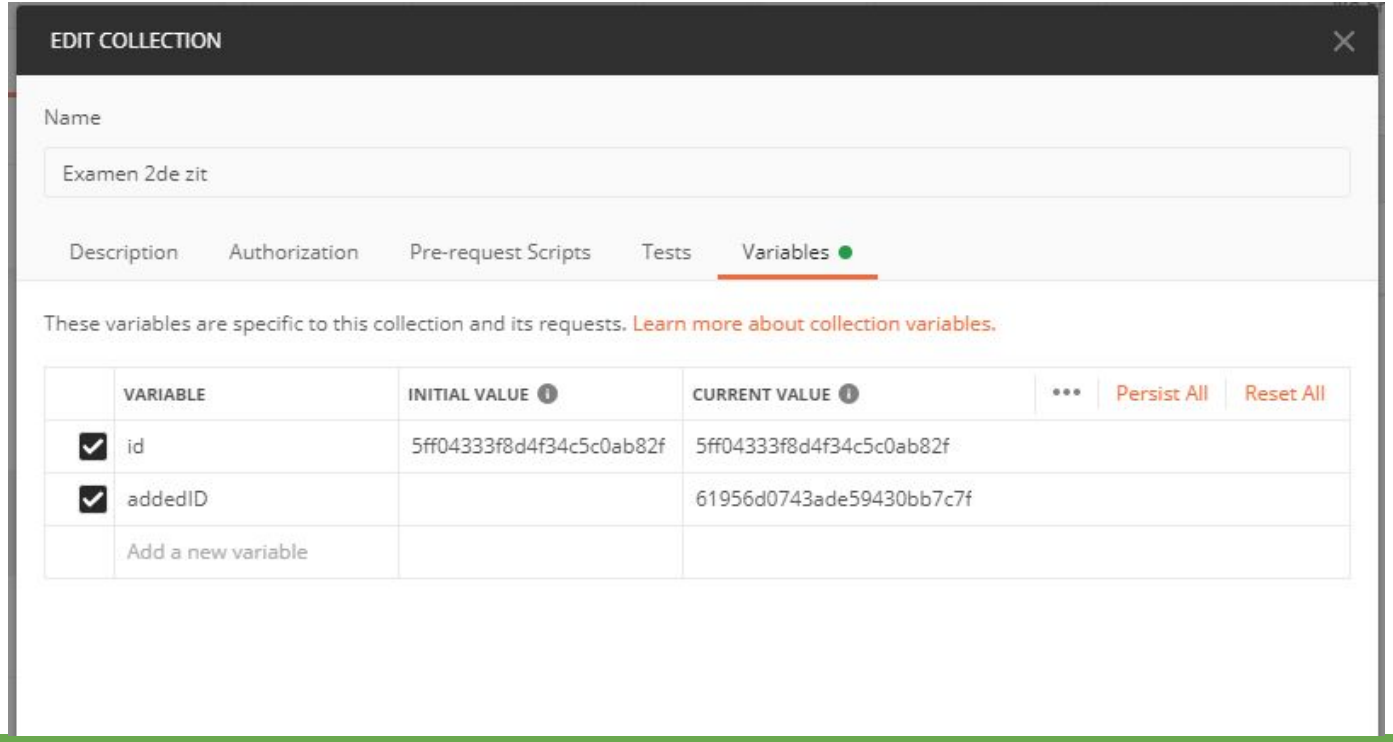
PUT localhost:3000/api/artist

Params Authorization Headers (8) Body Pre-request Script Tests Settings

```
1
2 pm.test("Body contains genre pelt",() => {
3   pm.expect(pm.response.text()).to.include("pelt");
4 });
```

```
pm.test("Person is Jane", () => {
  const responseJson = pm.response.json();
  pm.expect(responseJson.name).to.eql("Jane");
  pm.expect(responseJson.age).to.eql(23);
});
```

- Gebruik van variabelen



Integratie testen - Postman test scripts

- Gebruik van variabelen
  - Waarde opslaan:

```
const responseJson = pm.response.json();
pm.collectionVariables.set("addedID", responseJson._id);
```

  - Waarde opvragen

```
pm.test("Body contains _id with newly added",() => {
  pm.expect(pm.response.text()).to.include(pm.collectionVariables.get("addedID"));
});
```

Integratie testen - Postman CLI

- Postman = grafische applicatie
  - Niet integreerbaar in CI pipelines
- Newman = CLI applicatie
  - Integreerbaar in CI pipelines als npm package
  - Kan geexporteerde postman test suites / scripts uitvoeren
  -

https://learning.postman.com/docs/running-collections/using-newman-cli/command-line-integration-with-newman/

Functionele (E2E) testen

- Testen van UI
  - Niet meer op code niveau
- Applicatie moet *functioneel* zijn
- Applicatie wordt getest als (sub)geheel, niet meer als losse code-componenten

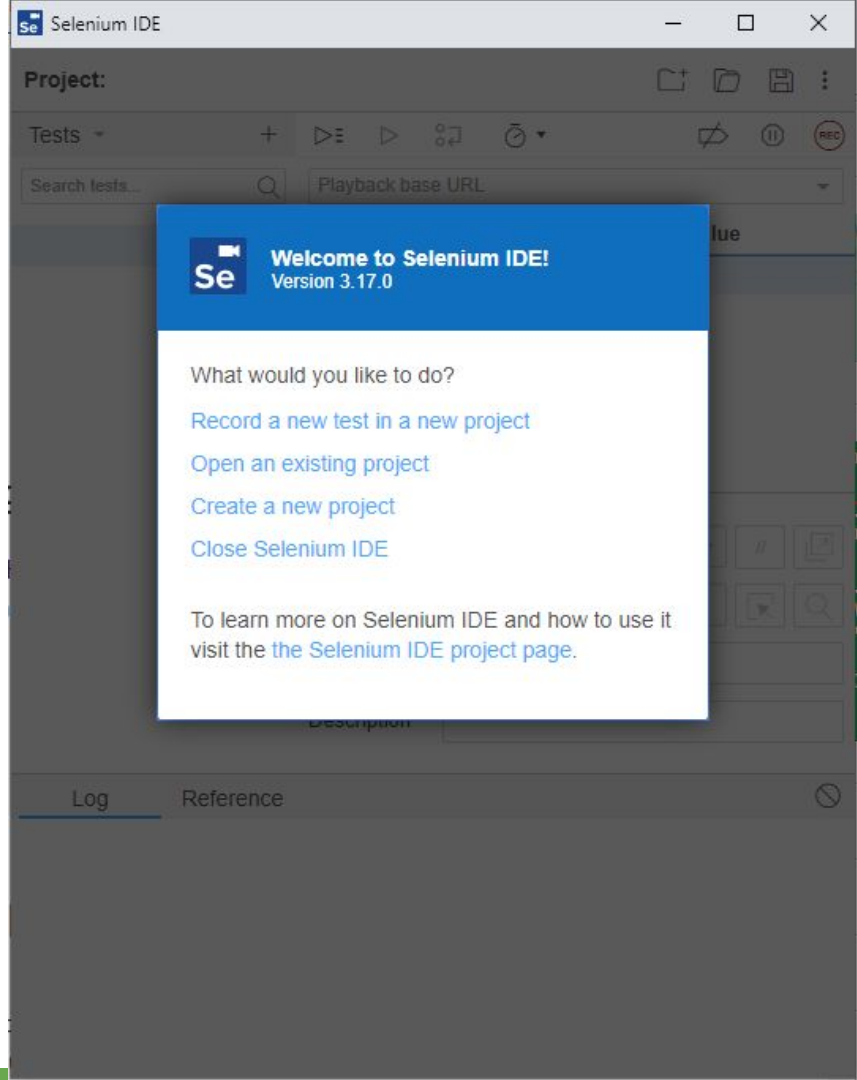
Functionele (E2E) testen

- Selenium IDE
  - Chrome en/of firefox plugin
- Lightweight add-on om E2E testen te schrijven
- .side files

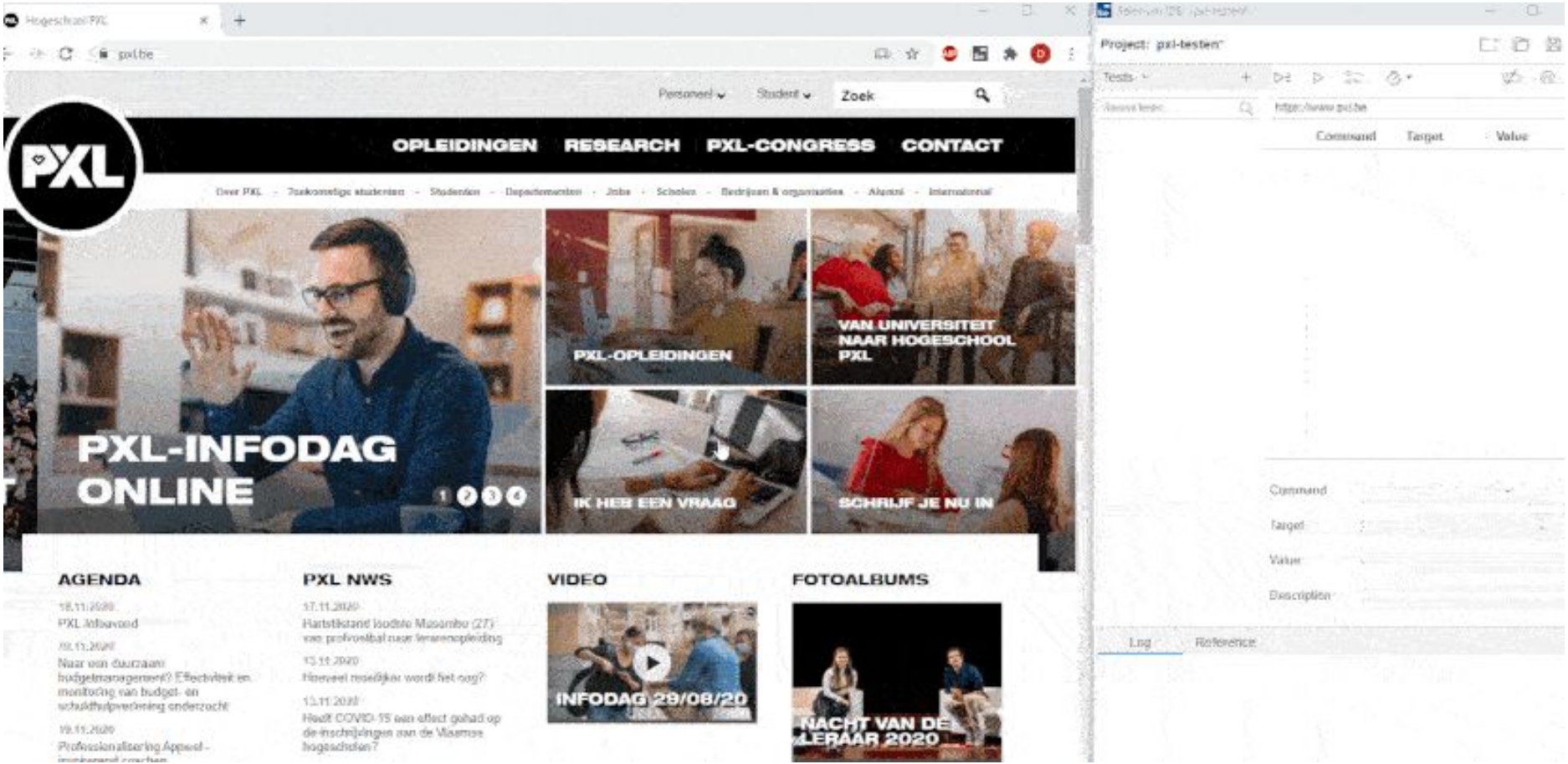
pxl-testen - Kladblok

Bestand Bewerken Opmaak Beeld Help


```
      "target": "/",
      "targets": [],
      "value": ""
    }, {
      "id": "8c9e4ca0-9532-4689-ac3b-9fd029959c6a",
      "comment": "",
      "command": "setWindowSize",
      "target": "1187x867",
      "targets": [],
      "value": ""
    }, {
      "id": "65e4c567-9f85-4828-a19c-cd1fb3d1231a",
      "comment": "",
      "command": "click",
      "target": "css=.nav_list > li:nth-child(1) > a",
      "targets": [
        ["css=.nav_list > li:nth-child(1) > a", "css:finder"],
        ["xpath=//a[contains(text(),'Opleidingen')]", "xpath:link"],
        ["xpath=//div[@id='nav1_container']/div/div[2]/ul/li/a", "xpath:idRelative"],
        ["xpath=//a[contains(@href, '/Opleidingen.html')]", "xpath:href"],
        ["xpath=//div[2]/div/div/div[2]/ul/li/a", "xpath:position"]
      ],
```

Functionele (E2E) testen



Selenium IDE

« Documenten › workspace › selenium-test-demo

Naam

pxl-testen.side

pxl-testen - Kladblok

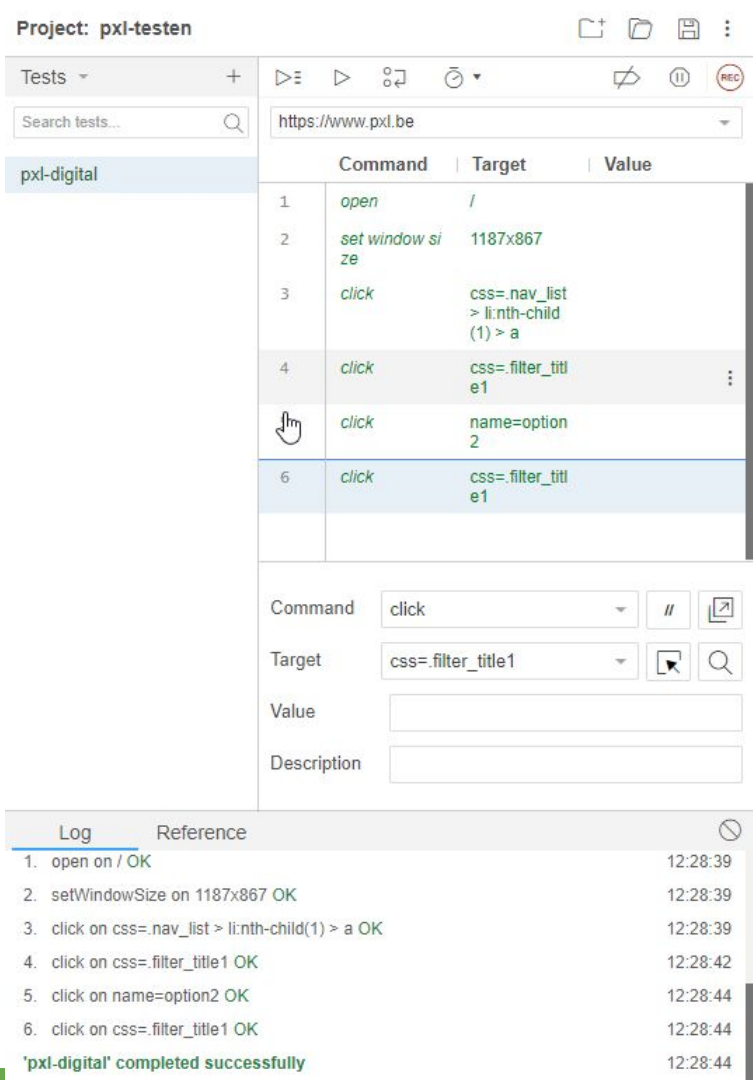
Bestand Bewerken Opmaak Beeld Help


```
      "target": "/",
      "targets": [],
      "value": ""
    }, {
      "id": "8c9e4ca0-9532-4689-ac3b-9fd029959c6a",
      "comment": "",
      "command": "setWindowSize",
      "target": "1187x867",
      "targets": [],
      "value": ""
    }, {
      "id": "65e4c567-9f85-4828-a19c-cd1fb3d1231a",
      "comment": "",
      "command": "click",
      "target": "css=.nav_list > li:nth-child(1) > a",
      "targets": [
        ["css=.nav_list > li:nth-child(1) > a", "css:finder"],
        ["xpath=//a[contains(text(),'Opleidingen')]", "xpath:link"],
        ["xpath=//div[@id='nav1_container']/div/div[2]/ul/li/a", "xpath:idRelative"],
        ["xpath=//a[contains(@href, '/Opleidingen.html')]", "xpath:href"],
        ["xpath=//div[2]/div/div/div[2]/ul/li/a", "xpath:position"]
      ],
      "value": ""
    }, {
      "id": "f233f1fb-fd19-4599-afcb-a56a3ab9d829",
      "comment": "",
      "command": "click",
      "target": "css=.filter_title1",
      "targets": [
        ["css=.filter_title1", "css:finder"],
        ["xpath=//div[@id='content']/div/div/span", "xpath:idRelative"],
```

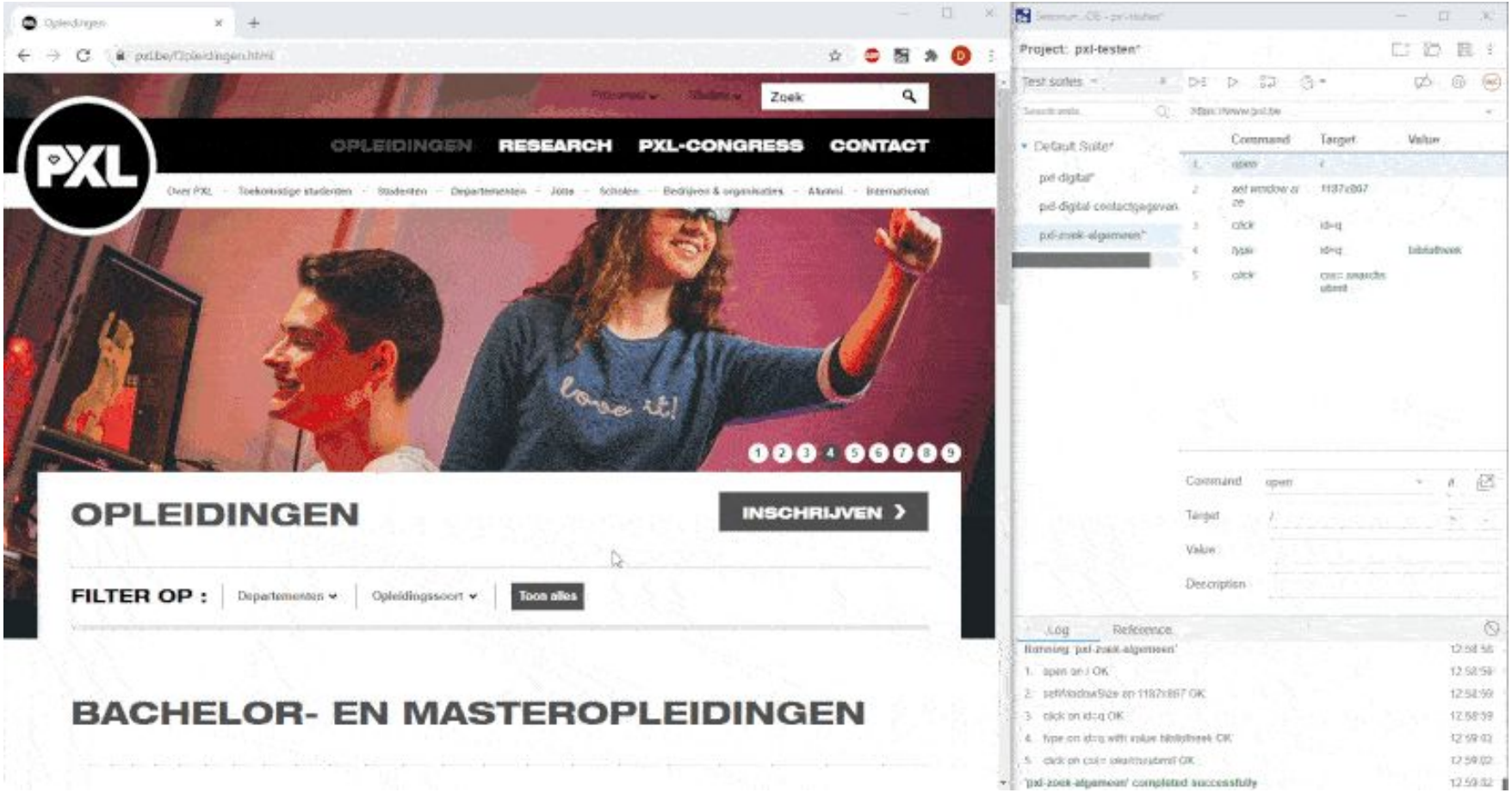

Ln 1, Col 1

Selenium IDE

- Custom commands
  - Controle of elementen (niet) aanwezig zijn
  - Waar die aanwezig moeten zijn
  - …
- Testen falen of slagen
- Bundeling van testen = test suite



Selenium IDE



Continious quality in een pipeline

- Testen runnen vanuit een pipeline
  - CLI omgevingen
  - Integratie met externe systemen (APIs?)
- Functionele testen
  - Selenium CLI

E2E testen in een pipeline

Command-line Runner

You can now run all of your Selenium IDE tests on any browser, in parallel, and on a Grid without needing to write any code.

There's just the small matter of installing the Selenium IDE command line runner, getting the necessary browser drivers (if running your tests locally), and launching the runner from a command prompt with the options you want.

```
3. selenium-side-runner /Users/corevo/Downloads (Google Chrome He)
corevo@sovereign:~/Downloads (master|✓)
» selenium-side-runner project.side
info:    Running project.side
 PASS  ./control flow.test.js (10.131s)
 PASS  ./waits.test.js (53.364s)

 RUNS  ./smoke.test.js
 RUNS  ./all tests.test.js

Test Suites: 2 passed, 2 of 4 total
Tests:       11 passed, 11 total
Snapshots:   0 total
Time:        70s
```

https://www.selenium.dev/selenium-ide/docs/en/introduction/command-line-runner

E2E testen in een pipeline - benodigdheden

Prerequisites

The following dependencies are needed for the command line runner to work:

- `node` (the Node.js programming language) version `8` or `10`
- `npm` (the NodeJS package manager) which typically gets installed with `node`
- `selenium-side-runner` (the Selenium IDE command line runner)
- and the browser driver we want to use (more on that in the next section)

```
> brew install node
> npm install -g selenium-side-runner
```

https://www.selenium.dev/selenium-ide/docs/en/introduction/command-line-runner

E2E testen in een pipeline - benodigdheden

- Browser nodig - Op een server? - Headless!
- Chrome webdriver

  https://chromedriver.chromium.org/

Continious quality in Jenkins

- Manuele tool installatie!

Install Chromedriver before use!

```
wget https://chromedriver.storage.googleapis.com/86.0.4240.22/chromedriver_linux64.zip
unzip chromedriver_linux64.zip
sudo mv ./chromedriver /usr/local/bin/chromedriver
```

Install Chrome before use!

```
wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
sudo dpkg -i google-chrome-stable_current_amd64.deb
```

https://chromedriver.chromium.org/

Continious quality in Jenkins

- Global tool configuration!

NodeJS

NodeJS installations

Add NodeJS

NodeJS

Name

node-selenium

Install automatically

Install from nodejs.org

Version

NodeJS 10.19.0

Force 32bit architecture

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

Global npm packages to install

selenium-side-runner

Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax `packageName@version`

Global npm packages refresh hours

72

Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache

Idem voor newman

Continious quality in Jenkins

- Bouwen van de pipeline
  - Binnenhalen .side testfiles
  - Selenium side runner CLI
    - Headless chrome
    - Genereren van rapporten

```
stage ('fetch tests'){
    steps{
        git 'https://github.com/d-ries/test-selenium.git'
    }
}
stage('Run automated tests') {
    steps {
        nodejs('node-selenium') {
            sh "npm -v"
            sh "selenium-side-runner -c \"goog:chromeOptions.args=[disable-infobars, headless] browserName=chrome\" --output-directory=seleniumReports --output-format=junit *.side"
            sh "ls -lah ."
        }
    }
}
```

Continious quality in Jenkins

```
selenium-side-runner -c \"goog:chromeOptions.args=[disable-infobars, headless] browserName=chrome\" --output-directory=seleniumReports --output-format=junit *.side"
```

- Selenium-side-runner CLI omgeving
- -c **configuration**: geeft aan dat we de testen draaien in een headless chrome omgeving
- Configuratie van de output van de testresultaten
  - In de map seleniumReports
  - In JUNIT formaat
- Uit te voeren test scripts (.side files)

Continious quality in Jenkins

Jenkins

search

1 Dries Swinnen log out

Dashboard › integraded-testing-selenium › #10

- Back to Project
- Status
- Changes
- Console Output
- Edit Build Information
- Delete build '#10'
- Git Build Data
- No Tags
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces
- Previous Build
- Next Build

Build #10 (Oct 28, 2020 1:58:35 PM)

Keep this build forever

add description

Started 19 days ago
Took 7.1 sec

Build Artifacts
calculator-app-selenium.xml 379 B view

Started by user Dries Swinnen

git

Revision: df6e45fcc8cf0705ad8248a59929a81f9f09f4cf

- refs/remotes/origin/master

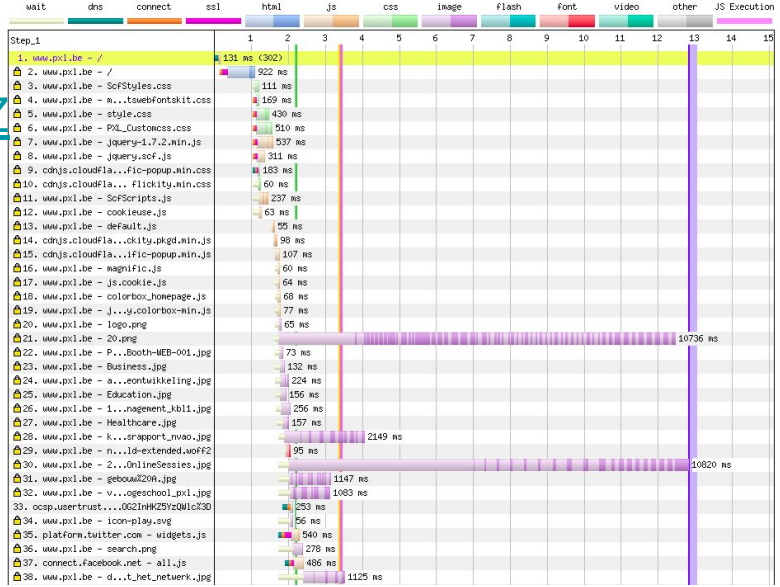
Non-functional testen

Performance testing tools

Front-end testing web apps:

Lighthouse voorbeeld: https://bit.ly/3kFrfNy

Webpagetest voorbeeld: https://bit.ly/2K5XHvZ

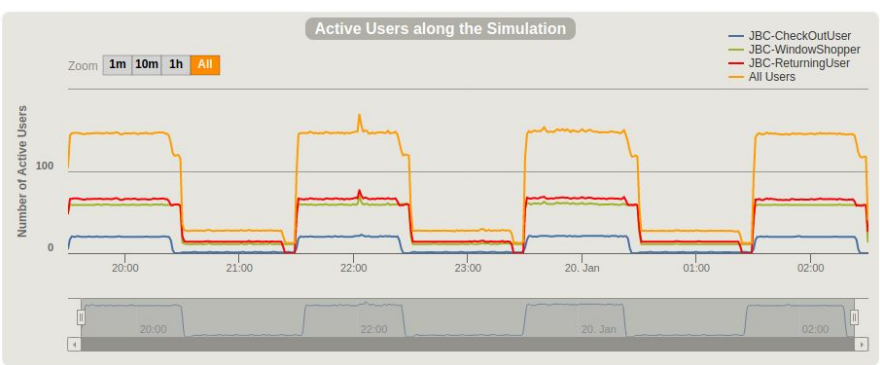


Non-functional testen

Performance testing: End-to-end Enterprise testing

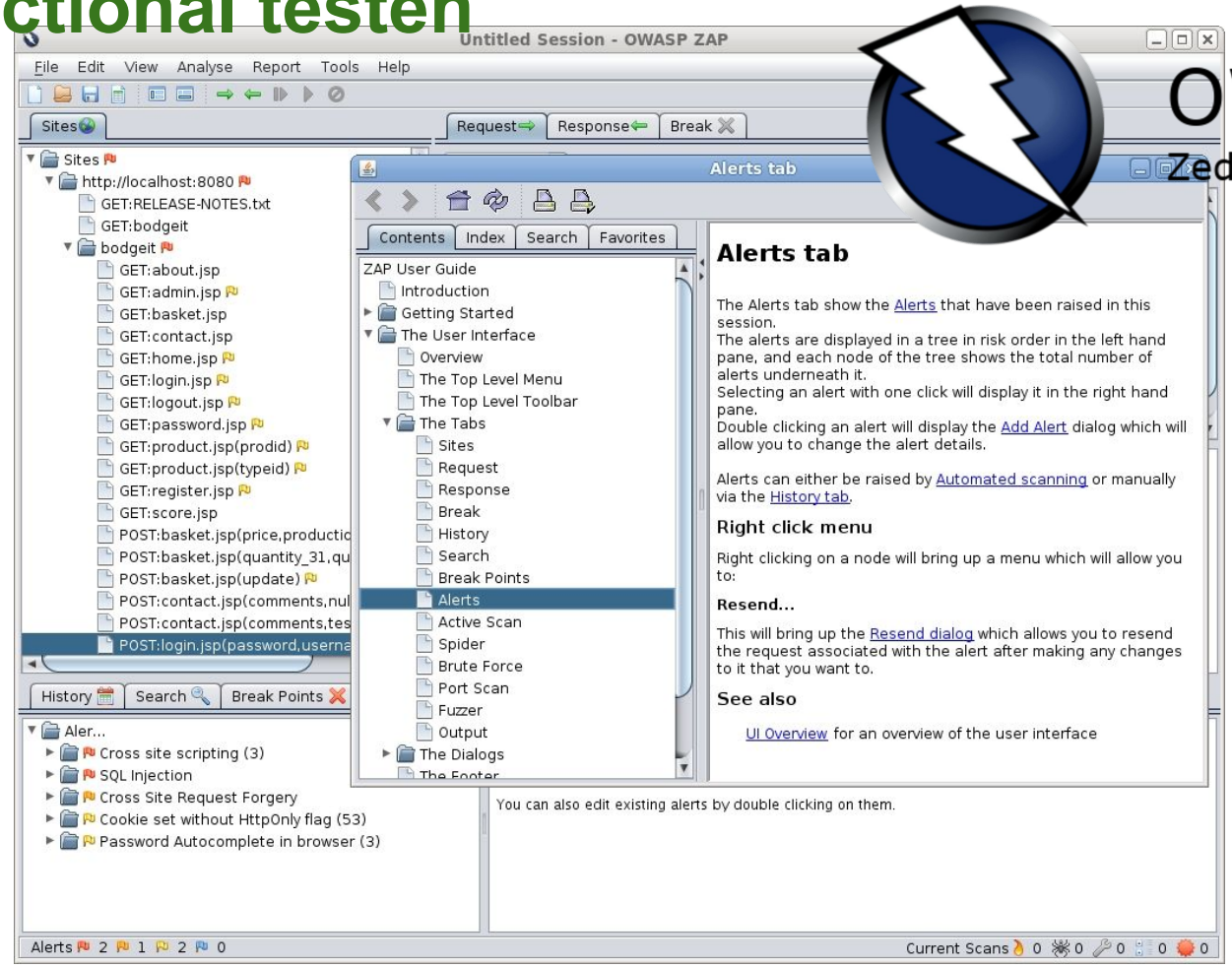
STATISTICS — Expand all groups | Collapse all groups

Requests	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Req/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
Global Information	627257	627209	48	0%	24.864	62	151	177	344	1060	42140	196	314
Homepage	94750	94750	0	0%	3.756	112	151	179	321	831	18219	190	249
Login	36800	36800	0	0%	1.459	460	835	1199	1997	2878	26238	1030	637
ProductDetail	149950	149949	1	0%	5.944	106	160	186	336	812	20913	194	204
Register	2750	2750	0	0%	0.109	340	534	782	1528	2			
Category	94750	94748	2	0%	3.756	99	150	177	356				
Search	94750	94750	0	0%	3.756	98	141	169	299				
StaticPage	92000	91998	2	0%	3.647	94	134	154	278				
Cart	2750	2750	0	0%	0.109	751	1020	1264	2310	3			
Checkout	2750	2707	43	2%	0.109	2612	3417	3624	4780	7			



Non-functional testen

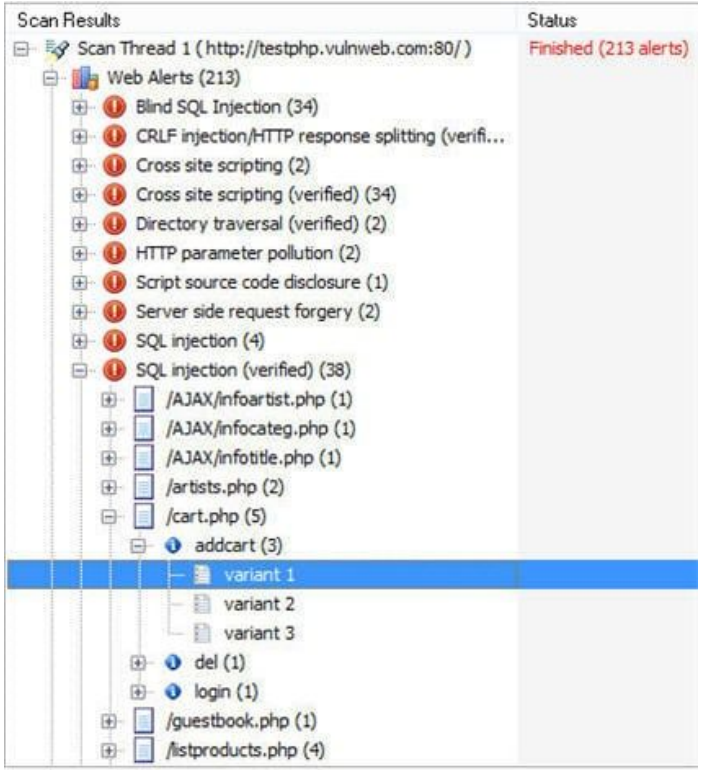
Security testing tools



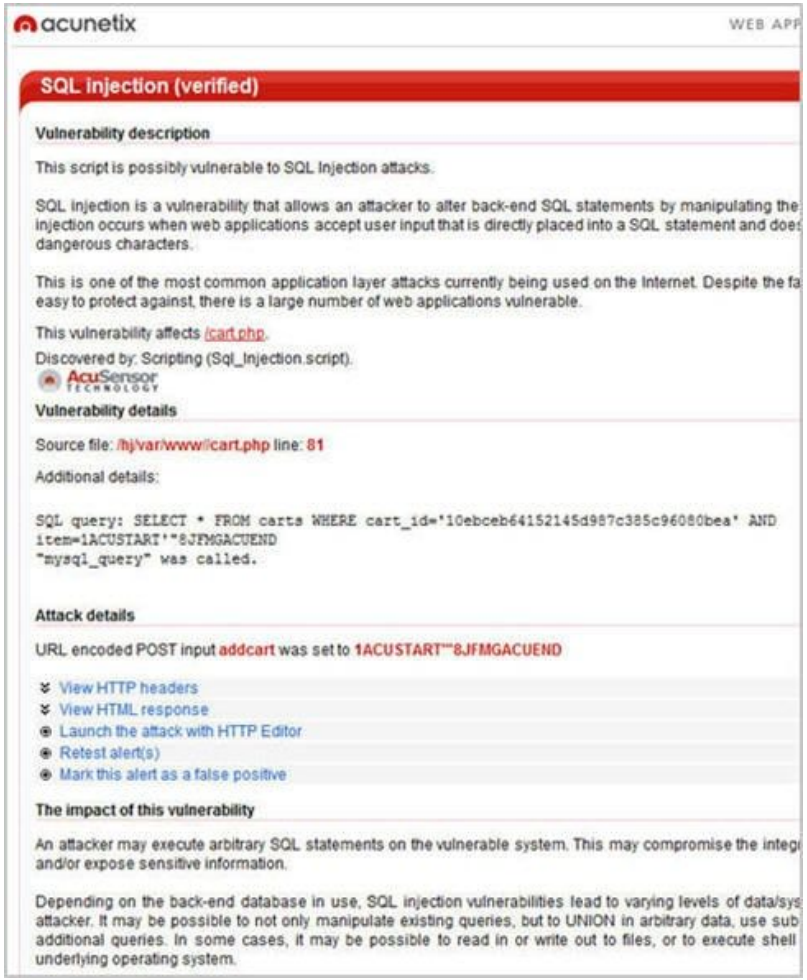
OWASP
Zed Attack Proxy

Non-functional testen

Security testing tools

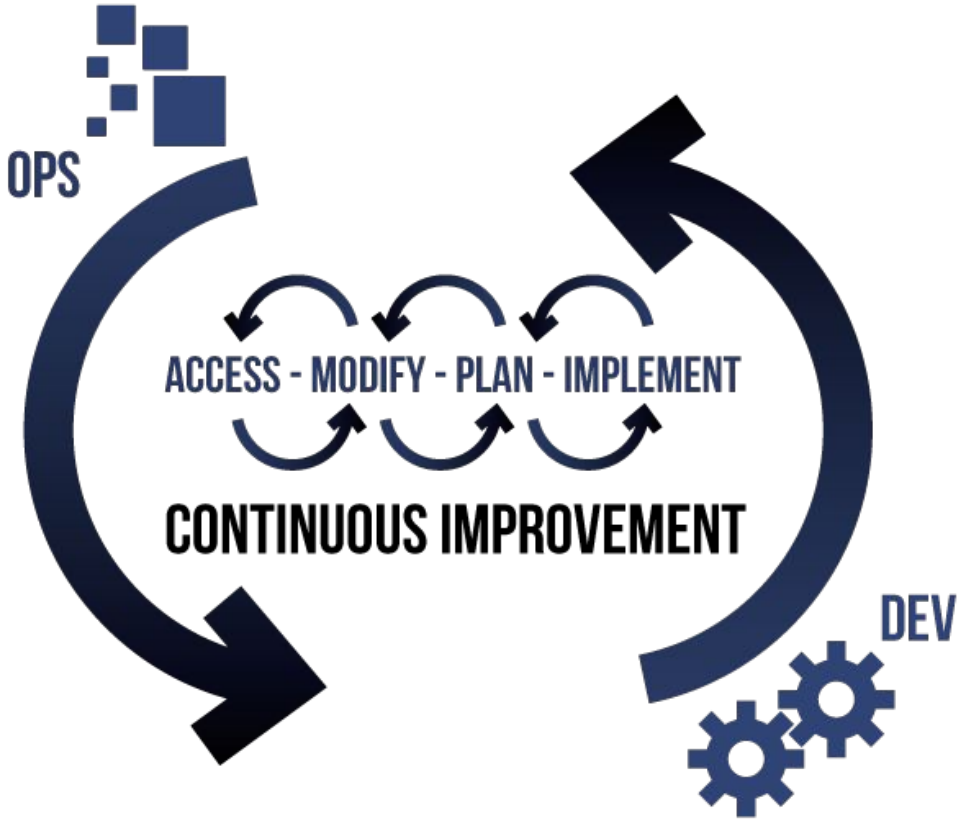


www.SoftwareTestingHelp.com

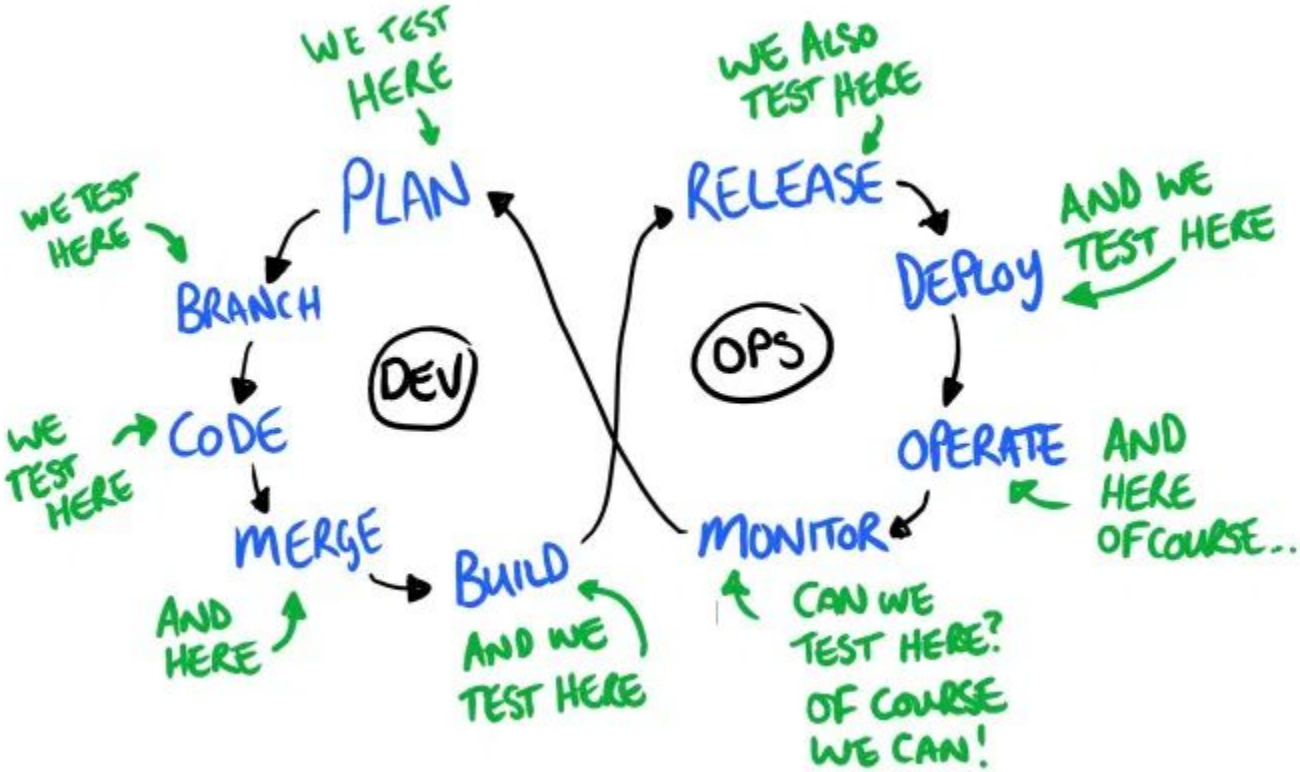


THE THIRD WAY

LEARN & EXPERIMENT CONTINUOUSLY



Integrated testing



PLURALSIGHT

Title: **Creating Automated Browser Tests with Selenium in C#**
Detailed course on writing durable browser tests in c#. [4h04mins]

Title: **Automated Web Testing with Selenium and WebDriver Using Java**
Detailed course on writing durable browser tests in Java. [3h14mins]

Title: **Postman fundamentals**
Introduction to all the features of the postman tool. [2h38mins]

Assignments

HOGESCHOOL PXL