Continuous deployment @wehkamp

a continuous improvement story

Hylke de Jong
Marco van de Bunt

TestNet Najaarsevent
11 oktober, 2016

Agenda

- Introductie
- Van monoliet naar microservices
- Test strategie
- Unit & Integration tests
- User scenario flow tests
- Performance, load & stress tests
- Production trends & health check
- Development pipeline
- Veranderende testrol

Over ons

Hylke de Jong

wehkamp.nl

email: hdjong1@wehkamp.nl

Marco van de Bunt

TRIVENTO

email: marco.van.de.bunt@trivento.nl
twitter: @mvdbunt

Over Wehkamp

Wehkamp

1952 - opgericht door Herman Wehkamp

2006 - transitie naar online

2010 - alle sales door Digitale Kanalen

Facts
- 180.000 producten
- 1.850 verschillende merken
- Grootste geautomatiseerde Ecommerce Warehouse in Europa (Zwolle)
- Same Day Delivery
- And much more...

Grootste online Department Store in NL

Digital Development bij Wehkamp

Ongeveer 80 FTE engineers

Agile Teams voor Frontend Ecosystem

Customer Facing Technology Stack

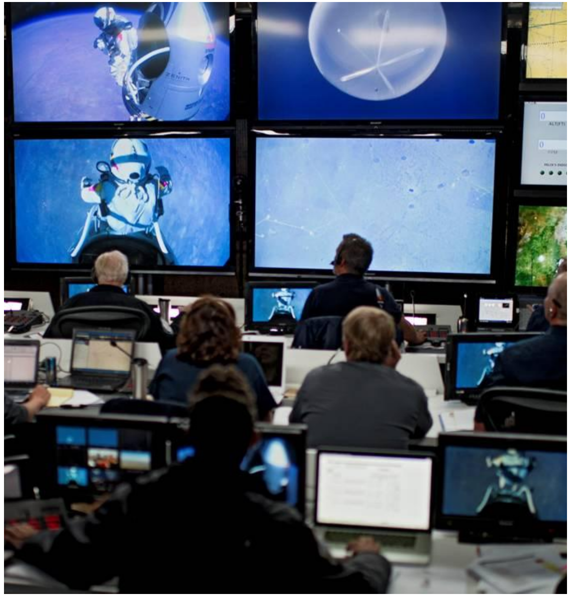
- Innovatie, full stack development
- DevOps/SRE
- Microservices
- Data Engineering
- Open Source, Scala, Java, Akka, Kafka
- And much more...

Technologie en Continuous Improvement

Innovatie is onderdeel van ons DNA

Een ambitieuze missie

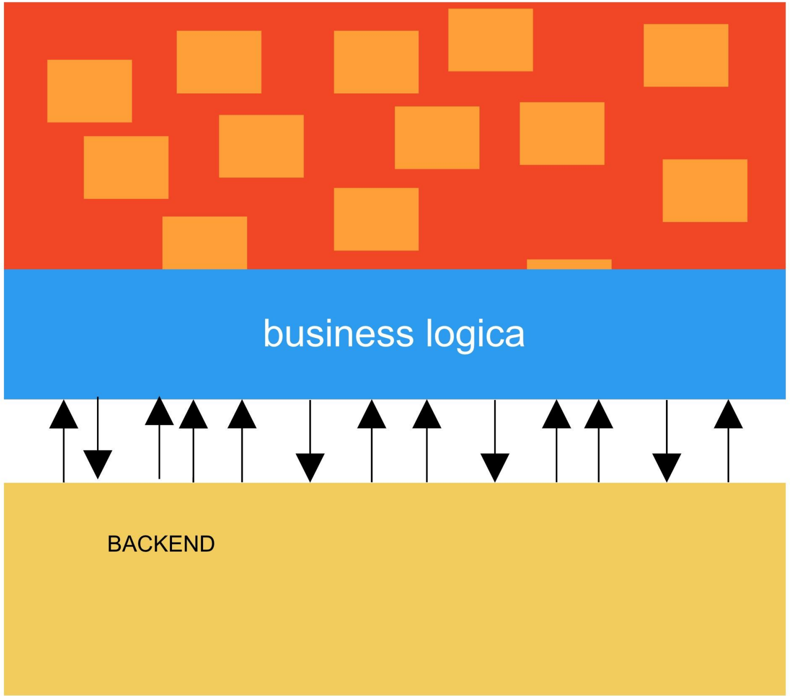
Moed en durf



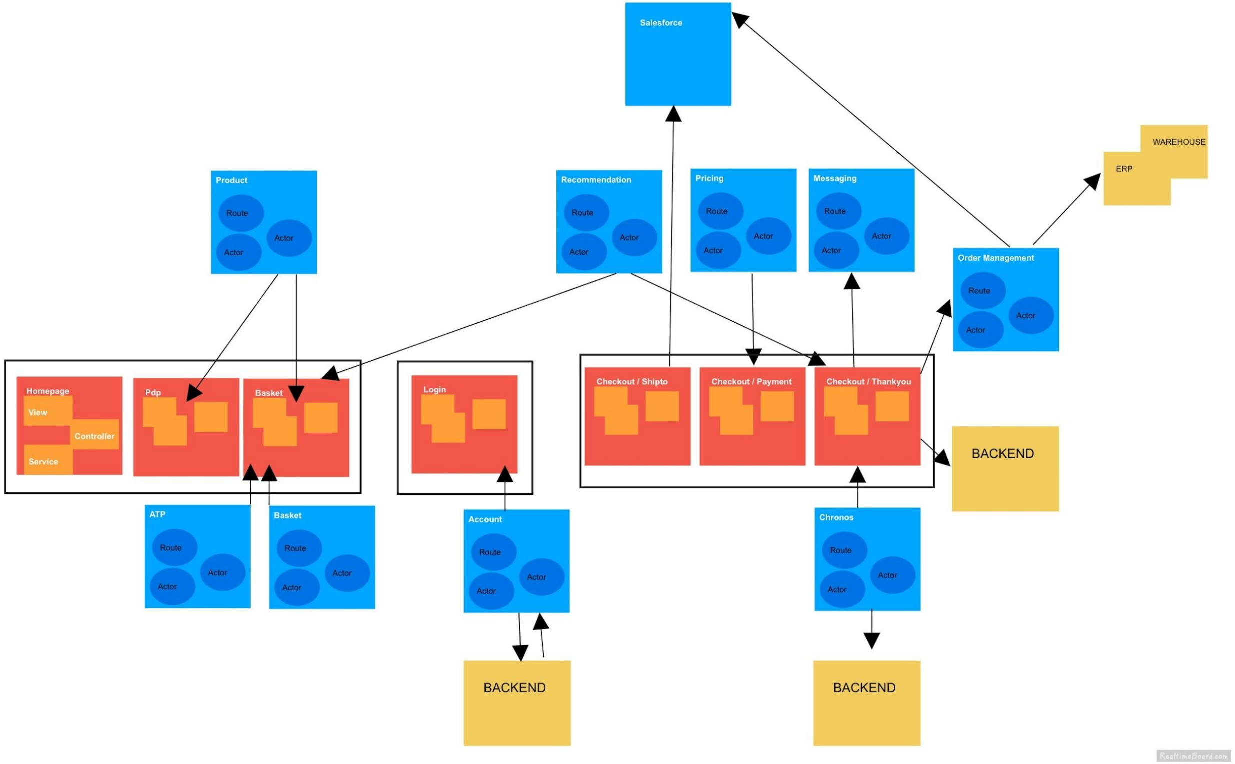
Training, Leiding, Geduld

Jump!
(geen garantie op succes)

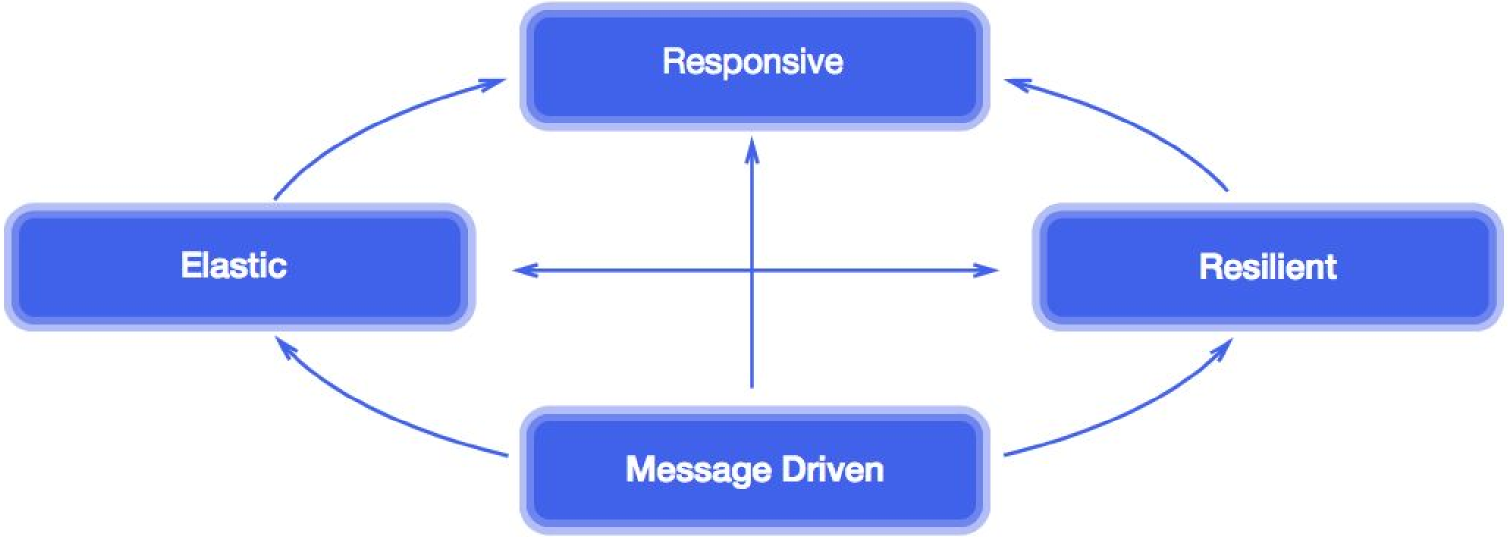
Van monoliet naar microservice



Van monoliet naar microservice

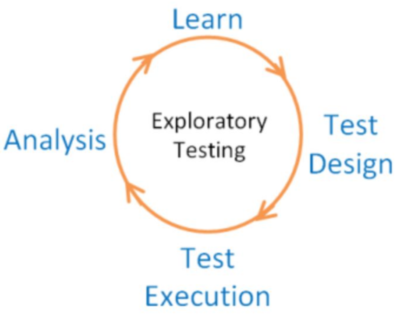


REACTIVE



http://www.reactivemanifesto.org/

Hoe gaan we dit testen?

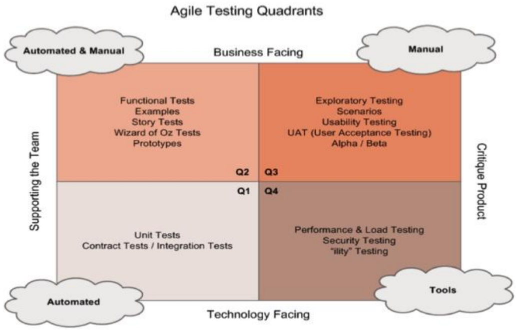

Learn

Exploratory Testing

Test Design

Test Execution

Analysis


Agile Testing Quadrants

Automated & Manual

Business Facing

Manual

Functional Tests
Examples
Story Tests
Wizard of Oz Tests
Prototypes

Q2

Exploratory Testing
Scenarios
Usability Testing
UAT (User Acceptance Testing)
Alpha / Beta

Q3

Supporting the Team

Critique Product

Q1

Unit Tests
Contract Tests / Integration Tests

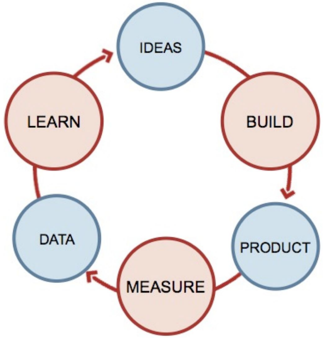
Q4

Performance & Load Testing
Security Testing
"ility" Testing

Automated

Tools

Technology Facing


IDEAS

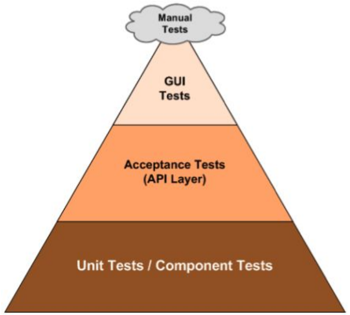
BUILD

PRODUCT

MEASURE

DATA

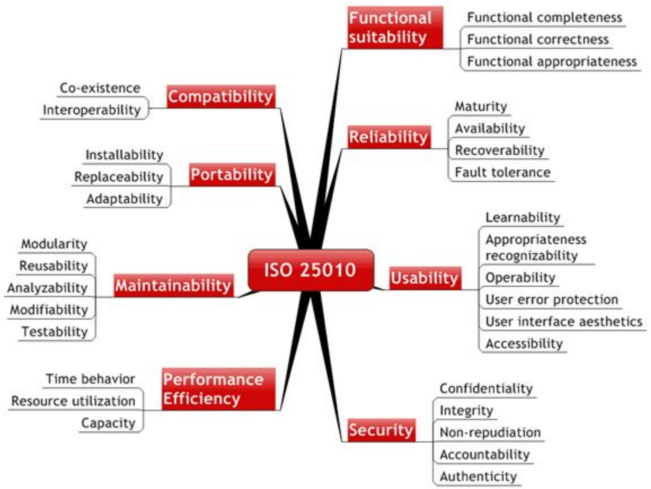
LEARN


Manual Tests

GUI Tests

Acceptance Tests (API Layer)

Unit Tests / Component Tests


ISO 25010

Functional suitability
- Functional completeness
- Functional correctness
- Functional appropriateness

Reliability
- Maturity
- Availability
- Recoverability
- Fault tolerance

Usability
- Learnability
- Appropriateness recognizability
- Operability
- User error protection
- User interface aesthetics
- Accessibility

Security
- Confidentiality
- Integrity
- Non-repudiation
- Accountability
- Authenticity

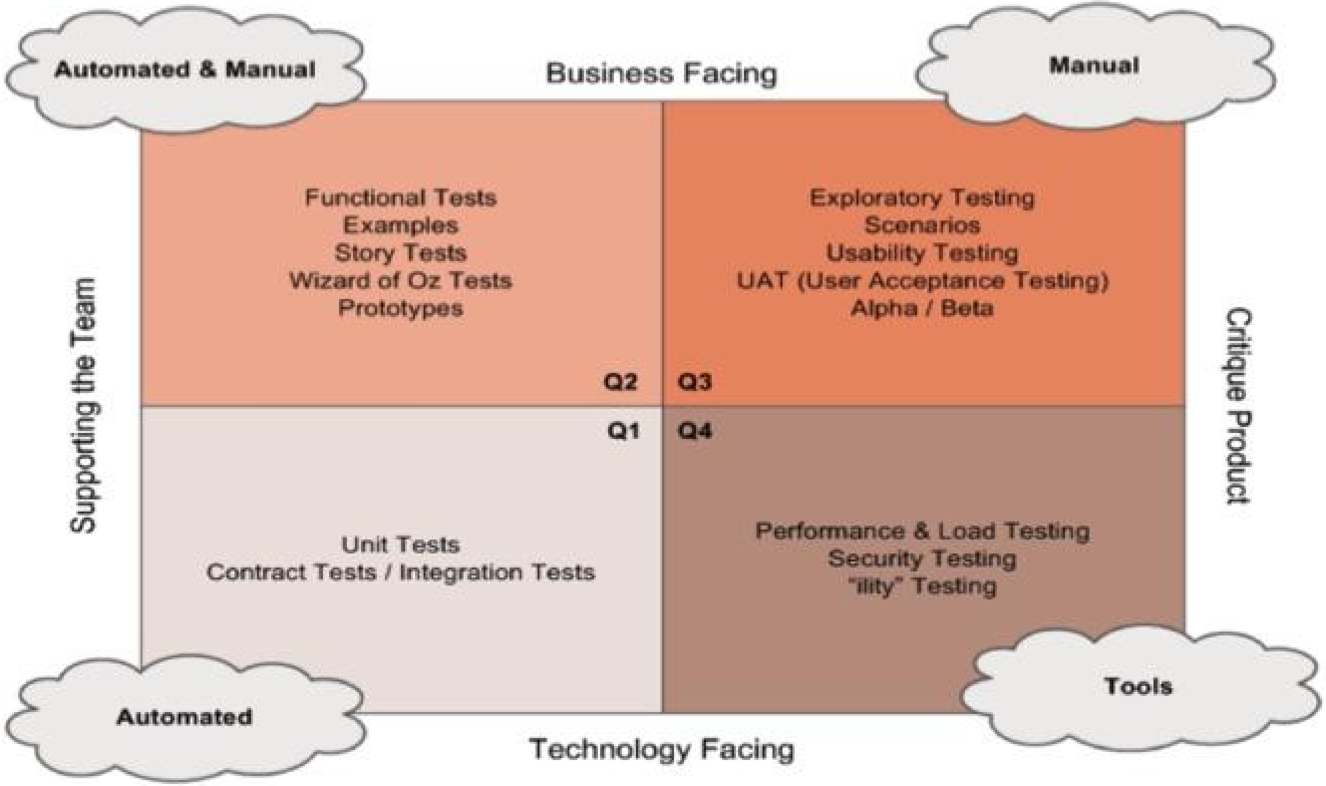
Compatibility
- Co-existence
- Interoperability

Portability
- Installability
- Replaceability
- Adaptability

Maintainability
- Modularity
- Reusability
- Analyzability
- Modifiability
- Testability

Performance Efficiency
- Time behavior
- Resource utilization
- Capacity

Agile Testing Quadrants



Automated & Manual

Business Facing

Manual

Supporting the Team

Critique Product

Q2

Functional Tests
Examples
Story Tests
Wizard of Oz Tests
Prototypes

Q3

Exploratory Testing
Scenarios
Usability Testing
UAT (User Acceptance Testing)
Alpha / Beta

Q1

Unit Tests
Contract Tests / Integration Tests

Q4

Performance & Load Testing
Security Testing
"ility" Testing

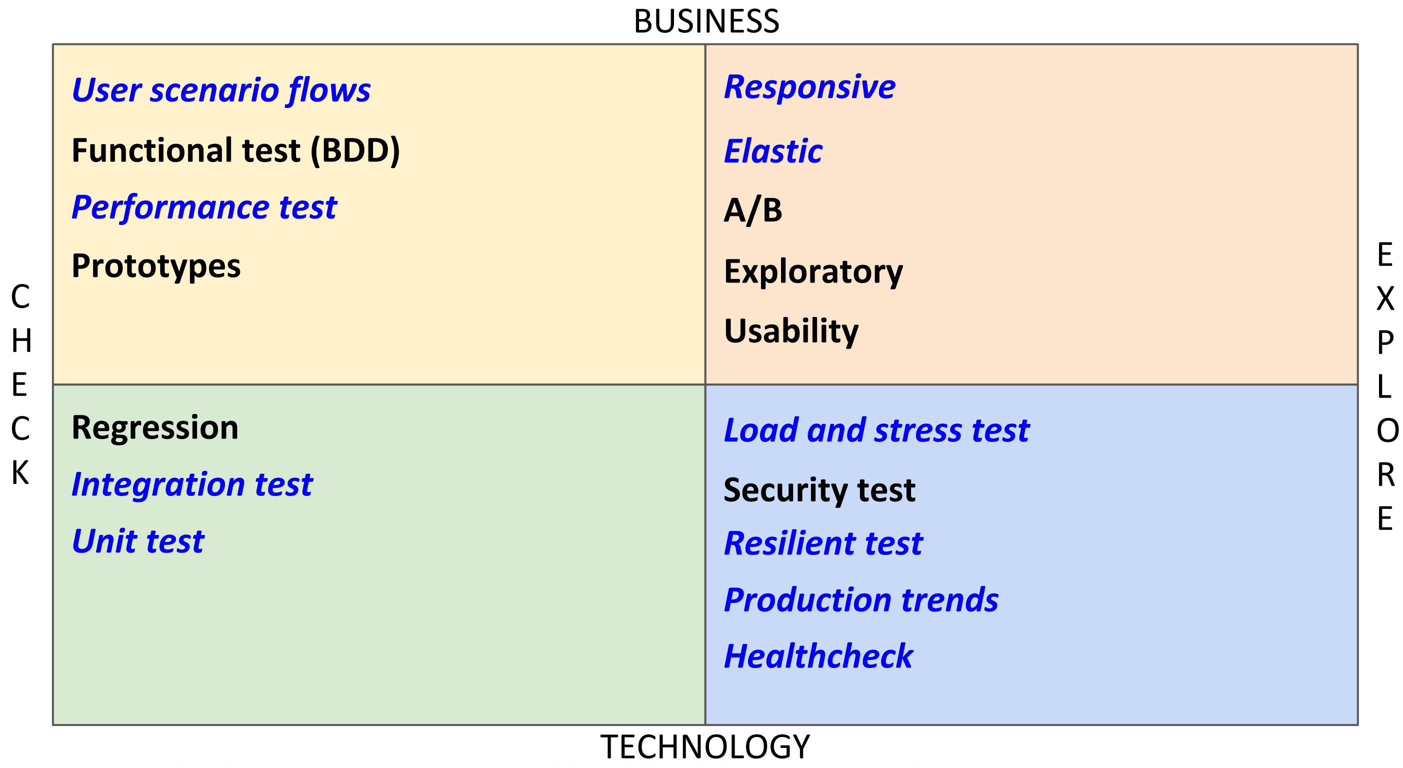
Automated

Technology Facing

Tools

http://lisacrispin.com/

Reactive testing quadrants



BUSINESS

CHECK			EXPLORE
	User scenario flows **Functional test (BDD)** ***Performance test*** **Prototypes**	***Responsive*** ***Elastic*** **A/B** **Exploratory** **Usability**	
	Regression ***Integration test*** ***Unit test***	***Load and stress test*** **Security test** ***Resilient test*** ***Production trends*** ***Healthcheck***	

TECHNOLOGY

Unit en integration

Unit en integration

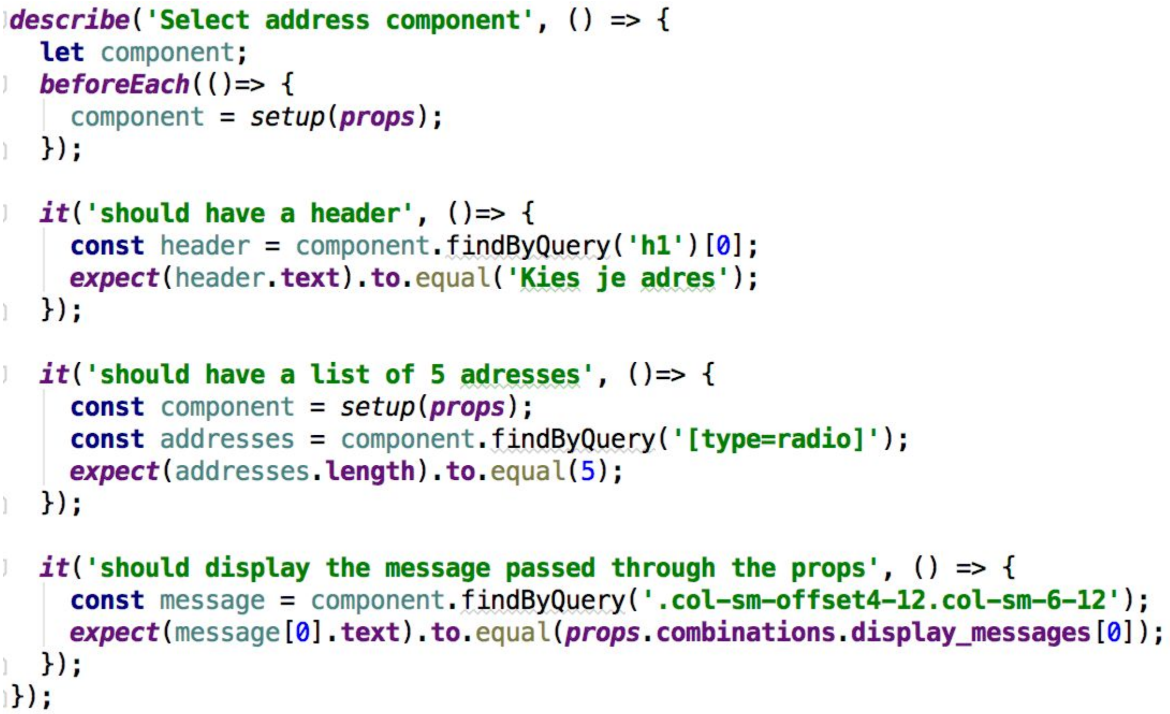
Jasmine is een behavior-driven framework voor het testen van JavaScript

Mocha is een test framewerk voor het testen van JavaScript in Node.js

Unit en integration

MOCHA

Jasmine



```
describe('Select address component', () => {
  let component;
  beforeEach(()=> {
    component = setup(props);
  });

  it('should have a header', ()=> {
    const header = component.findByQuery('h1')[0];
    expect(header.text).to.equal('Kies je adres');
  });

  it('should have a list of 5 adresses', ()=> {
    const component = setup(props);
    const addresses = component.findByQuery('[type=radio]');
    expect(addresses.length).to.equal(5);
  });

  it('should display the message passed through the props', () => {
    const message = component.findByQuery('.col-sm-offset4-12.col-sm-6-12');
    expect(message[0].text).to.equal(props.combinations.display_messages[0]);
  });
});
```

Unit en integration

>ScalaTest™

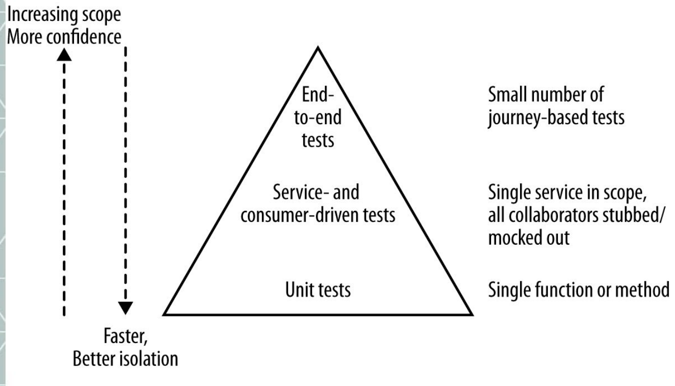
akka

Microservices gemaakt met scala en akka

Bijbehorende testframeworks zijn beschikbaar

Unit testing services

Integration testing services



Increasing scope
More confidence

End-to-end tests

Service- and consumer-driven tests

Unit tests

Small number of journey-based tests

Single service in scope, all collaborators stubbed/ mocked out

Single function or method

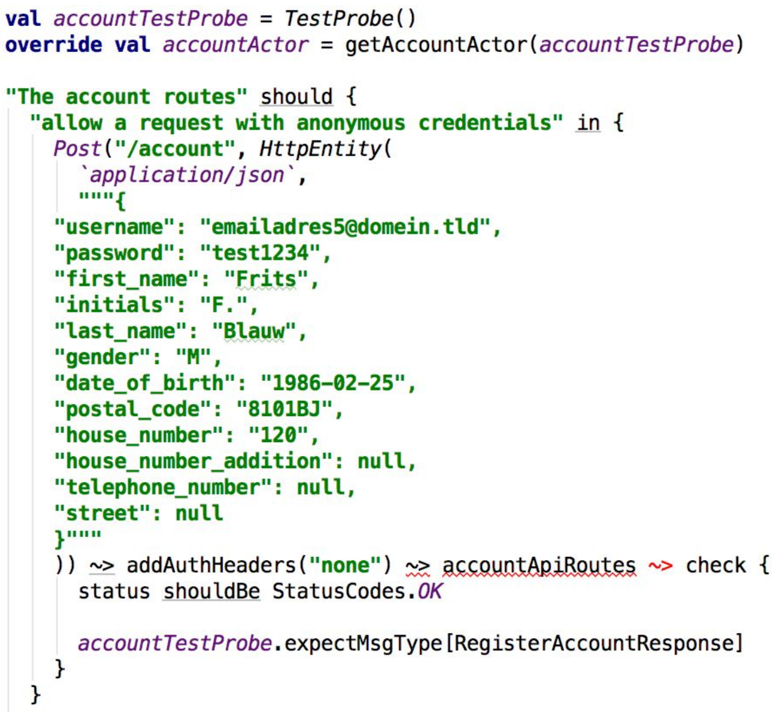
Faster,
Better isolation

Unit en integration

>ScalaTest™

akka

```
sbt test
sbt it:test
```

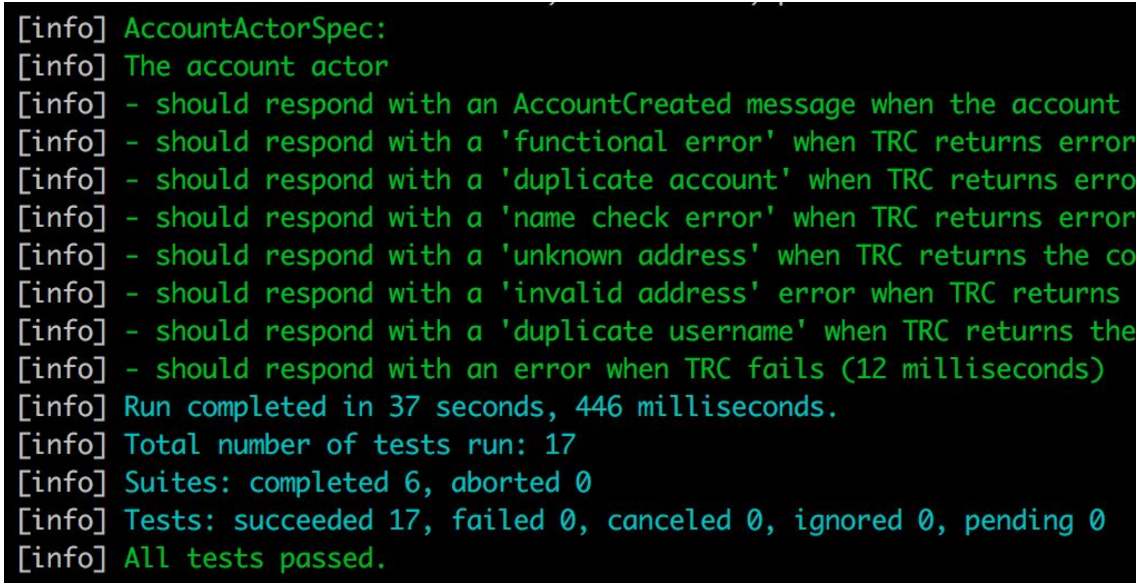


```scala
val accountTestProbe = TestProbe()
override val accountActor = getAccountActor(accountTestProbe)

"The account routes" should {
  "allow a request with anonymous credentials" in {
    Post("/account", HttpEntity(
      `application/json`,
      """{
    "username": "emailadres5@domein.tld",
    "password": "test1234",
    "first_name": "Frits",
    "initials": "F.",
    "last_name": "Blauw",
    "gender": "M",
    "date_of_birth": "1986-02-25",
    "postal_code": "8101BJ",
    "house_number": "120",
    "house_number_addition": null,
    "telephone_number": null,
    "street": null
    }"""
    )) ~> addAuthHeaders("none") ~> accountApiRoutes ~> check {
      status shouldBe StatusCodes.OK

      accountTestProbe.expectMsgType[RegisterAccountResponse]
    }
  }
```

Unit en integration

>ScalaTest™

akka

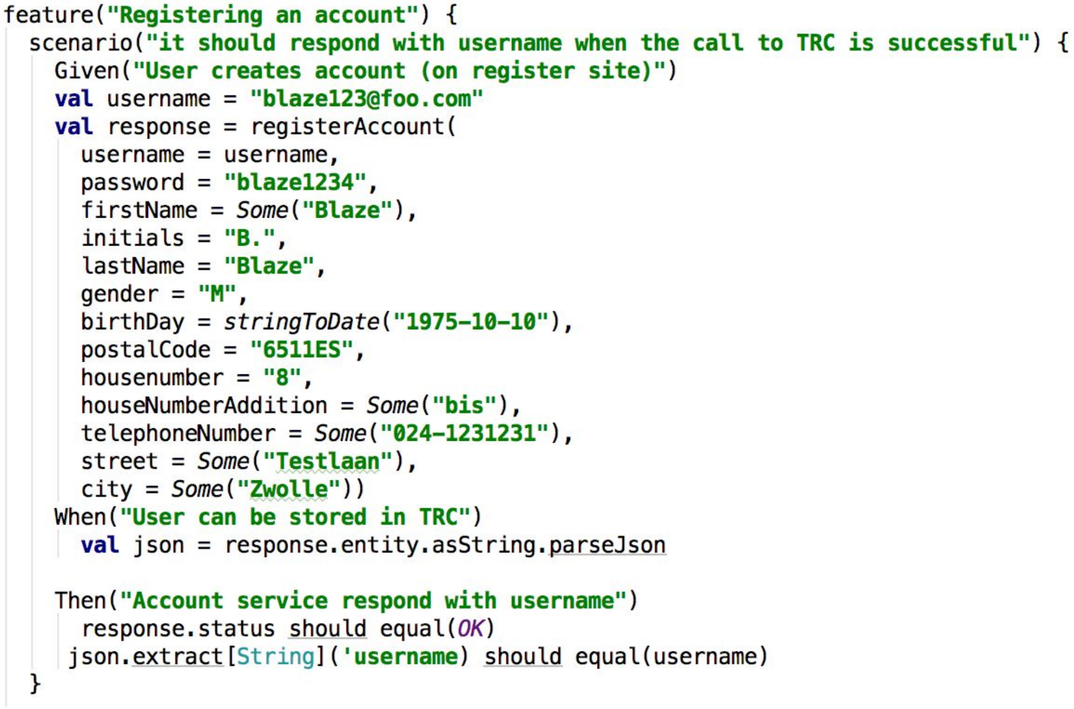


```
[info] AccountActorSpec:
[info] The account actor
[info] - should respond with an AccountCreated message when the account
[info] - should respond with a 'functional error' when TRC returns error
[info] - should respond with a 'duplicate account' when TRC returns erro
[info] - should respond with a 'name check error' when TRC returns error
[info] - should respond with a 'unknown address' when TRC returns the co
[info] - should respond with a 'invalid address' error when TRC returns
[info] - should respond with a 'duplicate username' when TRC returns the
[info] - should respond with an error when TRC fails (12 milliseconds)
[info] Run completed in 37 seconds, 446 milliseconds.
[info] Total number of tests run: 17
[info] Suites: completed 6, aborted 0
[info] Tests: succeeded 17, failed 0, canceled 0, ignored 0, pending 0
[info] All tests passed.
```

Unit en integration

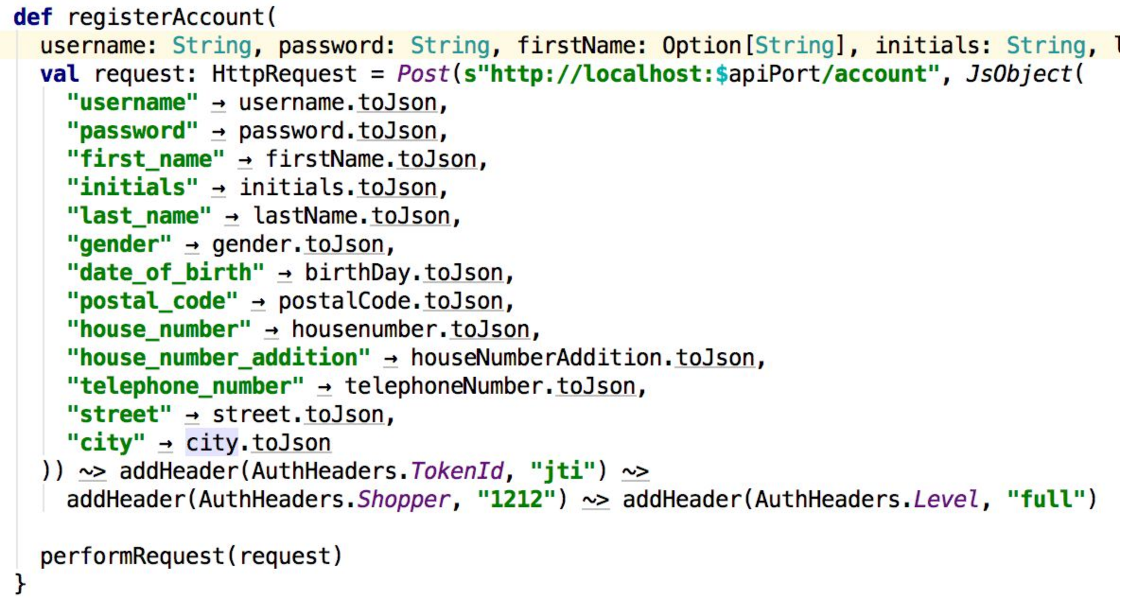
>ScalaTest™

akka



```
feature("Registering an account") {
  scenario("it should respond with username when the call to TRC is successful") {
    Given("User creates account (on register site)")
    val username = "blaze123@foo.com"
    val response = registerAccount(
      username = username,
      password = "blaze1234",
      firstName = Some("Blaze"),
      initials = "B.",
      lastName = "Blaze",
      gender = "M",
      birthDay = stringToDate("1975-10-10"),
      postalCode = "6511ES",
      housenumber = "8",
      houseNumberAddition = Some("bis"),
      telephoneNumber = Some("024-1231231"),
      street = Some("Testlaan"),
      city = Some("Zwolle"))
    When("User can be stored in TRC")
      val json = response.entity.asString.parseJson

    Then("Account service respond with username")
      response.status should equal(OK)
     json.extract[String]('username) should equal(username)
  }
```

Unit en integration



```
def registerAccount(
  username: String, password: String, firstName: Option[String], initials: String, l
  val request: HttpRequest = Post(s"http://localhost:$apiPort/account", JsObject(
    "username" → username.toJson,
    "password" → password.toJson,
    "first_name" → firstName.toJson,
    "initials" → initials.toJson,
    "last_name" → lastName.toJson,
    "gender" → gender.toJson,
    "date_of_birth" → birthDay.toJson,
    "postal_code" → postalCode.toJson,
    "house_number" → housenumber.toJson,
    "house_number_addition" → houseNumberAddition.toJson,
    "telephone_number" → telephoneNumber.toJson,
    "street" → street.toJson,
    "city" → city.toJson
  )) ~> addHeader(AuthHeaders.TokenId, "jti") ~>
    addHeader(AuthHeaders.Shopper, "1212") ~> addHeader(AuthHeaders.Level, "full")

  performRequest(request)
}
```

User scenario flows

Protractor

end to end testing for AngularJS

User scenario flows

Protractor
end to end testing for AngularJS

Protractor is een end-to-end test framework voor AngularJS applicaties.

Protractor voert testen uit in de browser, doorloopt de testen zoals een gebruiker de applicatie en browser zou gebruiken.

Ook bruikbaar bij andere applicaties, zoals DotNet of React

User scenario flows

Protractor
end to end testing for AngularJS

Testen voor pagina’s en flows binnen 1 site (1 applicatie) opgenomen in de repo van die applicatie

Testen over meerdere site heen in aparte E2E repo

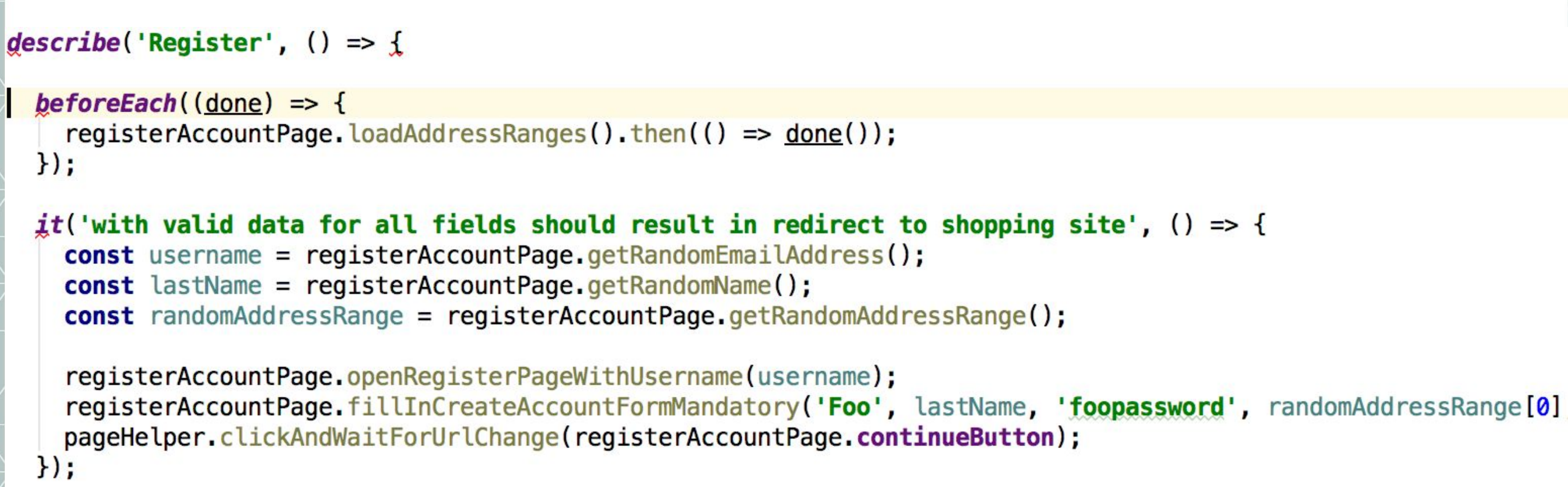
Gebruik Page Objects en Helpers

Hoe de testen klein, focus op een onderdeel

User scenario flows

Protractor

end to end testing for AngularJS

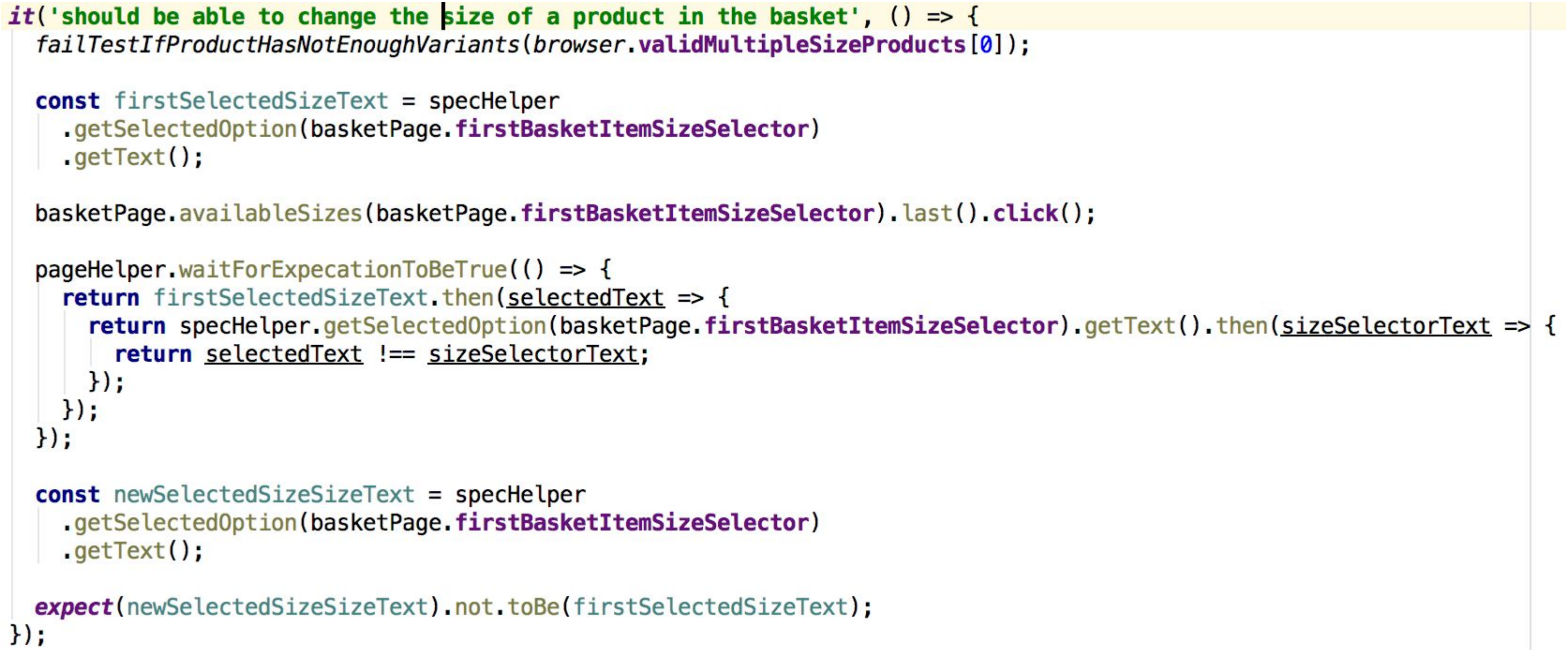


```
describe('Register', () => {

  beforeEach((done) => {
    registerAccountPage.loadAddressRanges().then(() => done());
  });

  it('with valid data for all fields should result in redirect to shopping site', () => {
    const username = registerAccountPage.getRandomEmailAddress();
    const lastName = registerAccountPage.getRandomName();
    const randomAddressRange = registerAccountPage.getRandomAddressRange();

    registerAccountPage.openRegisterPageWithUsername(username);
    registerAccountPage.fillInCreateAccountFormMandatory('Foo', lastName, 'foopassword', randomAddressRange[0]
    pageHelper.clickAndWaitForUrlChange(registerAccountPage.continueButton);
  });
```

Protractor

end to end testing for AngularJS

User scenario flows



```
it('should be able to change the size of a product in the basket', () => {
  failTestIfProductHasNotEnoughVariants(browser.validMultipleSizeProducts[0]);

  const firstSelectedSizeText = specHelper
    .getSelectedOption(basketPage.firstBasketItemSizeSelector)
    .getText();

  basketPage.availableSizes(basketPage.firstBasketItemSizeSelector).last().click();

  pageHelper.waitForExpecationToBeTrue(() => {
    return firstSelectedSizeText.then(selectedText => {
      return specHelper.getSelectedOption(basketPage.firstBasketItemSizeSelector).getText().then(sizeSelectorText => {
        return selectedText !== sizeSelectorText;
      });
    });
  });

  const newSelectedSizeSizeText = specHelper
    .getSelectedOption(basketPage.firstBasketItemSizeSelector)
    .getText();

  expect(newSelectedSizeSizeText).not.toBe(firstSelectedSizeText);
});
```

User scenario flows

Protractor
end to end testing for AngularJS

BrowserStack

SAUCELABS

docker

Testen draaien lokaal (tijdens development)

In Jenkins (tijdens deployment pipeline)

Opties:

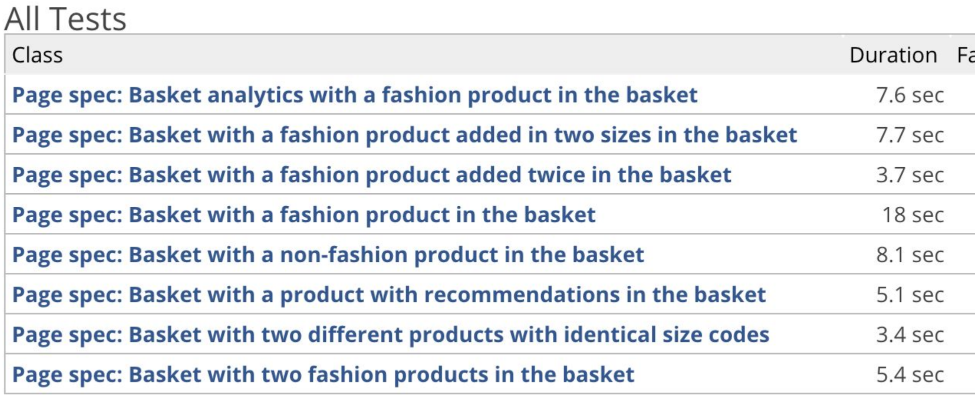
- Browserstack
- Saucelabs
- Eigen implementatie met docker

```
npm run devci_local_chrome:nl -- --suite basket-nl
```

User scenario flows

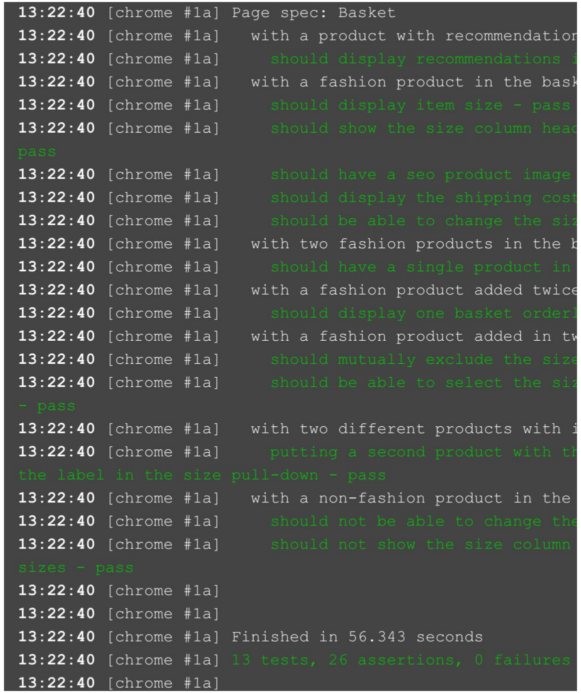
Protractor

end to end testing for AngularJS



All Tests

Class	Duration	Fa
Page spec: Basket analytics with a fashion product in the basket	7.6 sec	
Page spec: Basket with a fashion product added in two sizes in the basket	7.7 sec	
Page spec: Basket with a fashion product added twice in the basket	3.7 sec	
Page spec: Basket with a fashion product in the basket	18 sec	
Page spec: Basket with a non-fashion product in the basket	8.1 sec	
Page spec: Basket with a product with recommendations in the basket	5.1 sec	
Page spec: Basket with two different products with identical size codes	3.4 sec	
Page spec: Basket with two fashion products in the basket	5.4 sec	

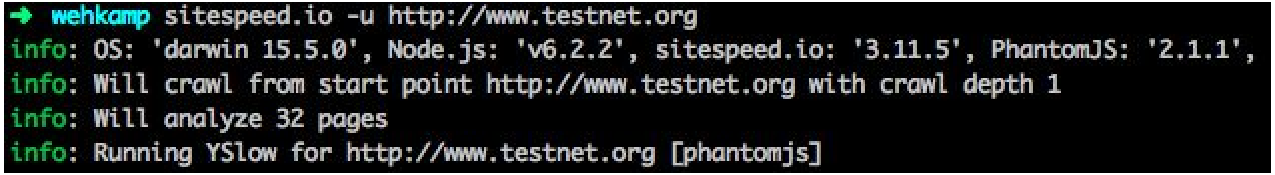


```
13:22:40 [chrome #1a] Page spec: Basket
13:22:40 [chrome #1a]   with a product with recommendatio
13:22:40 [chrome #1a]     should display recommendations i
13:22:40 [chrome #1a]   with a fashion product in the bask
13:22:40 [chrome #1a]     should display item size - pass
13:22:40 [chrome #1a]     should show the size column hea
pass
13:22:40 [chrome #1a]     should have a seo product image
13:22:40 [chrome #1a]     should display the shipping cost
13:22:40 [chrome #1a]     should be able to change the siz
13:22:40 [chrome #1a]   with two fashion products in the b
13:22:40 [chrome #1a]     should have a single product in
13:22:40 [chrome #1a]   with a fashion product added twice
13:22:40 [chrome #1a]     should display one basket orderl
13:22:40 [chrome #1a]   with a fashion product added in tw
13:22:40 [chrome #1a]     should mutually exclude the size
13:22:40 [chrome #1a]     should be able to select the siz
- pass
13:22:40 [chrome #1a]   with two different products with i
13:22:40 [chrome #1a]     putting a second product with th
the label in the size pull-down - pass
13:22:40 [chrome #1a]   with a non-fashion product in the
13:22:40 [chrome #1a]     should not be able to change the
13:22:40 [chrome #1a]     should not show the size column
sizes - pass
13:22:40 [chrome #1a]
13:22:40 [chrome #1a]
13:22:40 [chrome #1a] Finished in 56.343 seconds
13:22:40 [chrome #1a] 13 tests, 26 assertions, 0 failures
13:22:40 [chrome #1a]
```

Performance load en stress

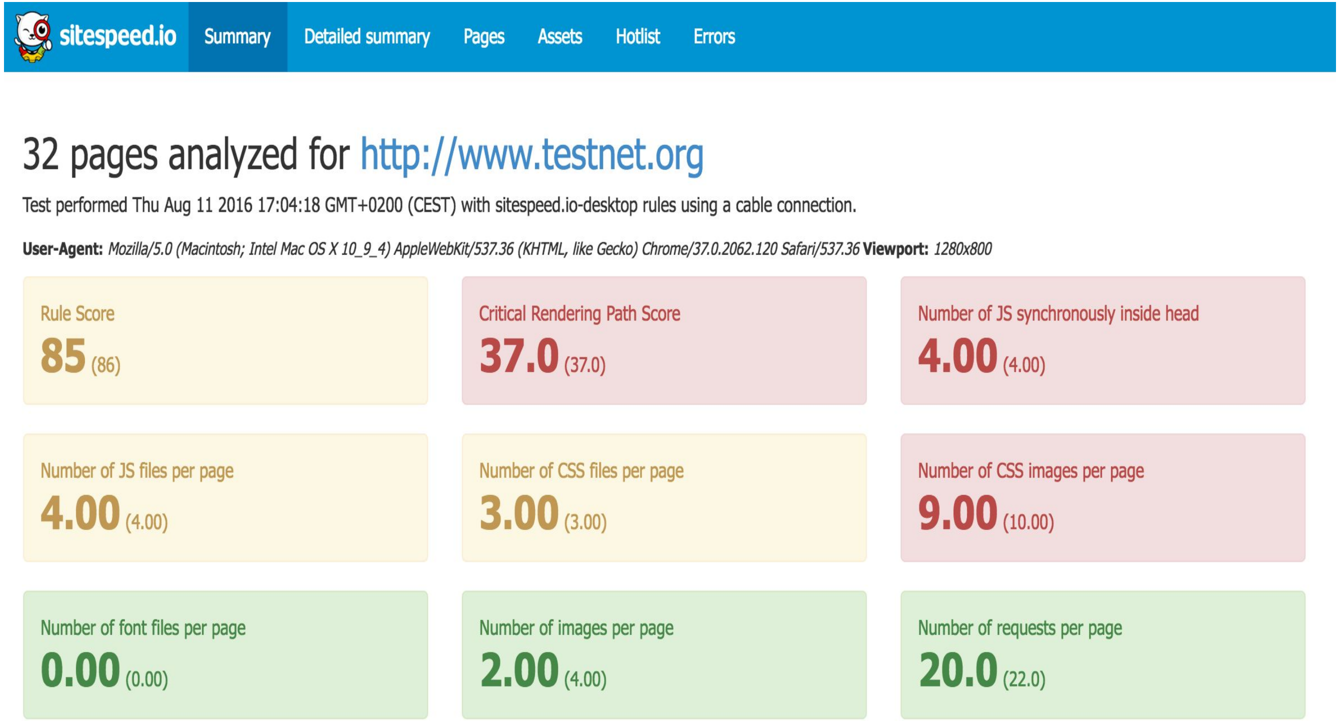
Performance load en stress

Sitespeed.io is een open source tool voor het anlyseren van speed en performance van je website. Gebaseerd op best practices en metrics



```
➜  wehkamp sitespeed.io -u http://www.testnet.org
info: OS: 'darwin 15.5.0', Node.js: 'v6.2.2', sitespeed.io: '3.11.5', PhantomJS: '2.1.1',
info: Will crawl from start point http://www.testnet.org with crawl depth 1
info: Will analyze 32 pages
info: Running YSlow for http://www.testnet.org [phantomjs]
```

Performance load en stress



32 pages analyzed for http://www.testnet.org

Test performed Thu Aug 11 2016 17:04:18 GMT+0200 (CEST) with sitespeed.io-desktop rules using a cable connection.

User-Agent: *Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.120 Safari/537.36* **Viewport:** *1280x800*

Rule Score
85 (86)

Critical Rendering Path Score
37.0 (37.0)

Number of JS synchronously inside head
4.00 (4.00)

Number of JS files per page
4.00 (4.00)

Number of CSS files per page
3.00 (3.00)

Number of CSS images per page
9.00 (10.00)

Number of font files per page
0.00 (0.00)

Number of images per page
2.00 (4.00)

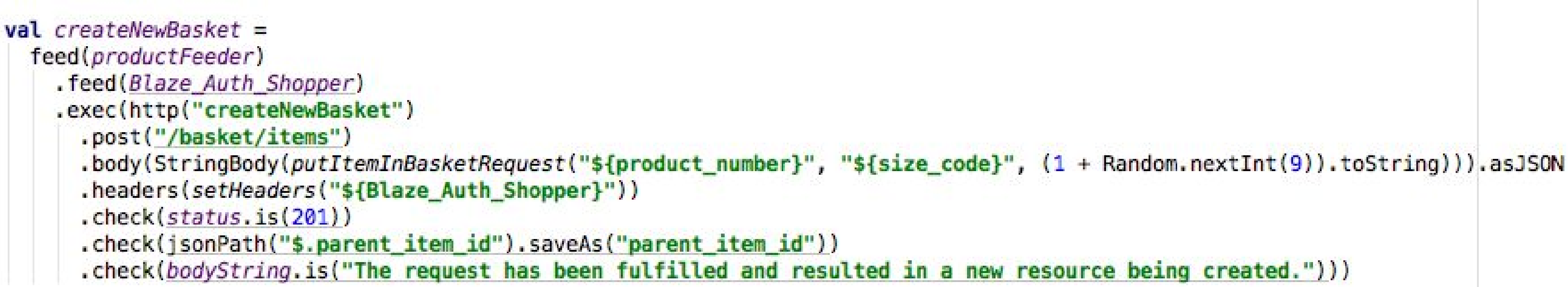
Number of requests per page
20.0 (22.0)

Performance load en stress

Gatling.io is een open source load testing framework gebasseerd op Scala, Akka en Netty.

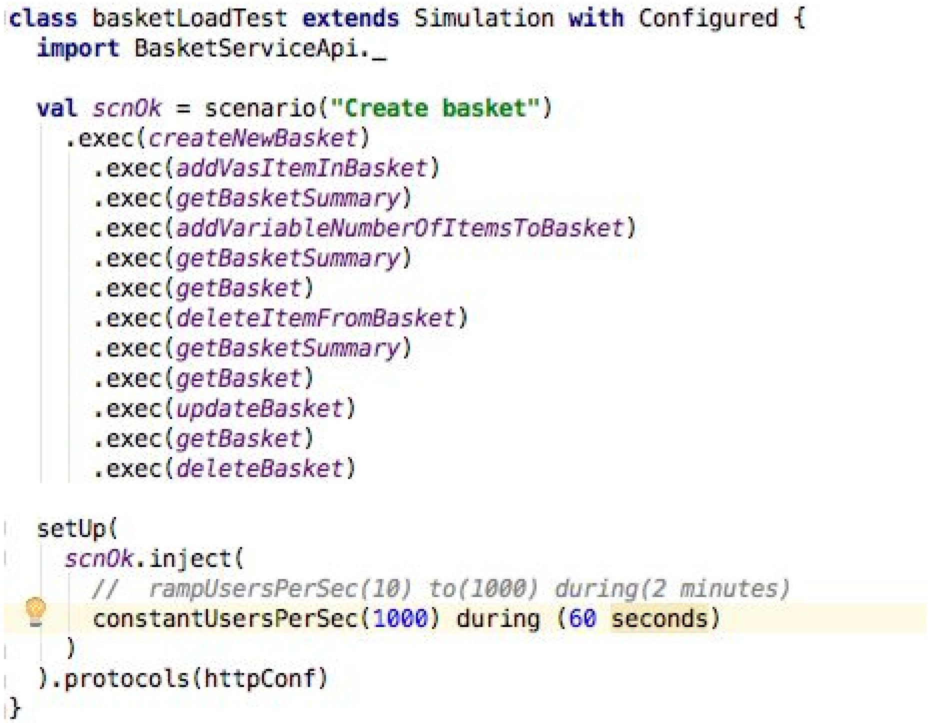
Doel is het analyseren en meten van performance van services

Performance load en stress



```
val createNewBasket =
  feed(productFeeder)
    .feed(Blaze_Auth_Shopper)
    .exec(http("createNewBasket")
      .post("/basket/items")
      .body(StringBody(putItemInBasketRequest("${product_number}", "${size_code}", (1 + Random.nextInt(9)).toString))).asJSON
      .headers(setHeaders("${Blaze_Auth_Shopper}"))
      .check(status.is(201))
      .check(jsonPath("$.parent_item_id").saveAs("parent_item_id"))
      .check(bodyString.is("The request has been fulfilled and resulted in a new resource being created.")))
```

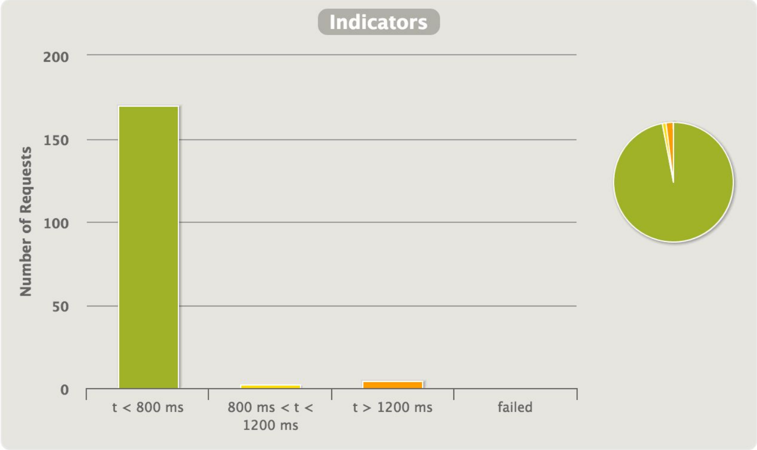
Performance load en stress

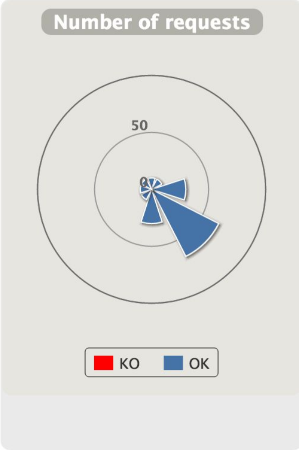


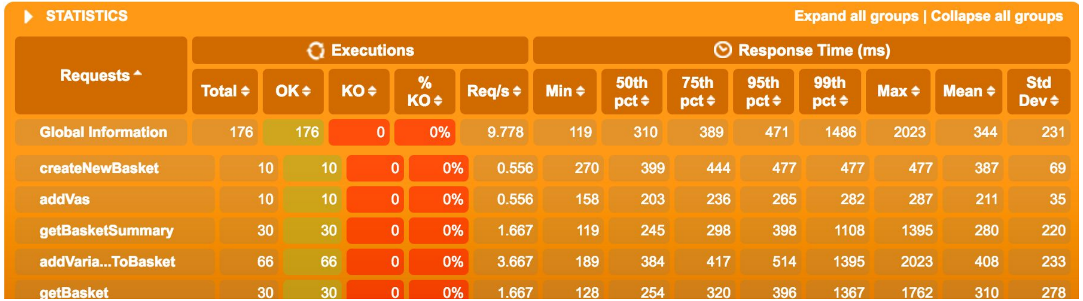
```
class basketLoadTest extends Simulation with Configured {
  import BasketServiceApi._

  val scnOk = scenario("Create basket")
    .exec(createNewBasket)
      .exec(addVasItemInBasket)
      .exec(getBasketSummary)
      .exec(addVariableNumberOfItemsToBasket)
      .exec(getBasketSummary)
      .exec(getBasket)
      .exec(deleteItemFromBasket)
      .exec(getBasketSummary)
      .exec(getBasket)
      .exec(updateBasket)
      .exec(getBasket)
      .exec(deleteBasket)

  setUp(
    scnOk.inject(
      //  rampUsersPerSec(10) to(1000) during(2 minutes)
      constantUsersPerSec(1000) during (60 seconds)
    )
  ).protocols(httpConf)
}
```

Gatling
STRESS TOOL

Performance load en stress





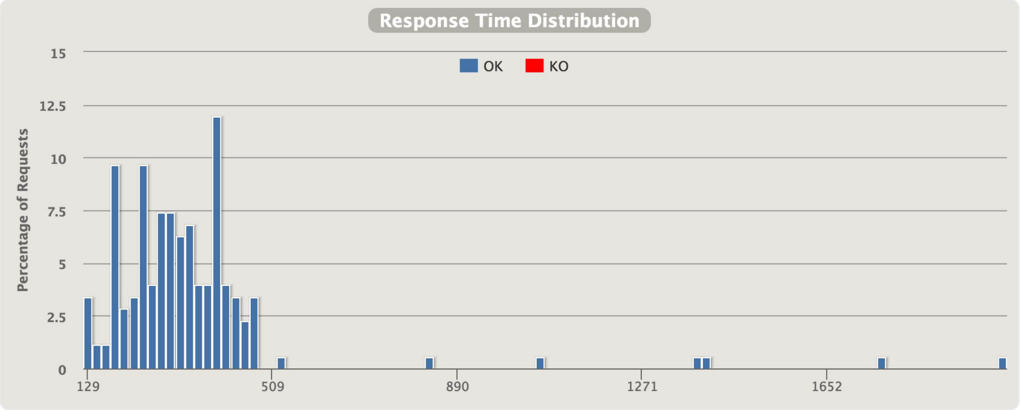


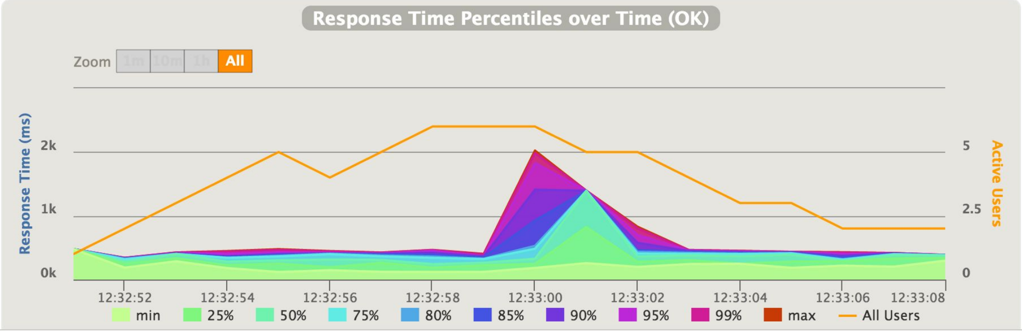
STATISTICS

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	176	176	0	0%	9.778	119	310	389	471	1486	2023	344	231
createNewBasket	10	10	0	0%	0.556	270	399	444	477	477	477	387	69
addVas	10	10	0	0%	0.556	158	203	236	265	282	287	211	35
getBasketSummary	30	30	0	0%	1.667	119	245	298	398	1108	1395	280	220
addVaria...ToBasket	66	66	0	0%	3.667	189	384	417	514	1395	2023	408	233
getBasket	30	30	0	0%	1.667	128	254	320	396	1367	1762	310	278

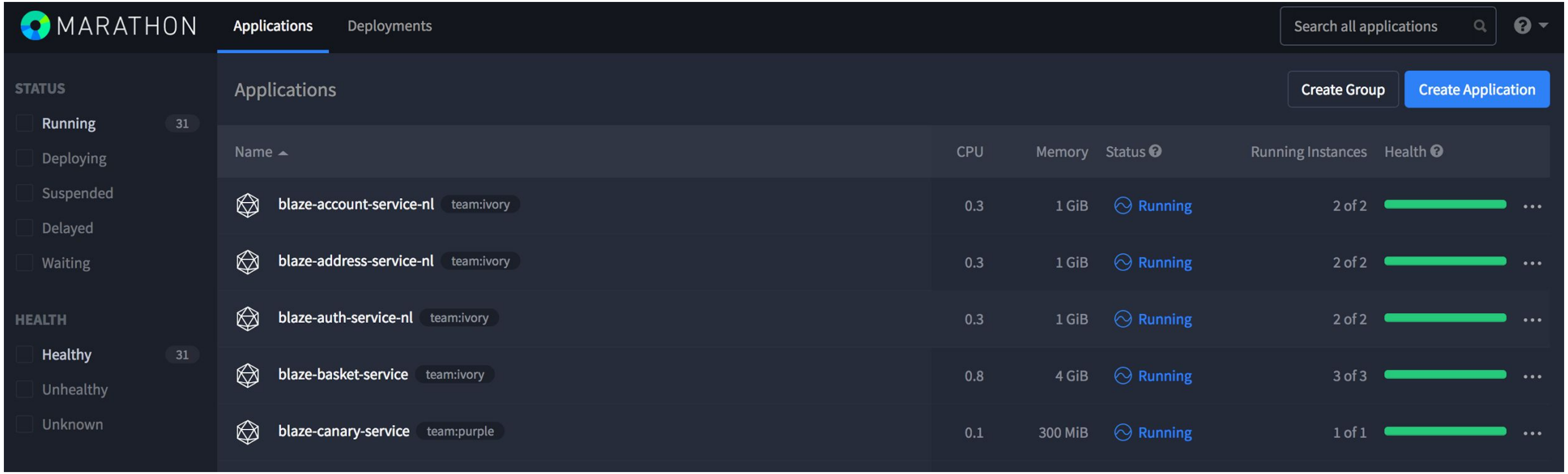
Performance load en stress





Production trends en healthcheck

Production trends en healthcheck



STATUS

- Running 31
- Deploying
- Suspended
- Delayed
- Waiting

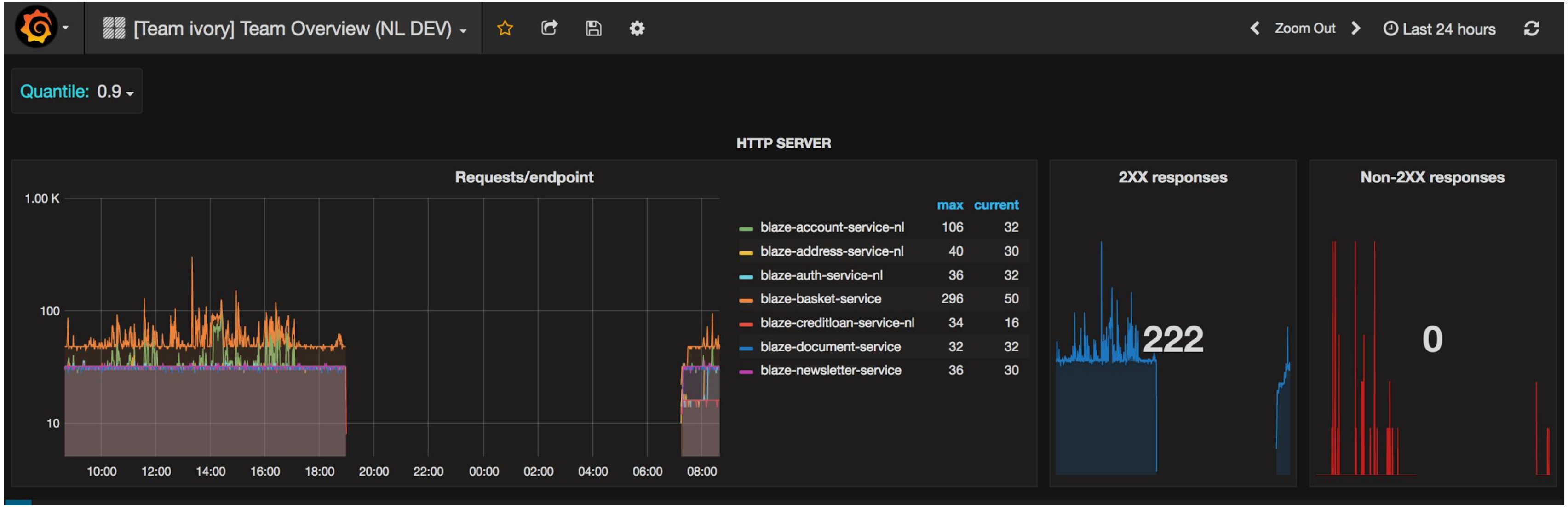
HEALTH

- Healthy 31
- Unhealthy
- Unknown

Applications

Name ▲	CPU	Memory	Status	Running Instances	Health
blaze-account-service-nl team:ivory	0.3	1 GiB	Running	2 of 2	🟩
blaze-address-service-nl team:ivory	0.3	1 GiB	Running	2 of 2	🟩
blaze-auth-service-nl team:ivory	0.3	1 GiB	Running	2 of 2	🟩
blaze-basket-service team:ivory	0.8	4 GiB	Running	3 of 3	🟩
blaze-canary-service team:purple	0.1	300 MiB	Running	1 of 1	🟩

Production trends en healthcheck



[Team ivory] Team Overview (NL DEV)

Quantile: 0.9

Last 24 hours

HTTP SERVER

Requests/endpoint

	max	current
🟩 blaze-account-service-nl	106	32
🟨 blaze-address-service-nl	40	30
🟦 blaze-auth-service-nl	36	32
🟧 blaze-basket-service	296	50
🟥 blaze-creditloan-service-nl	34	16
🟦 blaze-document-service	32	32
🟪 blaze-newsletter-service	36	30

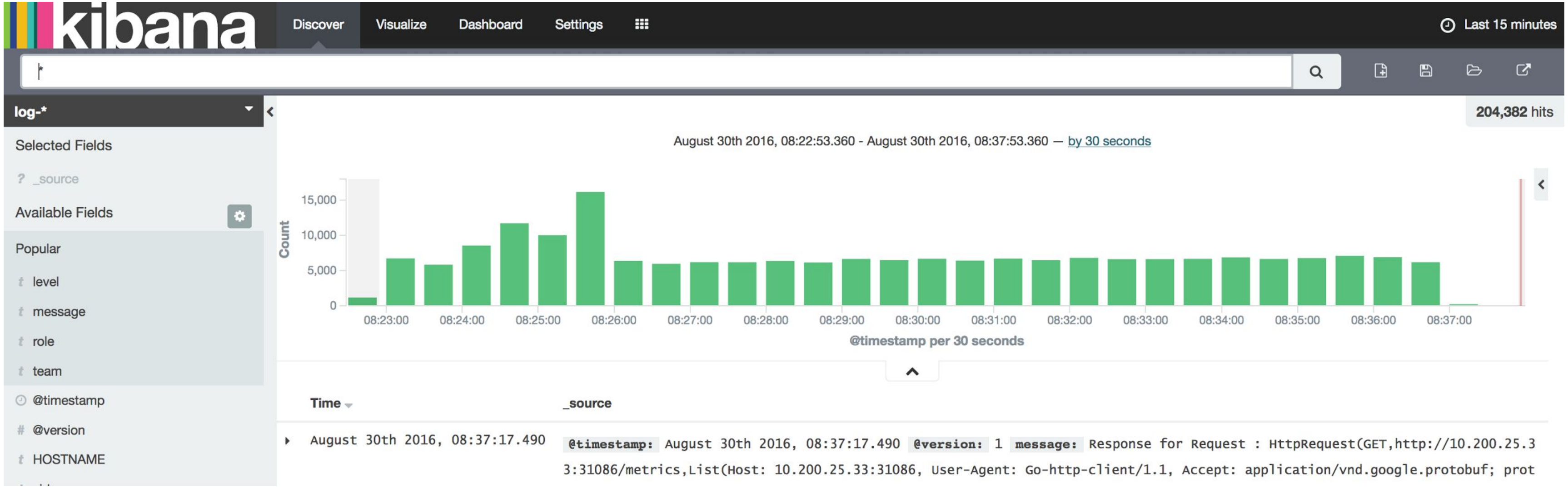
2XX responses

222

Non-2XX responses

0

Production trends en healthcheck

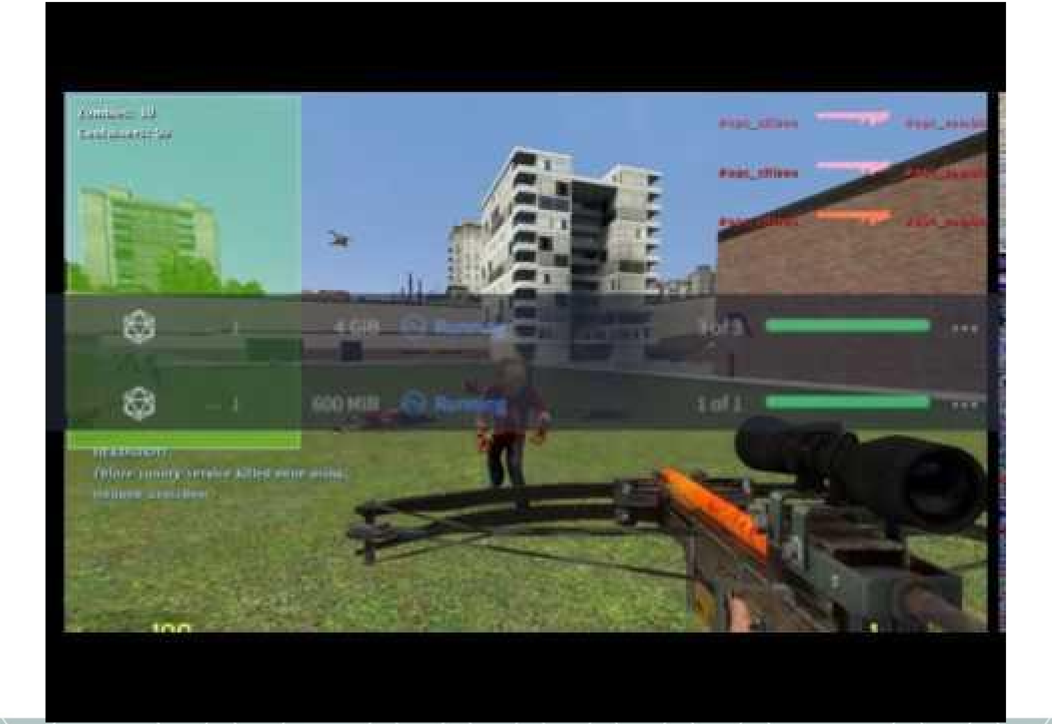


204,382 hits

August 30th 2016, 08:22:53.360 - August 30th 2016, 08:37:53.360 — by 30 seconds

```
August 30th 2016, 08:37:17.490   @timestamp: August 30th 2016, 08:37:17.490 @version: 1 message: Response for Request : HttpRequest(GET,http://10.200.25.3
                                 3:31086/metrics,List(Host: 10.200.25.33:31086, User-Agent: Go-http-client/1.1, Accept: application/vnd.google.protobuf; prot
```

Elastic, responsive, resilient



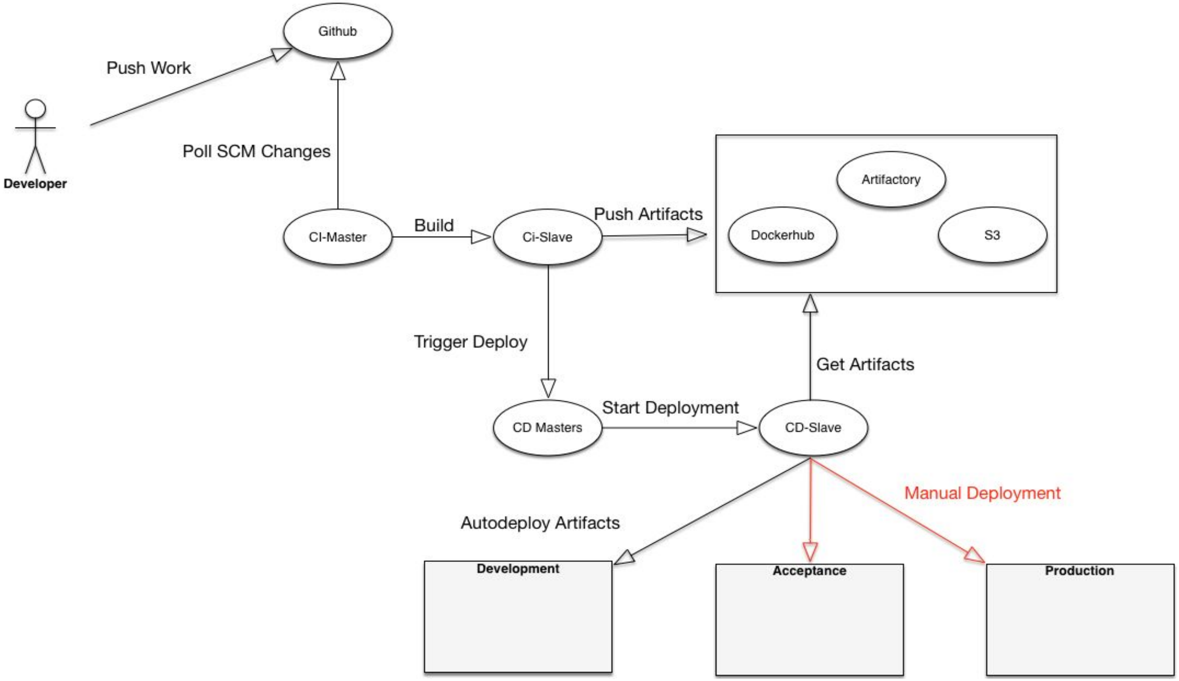
https://www.wehkamplabs.com/blog/2016/06/02/docker-and-zombies/

DEVELOPMENT PIPELINE

- Continuous Integration
- Continuous Delivery
- Continuous Deployment (?)

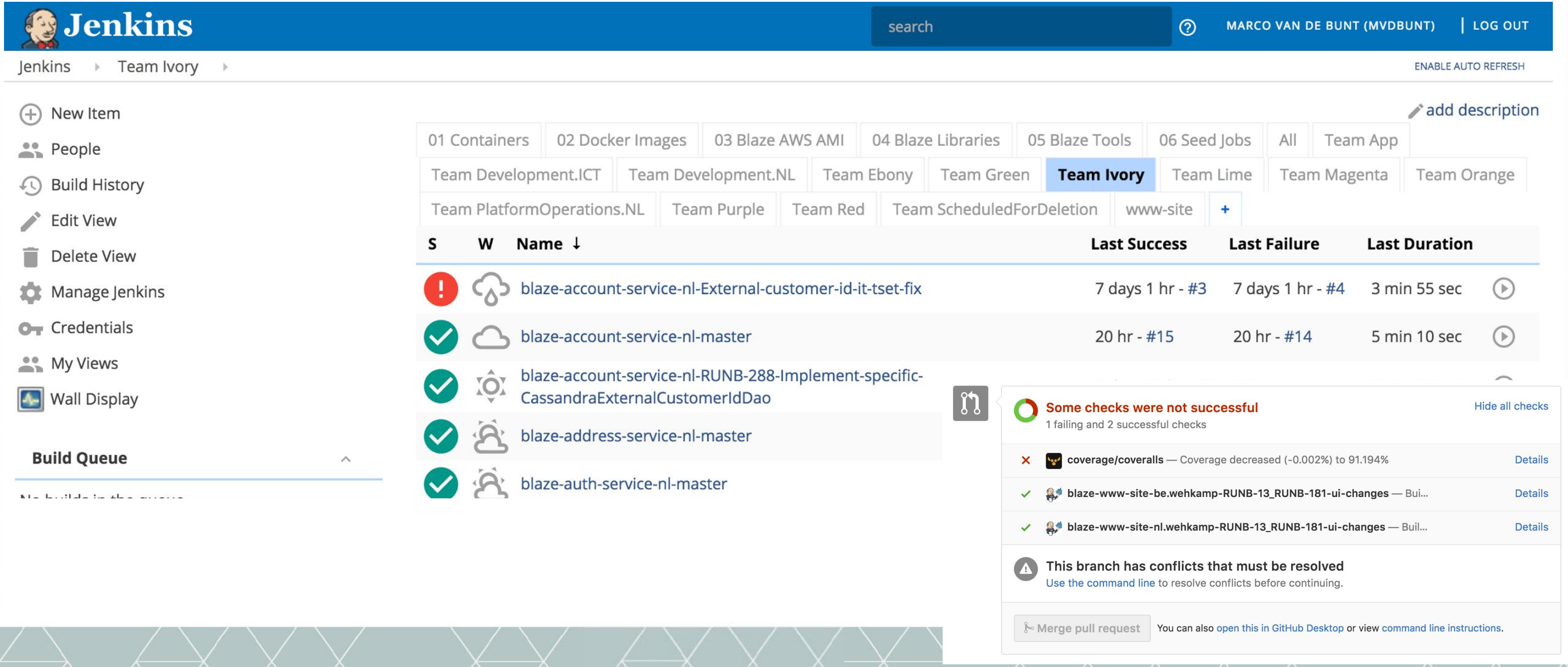
https://www.cloudbees.com/blog/guide-cutting-edge-jenkins-and-continuous-delivery

DEVELOPMENT PIPELINE



https://www.cloudbees.com/blog/guide-cutting-edge-jenkins-and-continuous-delivery

DEVELOPMENT PIPELINE



S	W	Name ↓	Last Success	Last Failure	Last Duration
!		blaze-account-service-nl-External-customer-id-it-tset-fix	7 days 1 hr - #3	7 days 1 hr - #4	3 min 55 sec
✓		blaze-account-service-nl-master	20 hr - #15	20 hr - #14	5 min 10 sec
✓		blaze-account-service-nl-RUNB-288-Implement-specific-CassandraExternalCustomerIdDao			
✓		blaze-address-service-nl-master			
✓		blaze-auth-service-nl-master			

Build Queue

No builds in the queue.

Some checks were not successful
1 failing and 2 successful checks

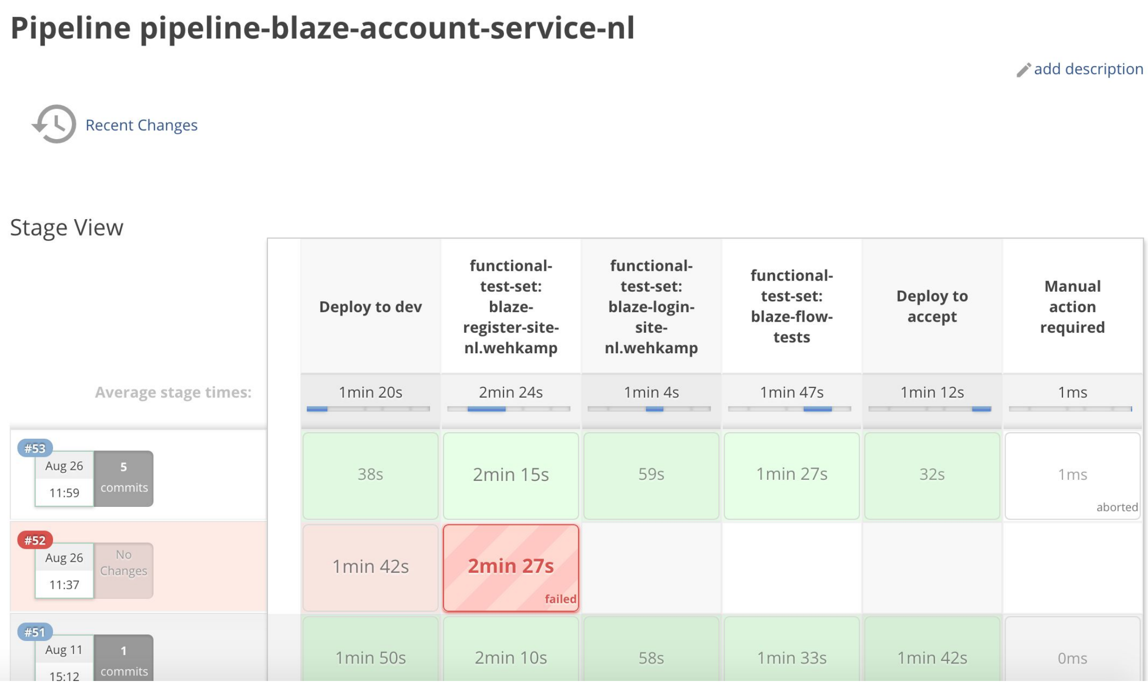
Hide all checks

- ✕ **coverage/coveralls** — Coverage decreased (-0.002%) to 91.194% Details
- ✓ **blaze-www-site-be.wehkamp-RUNB-13_RUNB-181-ui-changes** — Bui... Details
- ✓ **blaze-www-site-nl.wehkamp-RUNB-13_RUNB-181-ui-changes** — Buil... Details

This branch has conflicts that must be resolved
Use the command line to resolve conflicts before continuing.

Merge pull request — You can also open this in GitHub Desktop or view command line instructions.

DEVELOPMENT PIPELINE



Pipeline pipeline-blaze-account-service-nl

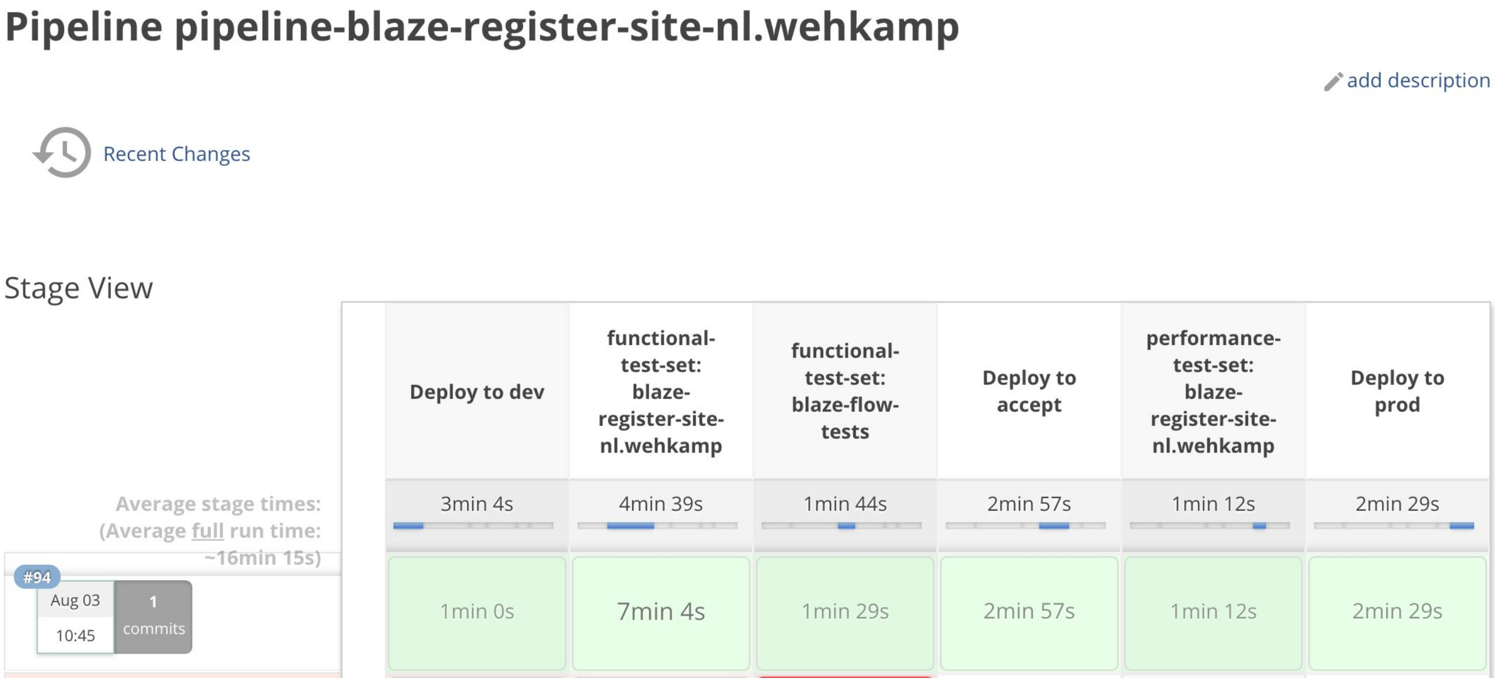
add description

Recent Changes

Stage View

	Deploy to dev	functional-test-set: blaze-register-site-nl.wehkamp	functional-test-set: blaze-login-site-nl.wehkamp	functional-test-set: blaze-flow-tests	Deploy to accept	Manual action required
Average stage times:	1min 20s	2min 24s	1min 4s	1min 47s	1min 12s	1ms
#53 Aug 26 11:59 5 commits	38s	2min 15s	59s	1min 27s	32s	1ms aborted
#52 Aug 26 11:37 No Changes	1min 42s	2min 27s failed				
#51 Aug 11 15:12 1 commits	1min 50s	2min 10s	58s	1min 33s	1min 42s	0ms

DEVELOPMENT PIPELINE



Pipeline pipeline-blaze-register-site-nl.wehkamp

add description

Recent Changes

Stage View

	Deploy to dev	functional-test-set: blaze-register-site-nl.wehkamp	functional-test-set: blaze-flow-tests	Deploy to accept	performance-test-set: blaze-register-site-nl.wehkamp	Deploy to prod
Average stage times: (Average full run time: ~16min 15s)	3min 4s	4min 39s	1min 44s	2min 57s	1min 12s	2min 29s
#94 Aug 03 10:45 1 commits	1min 0s	7min 4s	1min 29s	2min 57s	1min 12s	2min 29s

TOOL OVERVIEW

Grafana

GitHub

ScalaTest™

Jenkins

CONSUL

Gatling STRESS TOOL

Protractor
end to end testing for AngularJS

MARATHON

kibana

docker

COVERALLS

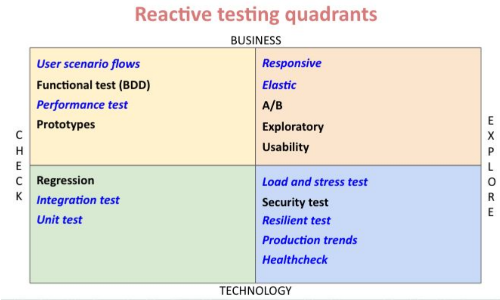
logstash

Bamboo

BrowserStack

amazon webservices™

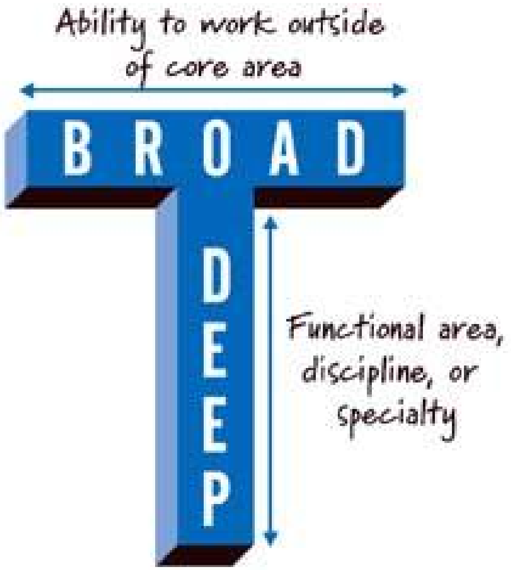
SAUCELABS



Reactive testing quadrants

	BUSINESS		
CHECK	*User scenario flows* Functional test (BDD) *Performance test* Prototypes	*Responsive* *Elastic* A/B Exploratory Usability	EXPLORE
CHECK	Regression *Integration test* *Unit test*	*Load and stress test* Security test *Resilient test* *Production trends* *Healthcheck*	EXPLORE
	TECHNOLOGY		

De veranderende testrol


Ability to work outside of core area
BROAD
DEEP
Functional area, discipline, or specialty

Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved

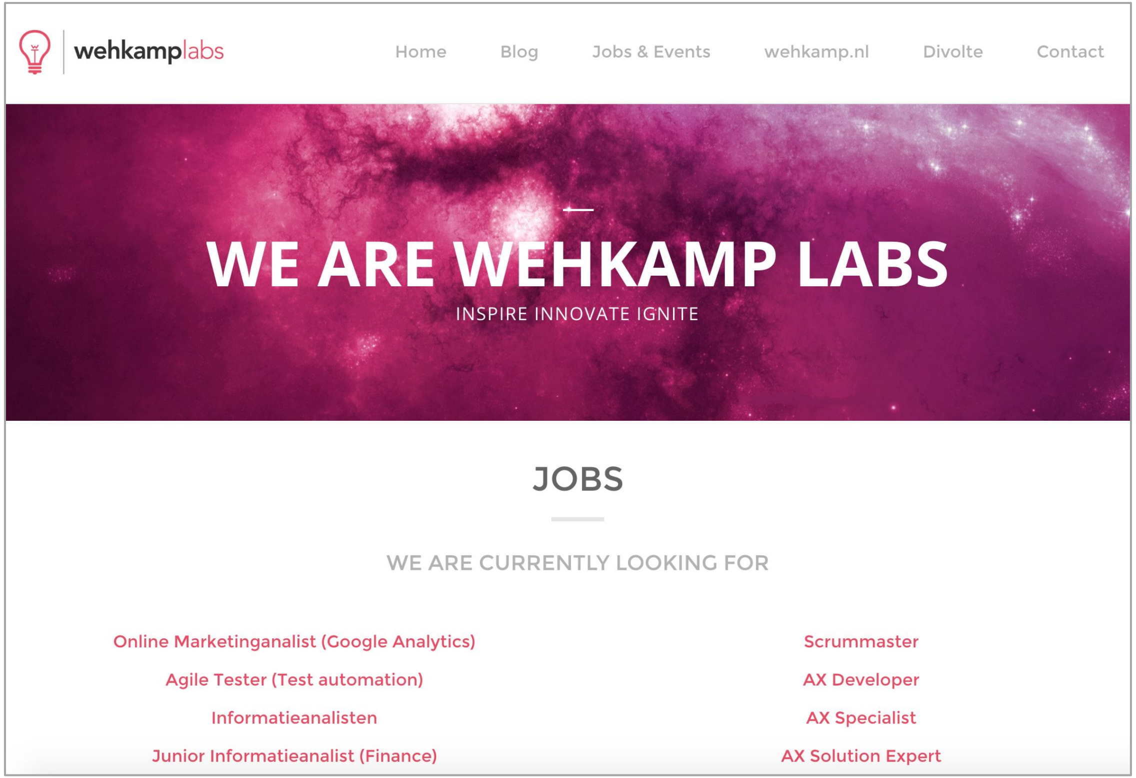
Testrol vervaagd

Test specialistisch ontwikkelen op het gebied van programmeren, automatiseren en tooling

Brede kennis van development- en beheerproces buiten testen om

Kansen!

www.wehkamplabs.com



WE ARE WEHKAMP LABS

INSPIRE INNOVATE IGNITE

JOBS

WE ARE CURRENTLY LOOKING FOR

Online Marketinganalist (Google Analytics)

Agile Tester (Test automation)

Informatieanalisten

Junior Informatieanalist (Finance)

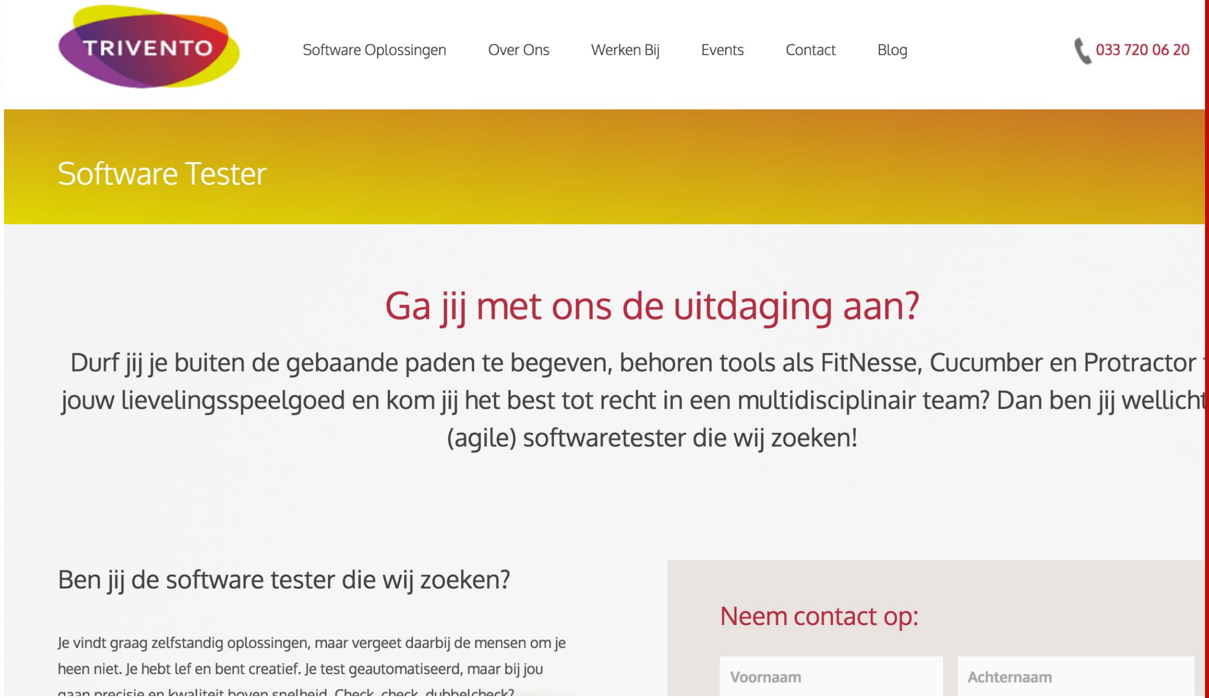
Scrummaster

AX Developer

AX Specialist

AX Solution Expert

www.trivento.nl



TRIVENTO

Software Oplossingen Over Ons Werken Bij Events Contact Blog

☎ 033 720 06 20

Software Tester

Ga jij met ons de uitdaging aan?

Durf jij je buiten de gebaande paden te begeven, behoren tools als FitNesse, Cucumber en Protractor t jouw lievelingsspeelgoed en kom jij het best tot recht in een multidisciplinair team? Dan ben jij wellicht (agile) softwaretester die wij zoeken!

Ben jij de software tester die wij zoeken?

Je vindt graag zelfstandig oplossingen, maar vergeet daarbij de mensen om je heen niet. Je hebt lef en bent creatief. Je test geautomatiseerd, maar bij jou gaan precisie en kwaliteit boven snelheid. Check, check, dubbelcheck?

Neem contact op:

Voornaam

Achternaam