Night watch

…with QA

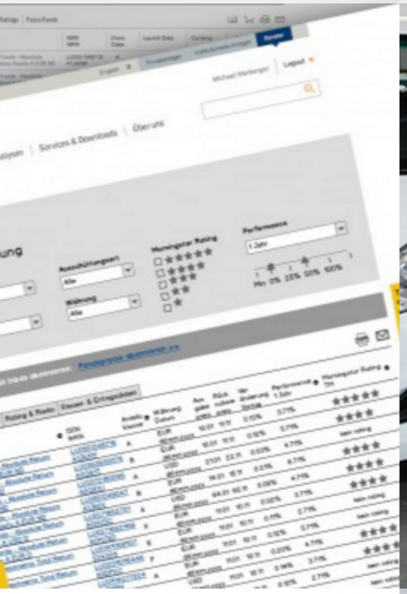
Carsten Sandtner (@casarock)
mediaman// Gesellschaft für Kommunikation mbH

about:me

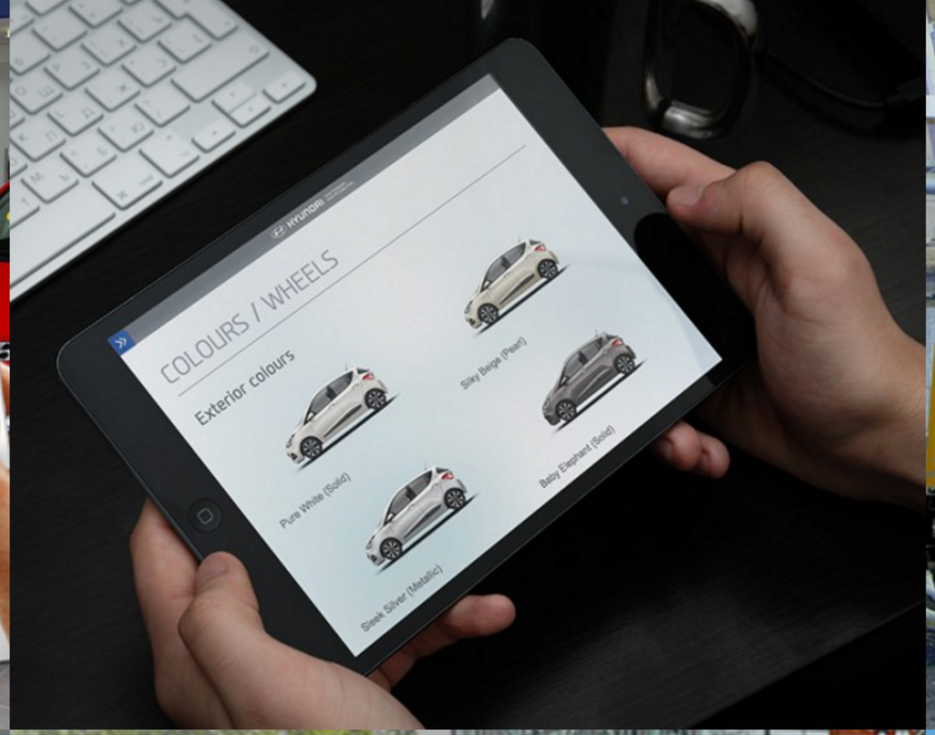
Carsten Sandtner
@casarock

Head of Software Development
//mediaman GmbH

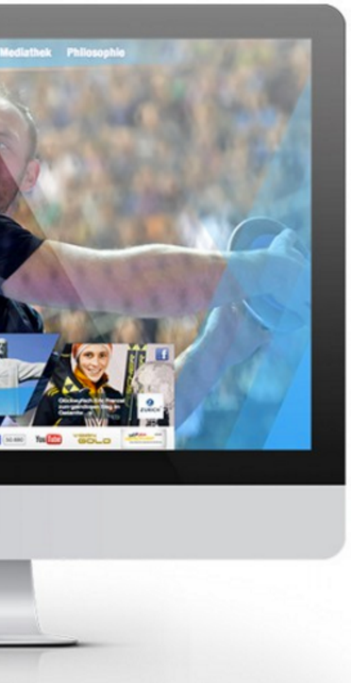
mediaman //

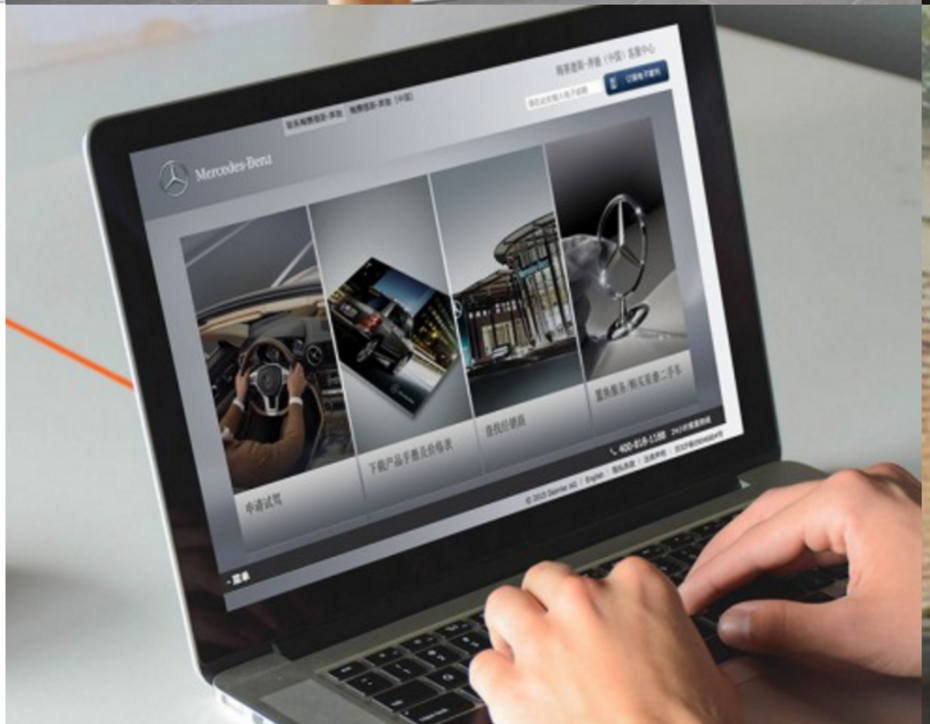



HYUNDAI
i20 5-Türer Style
Der große Kleine
ab 17.200,00 €
Bleiben Sie mit uns in Kontakt


COLOURS / WHEELS
Exterior colours
Silky Beige (Pearl)
Pure White (Solid)
Bitsy Elephant (Solid)
Sleek Silver (Metallic)


KAESER
M43


Mediathek
Philosophie

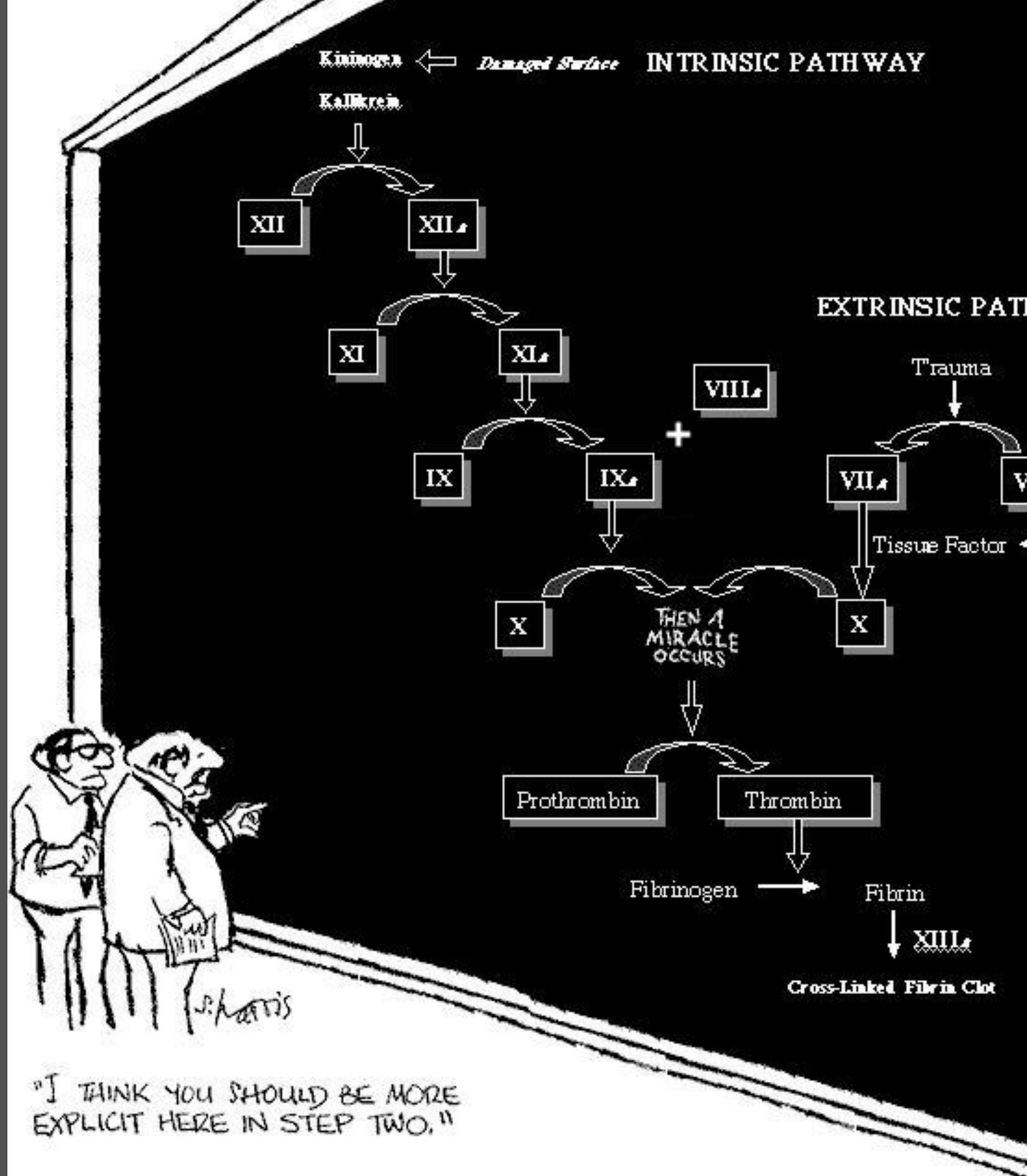

Mercedes-Benz


i20

Testing

Unit Tests

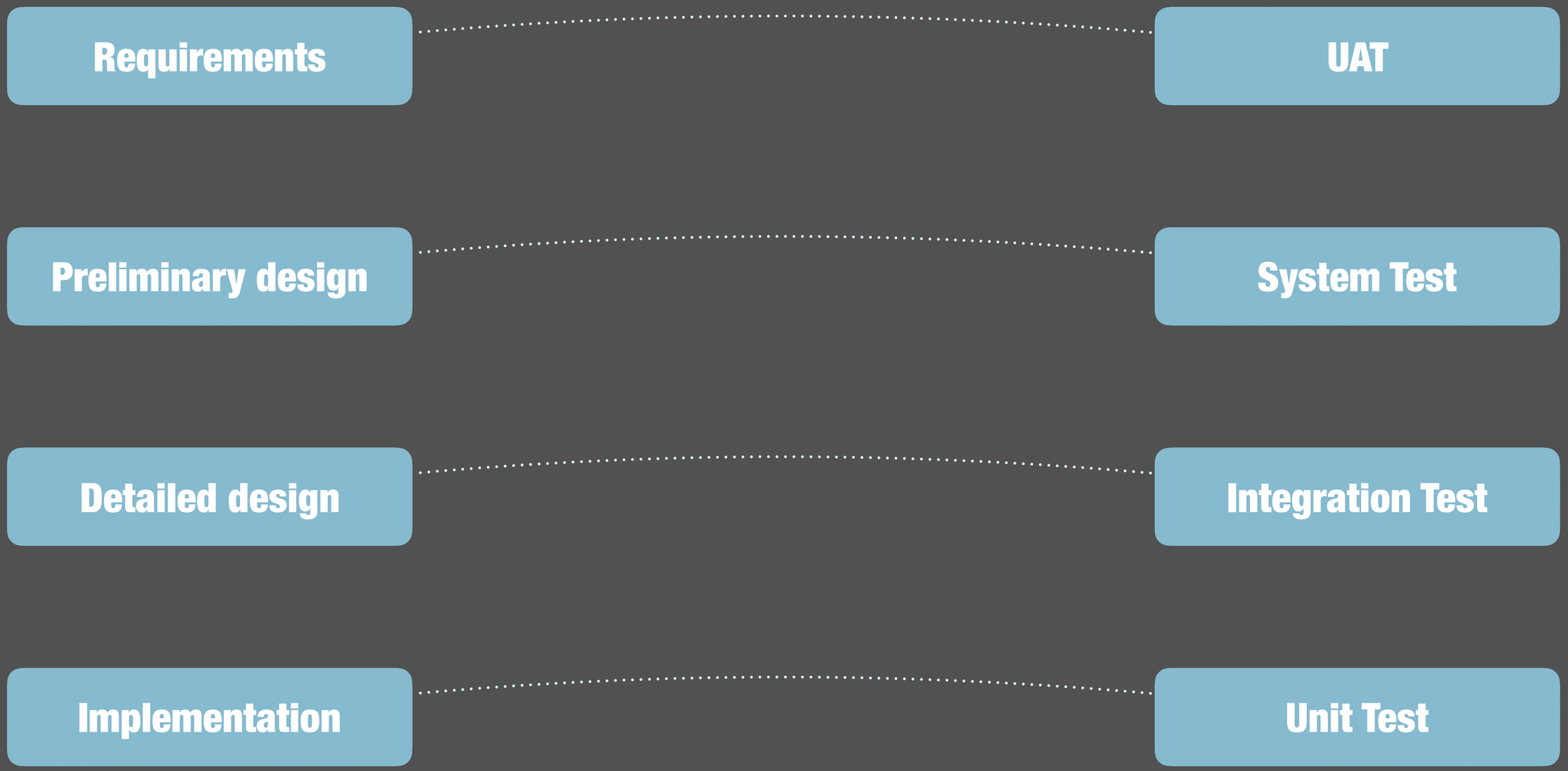
Integration Tests

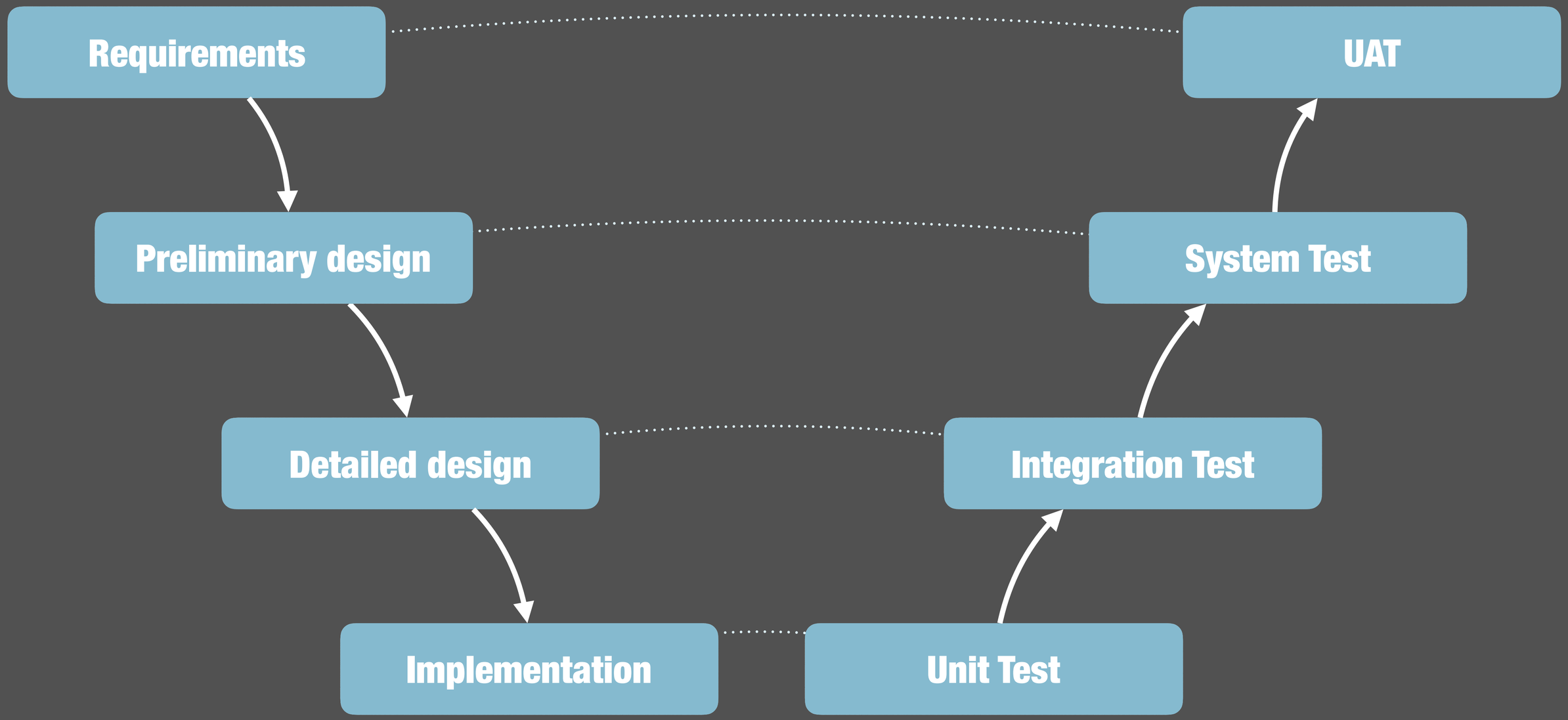


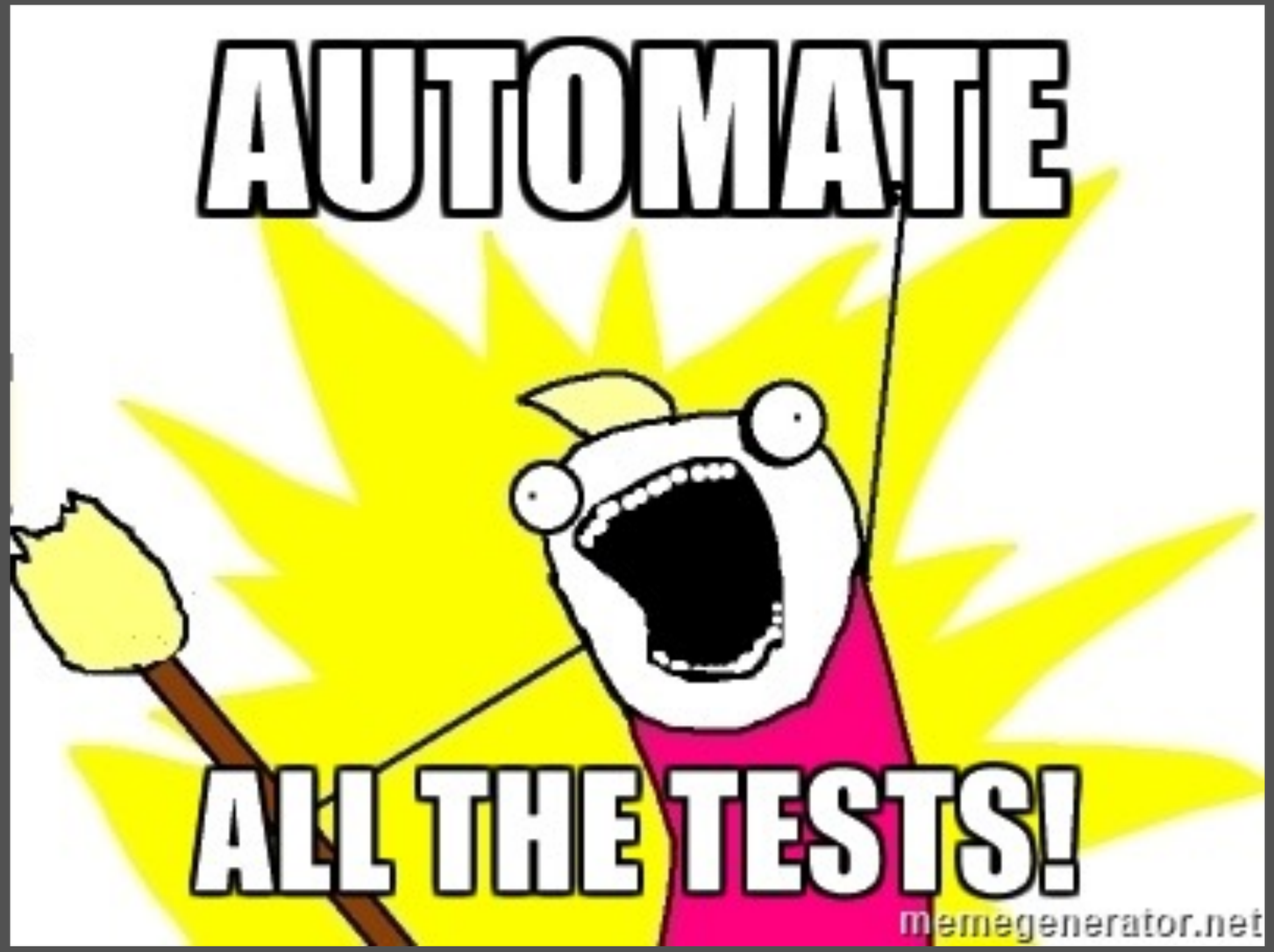
System Tests

National Museum of American History Smithsonian Institution - https://flic.kr/p/e6s1Lr

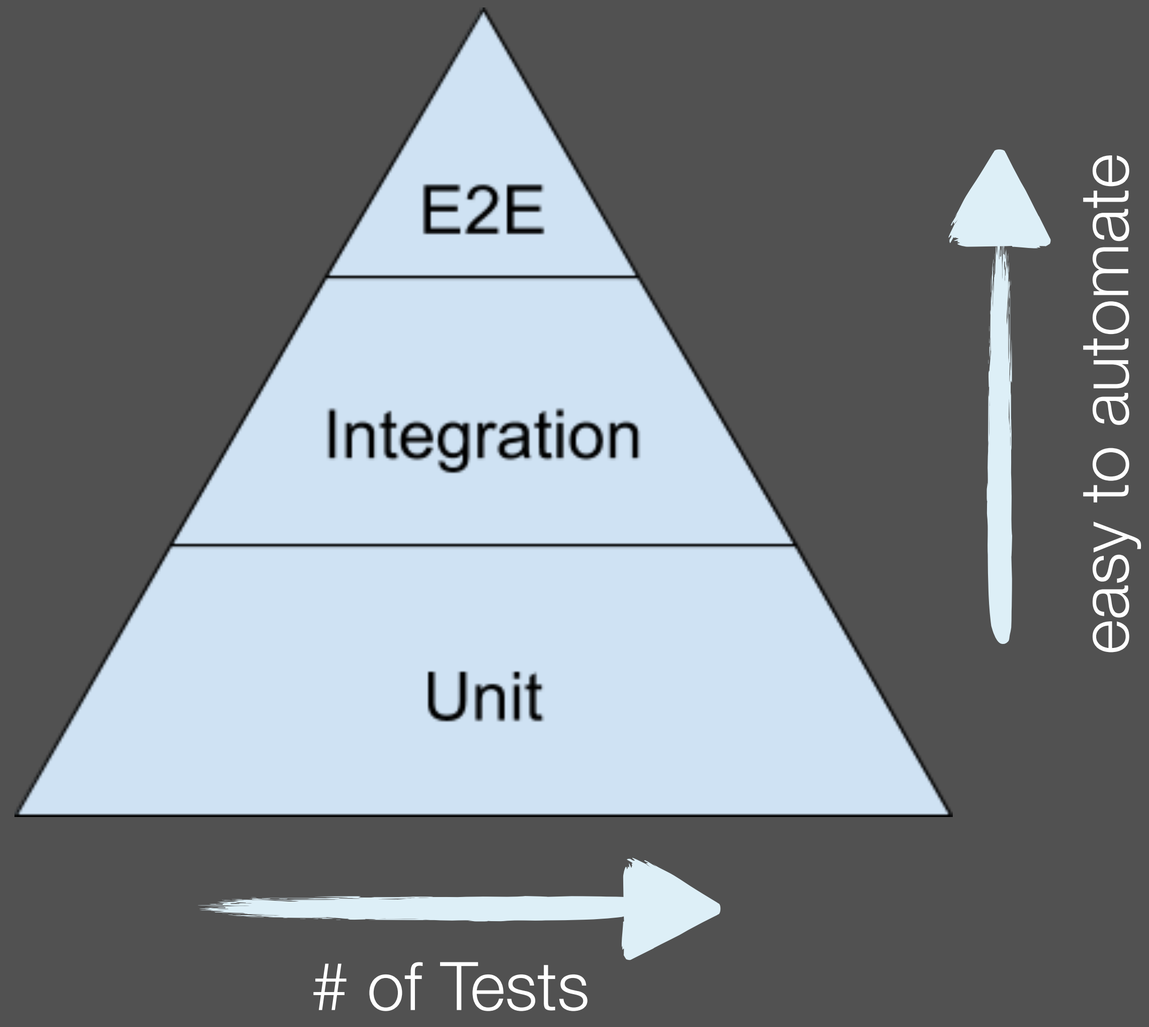
User Acceptance Tests


Requirements
UAT
Preliminary design
System Test
Detailed design
Integration Test
Implementation
Unit Test


Requirements
Preliminary design
Detailed design
Implementation
Unit Test
Integration Test
System Test
UAT


AUTOMATE
ALL THE TESTS!
memegenerator.net

Testing pyramid



TDD

Implementation

Unit Test

Detailed design

Integration Test

Implementation

Unit Test

Preliminary design

Detailed design

Implementation

System Test

Integration Test

Unit Test

Automated E2E Test

Protractor

end to end testing for AngularJS

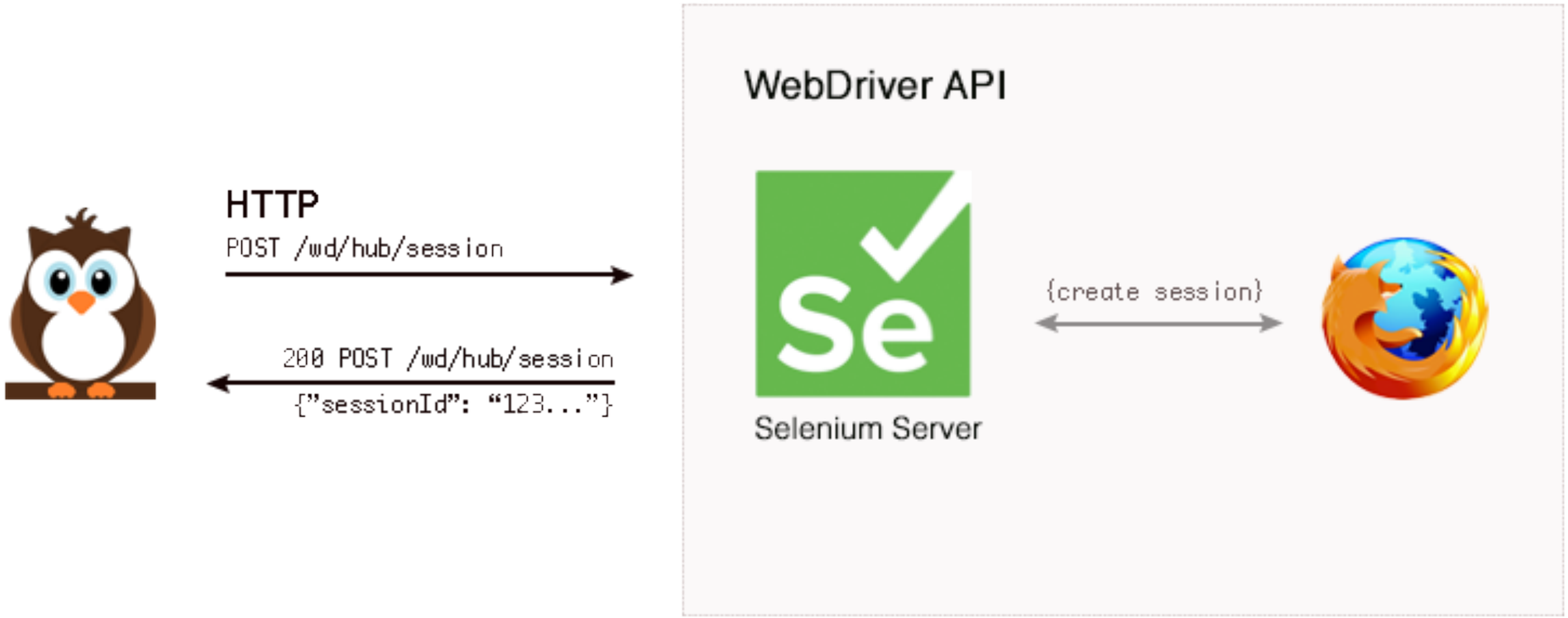
expecco

JS

„**Nightwatch.js** is an easy to use Node.js based End-to-End (E2E) testing solution for browser based apps and websites. It uses the powerful Selenium WebDriver API to perform commands and assertions on DOM elements.“

– http://nightwatchjs.org/

Nightwatch & WebDriver



Features

- Clean syntax
- Build-in test runner
- support for CSS and Xpath Selectors
- Grouping and filtering of Tests
- Saucelab and Browserstack support
- Extendable (Custom Commands)
- CI support!

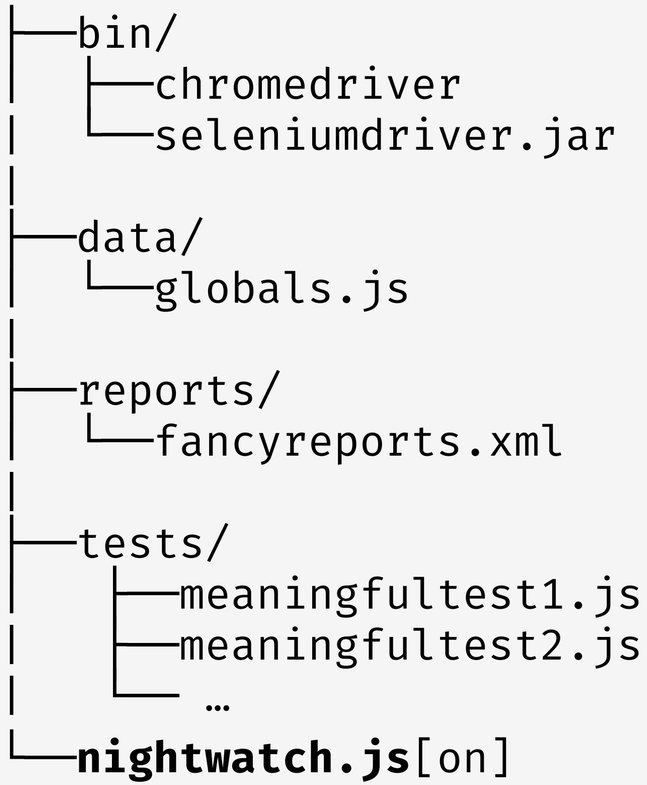
Installation

```
$ npm install [-g] nightwatch
```

… and Download Selenium-Server-Standalone!

Done.

Nightwatch project structure



Nightwatch.js

```
module.exports = {
    "src_folders": ["tests"],
    "output_folder": "reports",
    "custom_commands_path": "",
    "custom_assertions_path": "",
    "page_objects_path": "./objects/",
    "globals_path": "",

    "selenium": {
        "start_process": true,
        "server_path": "bin/selenium-server-standalone-2.53.1.jar",
        "log_path": "",
        "host": "127.0.0.1",
        "port": 4444,
        "cli_args": {
            "webdriver.chrome.driver": "./bin/chromedriver",
            "webdriver.ie.driver": "",
            "webdriver.gecko.driver" : "./bin/geckodriver090"
        }
    },
```

Nightwatch.js - Test setup.

```
"test_settings": {
        "default": {
            "launch_url": "http://localhost",
            "selenium_port": 4444,
            "selenium_host": "localhost",
            "silent": true,
            "screenshots": {
                "enabled": false,
                "path": ""
            },
            "desiredCapabilities": {
                "browserName": "chrome",
                "marionette": true
            }
        },
        “staging“: {

          . . .

       },
```

Nightwatch.js - Browser setup

```
        "chrome": {
            "desiredCapabilities": {
                "browserName": "chrome",
                "javascriptEnabled": true,
                "acceptSslCerts": true
            }
        }
    }
};
```

Simple Nightwatch test

```
module.exports = {
  'Google Webtechcon' : function (client) {
    client
      .url('http://www.google.com')
      .waitForElementVisible('body', 1000)
      .assert.title('Google')
      .assert.visible('input[type=text]')
      .setValue('input[type=text]', 'webtechcon')
      .waitForElementVisible('button[name=btnG]', 1000)
      .click('button[name=btnG]')
      .pause(1000)
      .assert.containsText('#rso > div.g > div > div > h3 > a',
        'WebTech Conference 2016')
      .end();
  }
};
```

Test execution

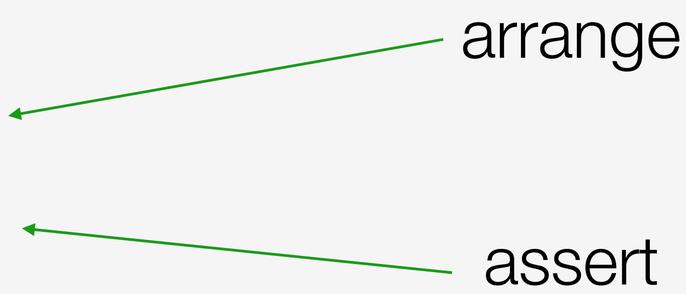
```
$ nightwatch [-c nightwatch.js] tests/simplegoogle.js
```


1. bash
casa_ at MMCasaBook Pro in ~/develop/nightwatch on master*
$ nightwatch -c nightwatch.js tests/simplegoogle.js

Tests in detail

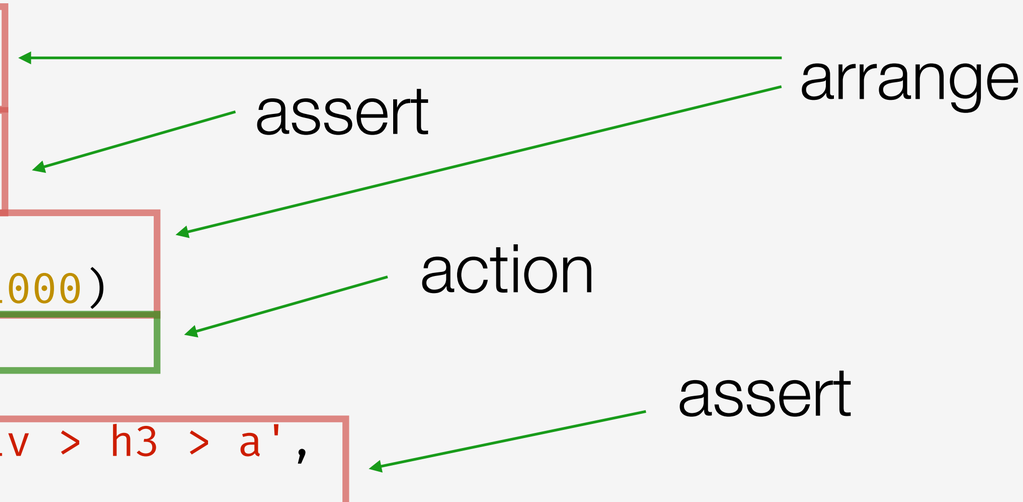
Arrange and assert

```
module.exports = {
  'Google Webtechcon' : function (client) {
    client
      .url('http://www.google.com')
      .waitForElementVisible('body', 1000)
      .assert.title('Google')
      .assert.visible('input[type=text]')
      .end();
  }
};
```



Arrange, action and assert

```
module.exports = {
  'Google Webtechcon' : function (client) {
    client
      .url('http://www.google.com')
      .waitForElementVisible('body', 1000)
      .assert.title('Google')
      .assert.visible('input[type=text]')
      .setValue('input[type=text]', 'webtechcon')
      .waitForElementVisible('button[name=btnG]', 1000)
      .click('button[name=btnG]')
      .pause(1000)
      .assert.containsText('#rso > div.g > div > div > h3 > a',
        'WebTech Conference 2016')
      .end();
  }
```



Hardcoded values?

Hardcoded values

```
module.exports = {
  'Google Webtechcon' : function (client) {
    client
      .url('http://www.google.com')
      .waitForElementVisible('body', 1000)
      .assert.title('Google')
      .assert.visible('input[type=text]')
      .end();
  }
};
```

Using globals

```
module.exports = {
  'Google Webtechcon' : function (client) {
    var globals = client.globals;

    client
      .url(globals.url)
      .waitForElementVisible('body', 1000)
      .assert.title(globals.static.pageTitle)
      .assert.visible(globals.selectors.searchField)
      .end();
  }
};
```

Define globals

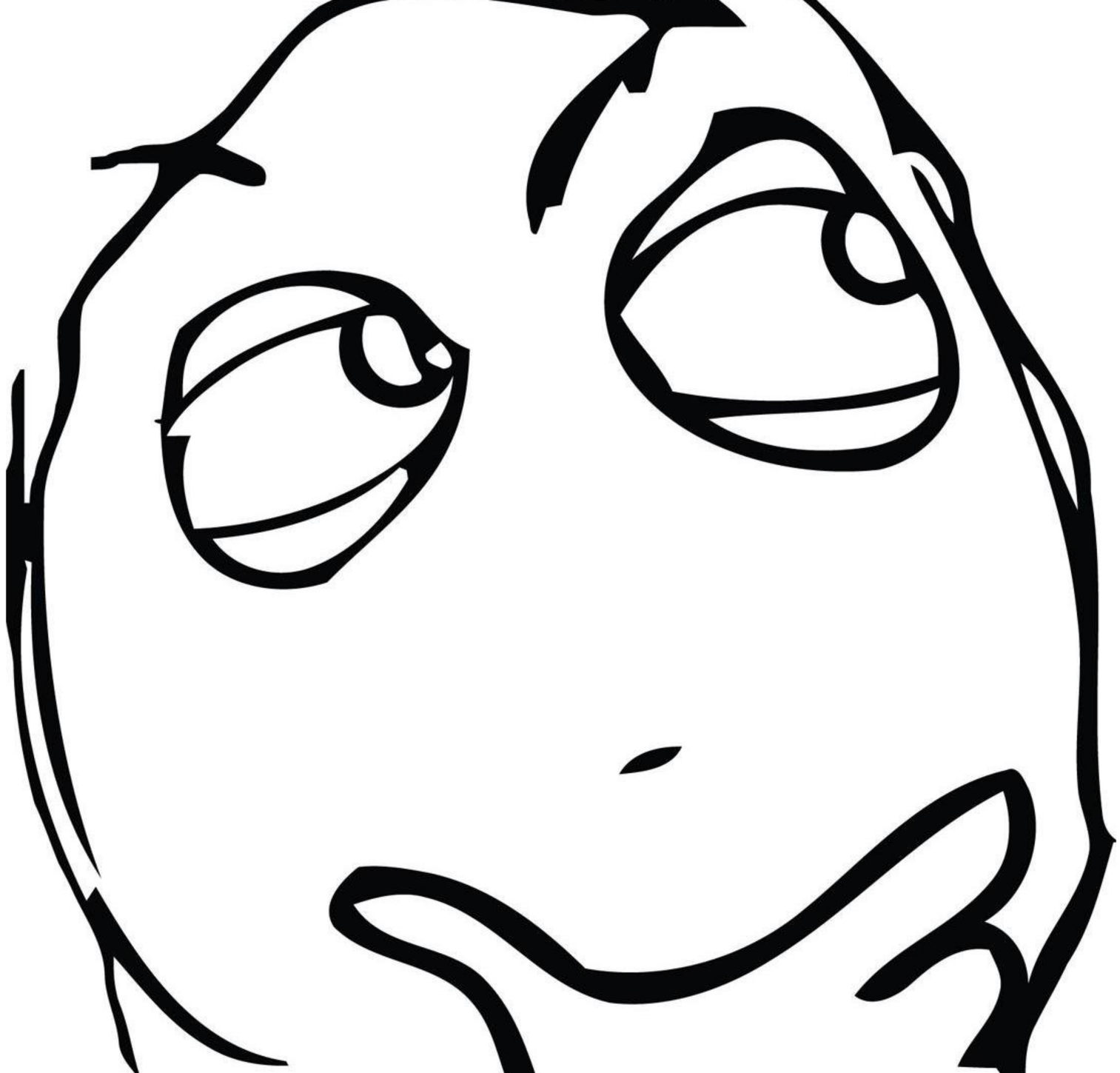
```
module.exports = {
    url: 'http://www.google.com',
    selectors: {
        'searchField': 'input[type=text]'
    },
    static: {
        'pageTitle': 'Google'
    }
}
```

Save as data/google.js

Add to config (test-settings)

```
module.exports = {
    . . .

    "test_settings": {
        "default": {
            "launch_url": "http://localhost",
            "selenium_port": 4444,
            "selenium_host": "localhost",
            "silent": true,
            "screenshots": {
                "enabled": false,
                "path": ""
            },
            "desiredCapabilities": {
                "browserName": "chrome",
                "marionette": true
            },
            "globals": require('./data/google')
        },
    . . .
```



Use Page Objects!

Using Page Objects

```
module.exports = {
    'url': 'http://www.google.com',

    'elements': {
        'searchField': 'input[type=text]'
    }
};
```

Save as object/google.js

Add objects to config

```
module.exports = {
    "src_folders": ["tests"],
    "output_folder": "reports",
    "custom_commands_path": "",
    "custom_assertions_path": "",
    "page_objects_path": "./objects/",
    "globals_path": "",

    "selenium": {. . .},

    "test_settings": {. . .}
};
```

Add objects to config

```
module.exports = {
  'Google Webtechcon' : function (client) {
    var googlePage = client.page.google(),
        globals = client.globals;

    googlePage.navigate()
      .waitForElementVisible('body', 1000)
      .assert.title(globals.static.pageTitle)
      .assert.visible('@searchField')
      .end();
  }
};
```

More PageObjects awesomeness

```
module.exports = {
    'url': 'http://www.some.url',
    'elements': {
        'passwordField': 'input[type=text]',
        'usernameField': 'input[type=password]',
        'submit': 'input[type=submit]'
    },
    commands: [{
        signInAsAdmin: function() {
            return this.setValue('@usernameField', 'admin')
                .setValue('@passwordField', 'password')
                .click('@submit');
        }
    }]
};
```

In your test

```
module.exports = {
   'Admin log in' : function (browser) {
      var login = browser.page.admin.login();

      login.navigate()
           .signInAsAdmin();

      browser.end();
   }
};
```

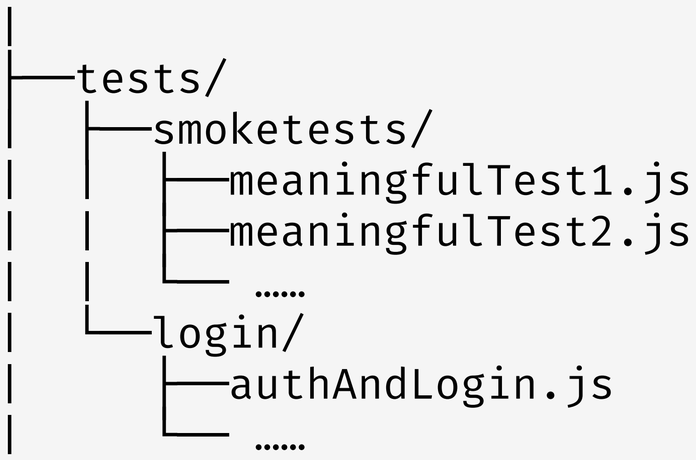
Grouping tests

```
$  nightwatch --group smoketests

$  nightwatch --skipgroup smoketests

$  nightwatch --skipgroup login,whatever
```

Test groups


tests/
smoketests/
meaningfulTest1.js
meaningfulTest2.js
……
login/
authAndLogin.js
……

Tag your tests

```
$ nightwatch --tag login

$ nightwatch --tag login --tag something_else

$ nightwatch --skiptags login

$ nightwatch --skiptags login,something_else
```

In your test

```
module.exports = {
  '@tags': ['login', 'sanity'],
  'Admin log in' : function (browser) {
    var login = browser.page.admin.login();

    login.navigate()
         .signInAsAdmin();

    browser.end();
  }
};
```

Demo

Nightwatch is extendable!

- Write custom commands
- add custom assertions
- write custom reporter

Nightwatch is Javascript!

- Using Filereaders
  - CSV, Excel etc.
- write helpers if needed!

Nightwatch @Mediaman

The problem

Before Nightwatch

The solution

Requirements

Preliminary design ✓

Detailed design ✓

Implementation ✓

UAT

✓ System Test

✓ Integration Test

✓ Unit Test

Thanks!

Carsten Sandtner

@casarock

sandtner@mediaman.de

https://github.com/casarock