

GANC

A web platform for the next century

LAMP

Linux

Apache

mySQL

Perl/PHP/Python/(p)Ruby

MEAN

Mongo

Express.js

AngularJS

Node.js

VLAD

Vagrant

Linux

Ansible

Drupal

Gherkin
AngularJS
Node.JS
COBOL

LAMP

VLAD

MEAN

Gherkin

- Explain test driven development
- Show how unit tests work (using some Perl code)
- Compare unit tests (TDD) with integration tests (BDD)
- Introduce Gherkin & Cucumber
- Show how it works using Cucumber for Java and Maven
- Show how it can be used for acceptance testing a web page developed (in a tiny bit of AngularJS)

Test Driven Development

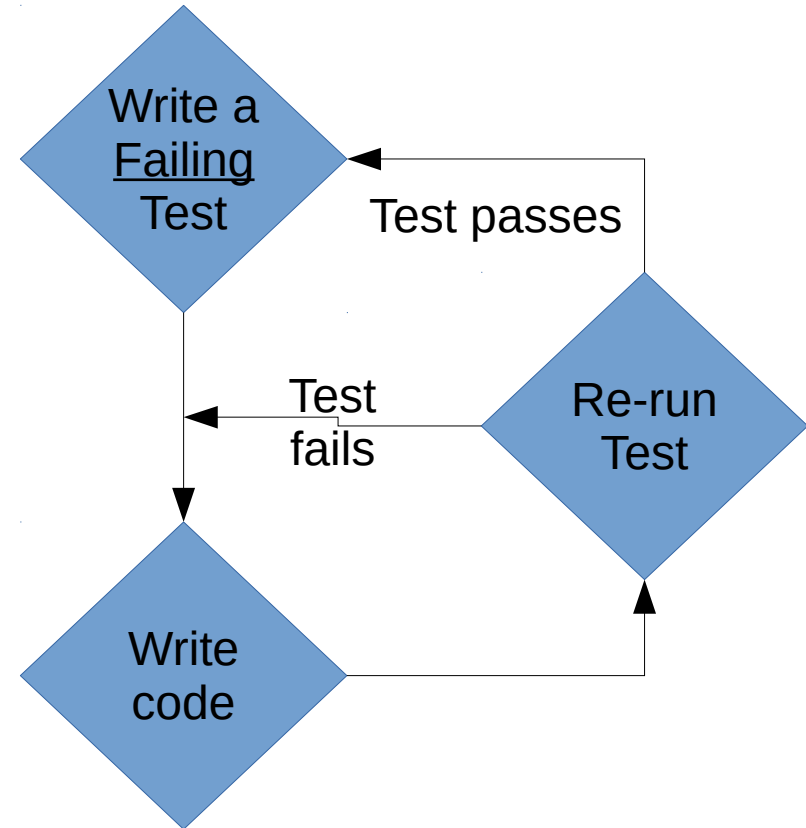
Given I click the button
Then I see the message "Hello"

```
button.onclick=function() {  
    message.value("Hola")  
}
```

Given I click the button
Then I see the message "Hello"

```
button.onclick=function() {  
    message.value("Hello")  
}
```

Given I click the button
Then I see the message "Hello"



BDD vs TDD

- aka Integration Tests
 - Outside In
 - Domain-driven design
 - Leaning towards the organisational case
 - Written in Gherkin, Jspec and others
 - Used by business analysts and testers
- aka Unit Tests
 - Inside Out
 - Functionality-driven design
 - Leaning towards the implementation case
 - Written using jUnit, PHPUnit, Test::More (Perl)
 - Used by software developers

Gherkin example

Feature: Provide a form that users can fill in with their details to capture their interest in coming to HacktionLab

Scenario: User can enter their details into a form

Given I enter my Name

And I enter my Email Address

When I submit my details

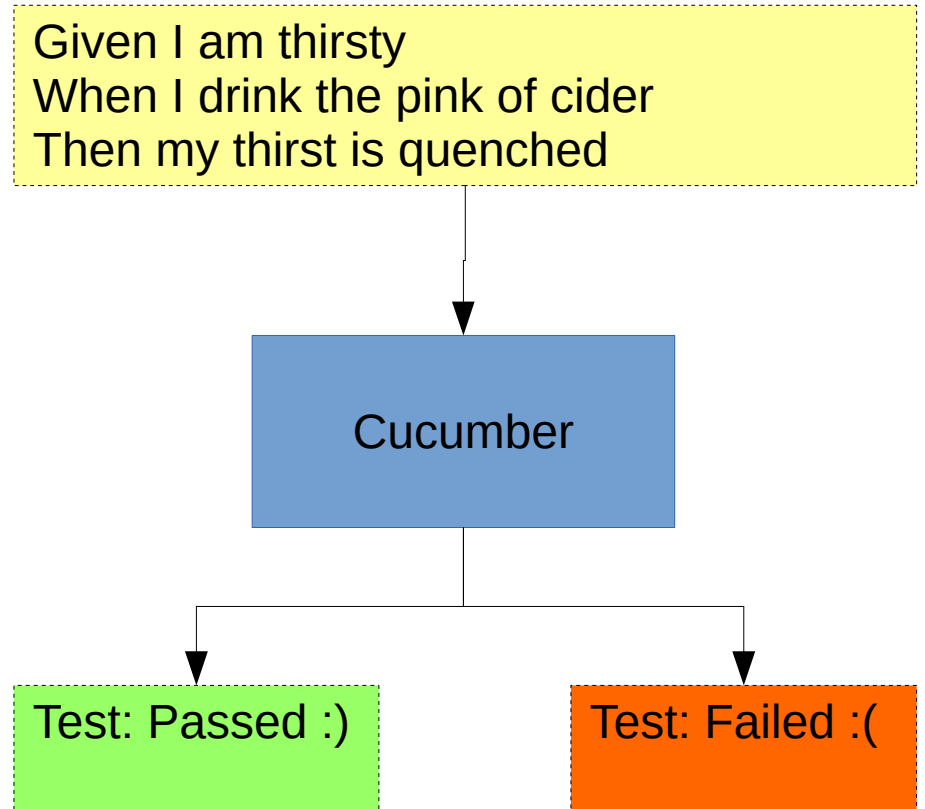
Then I see a thank you message

And ...

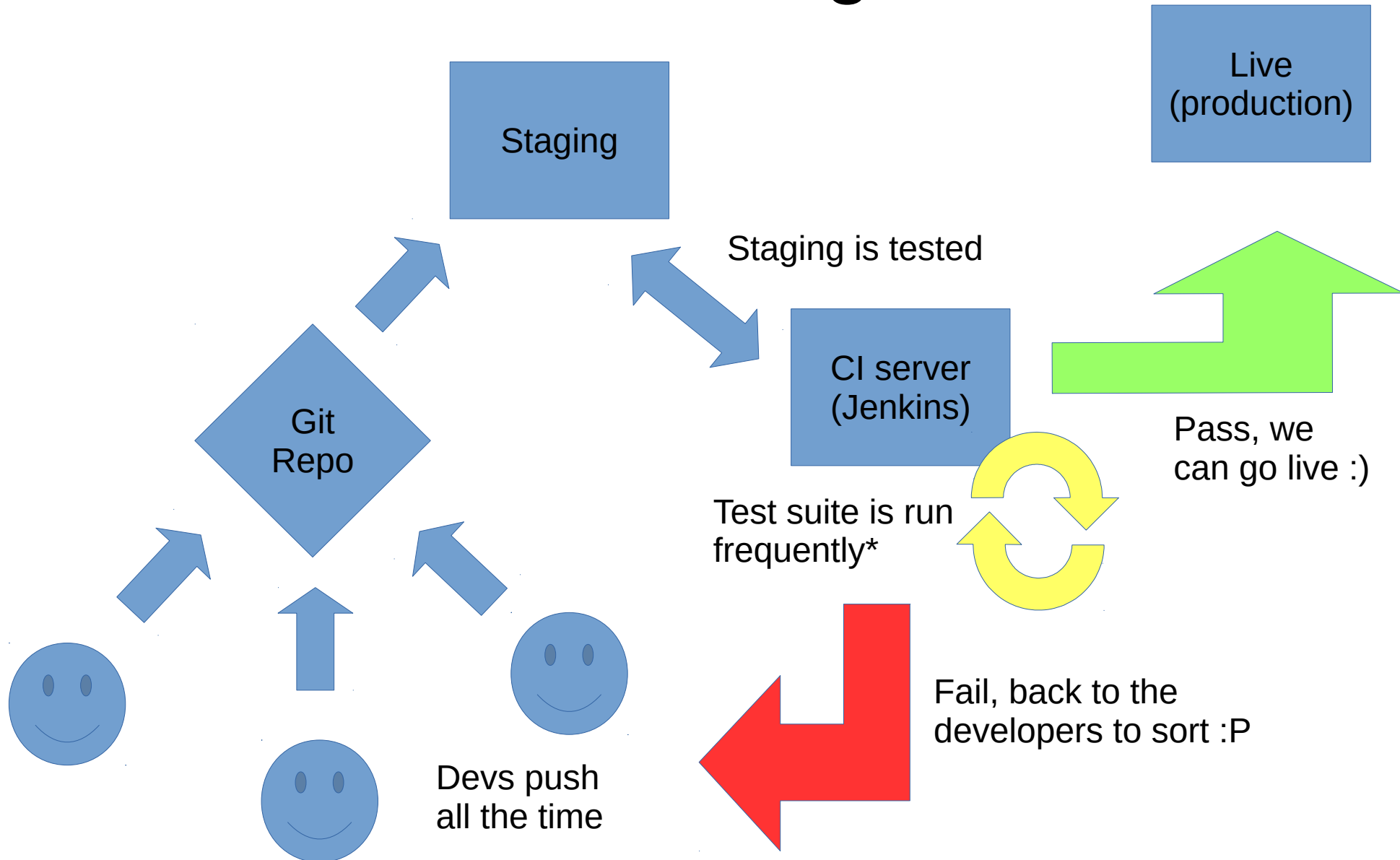
But ...

BDD Tools

- Cucumber (Ruby)
Cucumber for Java
Cucumber-JS
- Behat (PHP/Drupal)
- Behave (Python)
- SpecFlow (.NET)
- Jasmine (JS)
- Spock (Groovy)
- Yadda
(Gherkin for Jasmine)
- Maven, Karma (test runners)
- Jenkins (continuous integration server)



Continuous integration



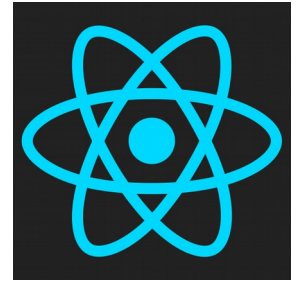
AngularJS

- Introduce
- Show some examples
- Explain Pros and Cons
- Working example linking in to Cucumber
- Compare w/ React.js and others

- Appeared in 2009
- Written and owned by Google



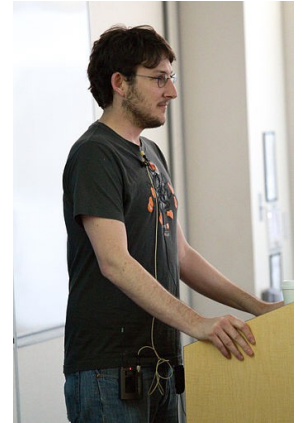
Angular vs React.js



AngularJS

React.js

- Google in 2009
 - Larger framework
 - Implements the model, view & controller
 - Part of the MEAN stack (Mongo, Express, Angular, Node)
 - Two-way data binding, model can be directly managed by AngularJS
 - JavaScript and HTML are separated in distinct files
 - Steep learning curve
 - Requires advanced-level structured JavaScript programming knowledge
 - Version 2 introduces radical changes
- Facebook in 2013, also maintained by Instagram
 - Smaller framework
 - Implements the view of MVC; Orientated towards creating and manipulating DOM components.
 - One-way data flow
 - Written in JSX files that in-line HTML in JavaScript
 - Not such a steep learning curve.
 - Requires an intermediate-level of JavaScript programming knowledge.
 - No major backwardsly-incompatible update yet released



- Appeared in 2009
- Created by Ryan Dahl
- Authors are Node.js Foundation developers, Joyent and GitHub
- GPL and managed under Node.js Foundation and Linux Foundation
- Provides a JavaScript command line interpreter
- Used primarily for writing server-side applications in JavaScript, but has other uses..
- Such as npm (node package manager), which is a very useful tool for installing JavaScript software, such as AngularJS itself.
- Supported in lots of IDEs, such as Atom, Eclipse, Sublime Text, IntelliJ, WebStorm and Visual Studio
- io.js was a fork, but has now merged back in to node.js to become nodejs
- Express.js is a web application framework for Node.js and the “E” in the MEAN stack

Node.js example

```
var koa = require('koa');
var mount = require('koa-mount');
var router = require('koa-route');
var bodyParser = require('koa-bodyparser');
var db = require('./db');

var app = module.exports = koa();

app.use(mount('/', db));

app.use(bodyParser());

app.use(router.get('/basketItems', basketItems));
app.use(router.post('/:reorder', reorderItems));

function * basketItems() {
  var quoteId = '123812';
  var dbResult = yield this.db.query(
    'CALL dale_live_db.getBasketItems(?)',
    [quoteId]
  );
  this.body = dbResult[0][0];
}

function * reorderItems() {
  var itemA = this.request.body.itemA;
  var itemB = this.request.body.itemB;

  if (itemA && itemB) {
    yield this.db.query(
      'CALL dale_live_db.swapBasketItemReferenceIds(?, ?)',
      [itemA, itemB]
    );
    this.status = 200;
  } else {
    this.status = 204;
  }
}
```

Node.js?

- Tenuous link part of talk
- <https://github.com/IonicaBizau/node-cobol>