



@somkiat

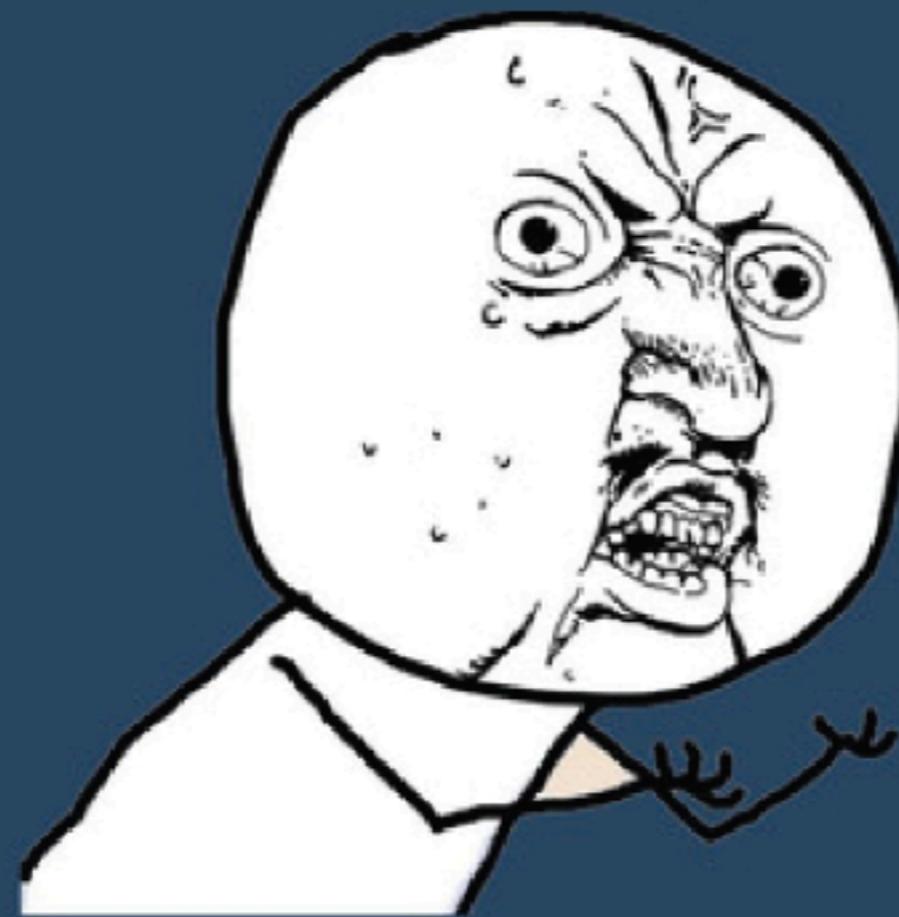


# Test anti-patterns





# WHY YOU COPY AND PASTE ?!



## STOP PLAGIARISM

memegenerator.net



# Use Version Control ?







TETBK  
EX.





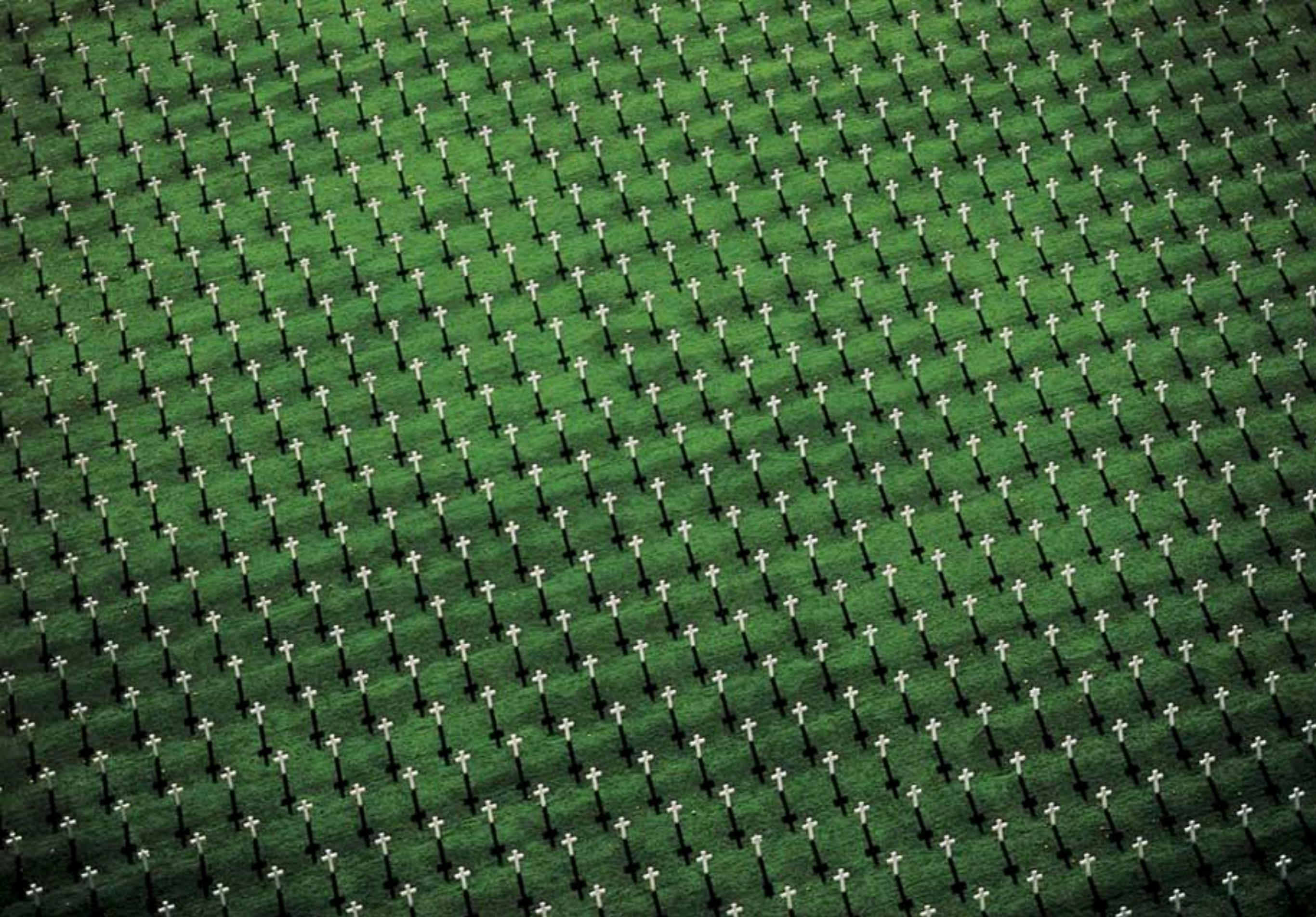
© BRIAN CARLIN





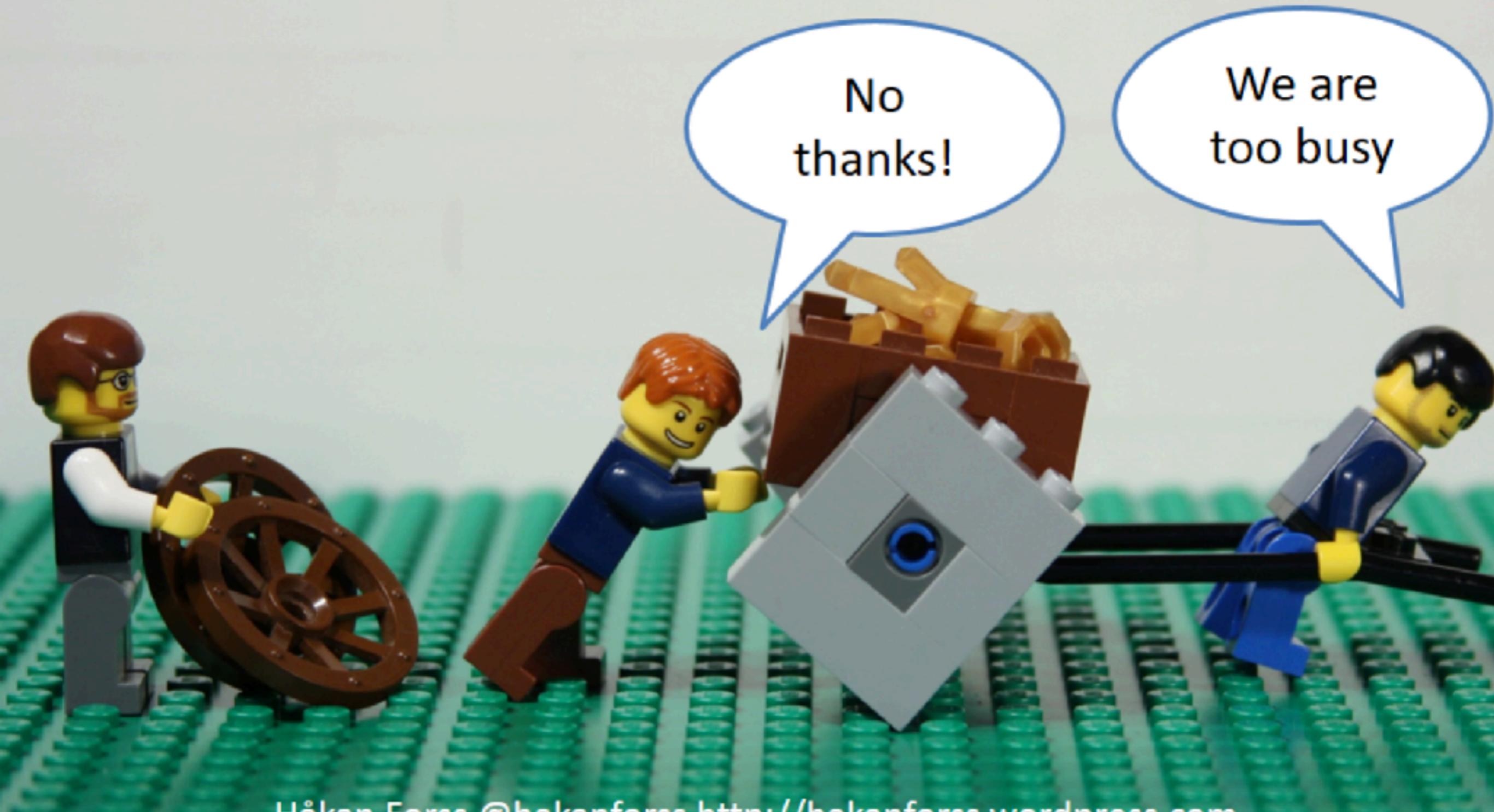






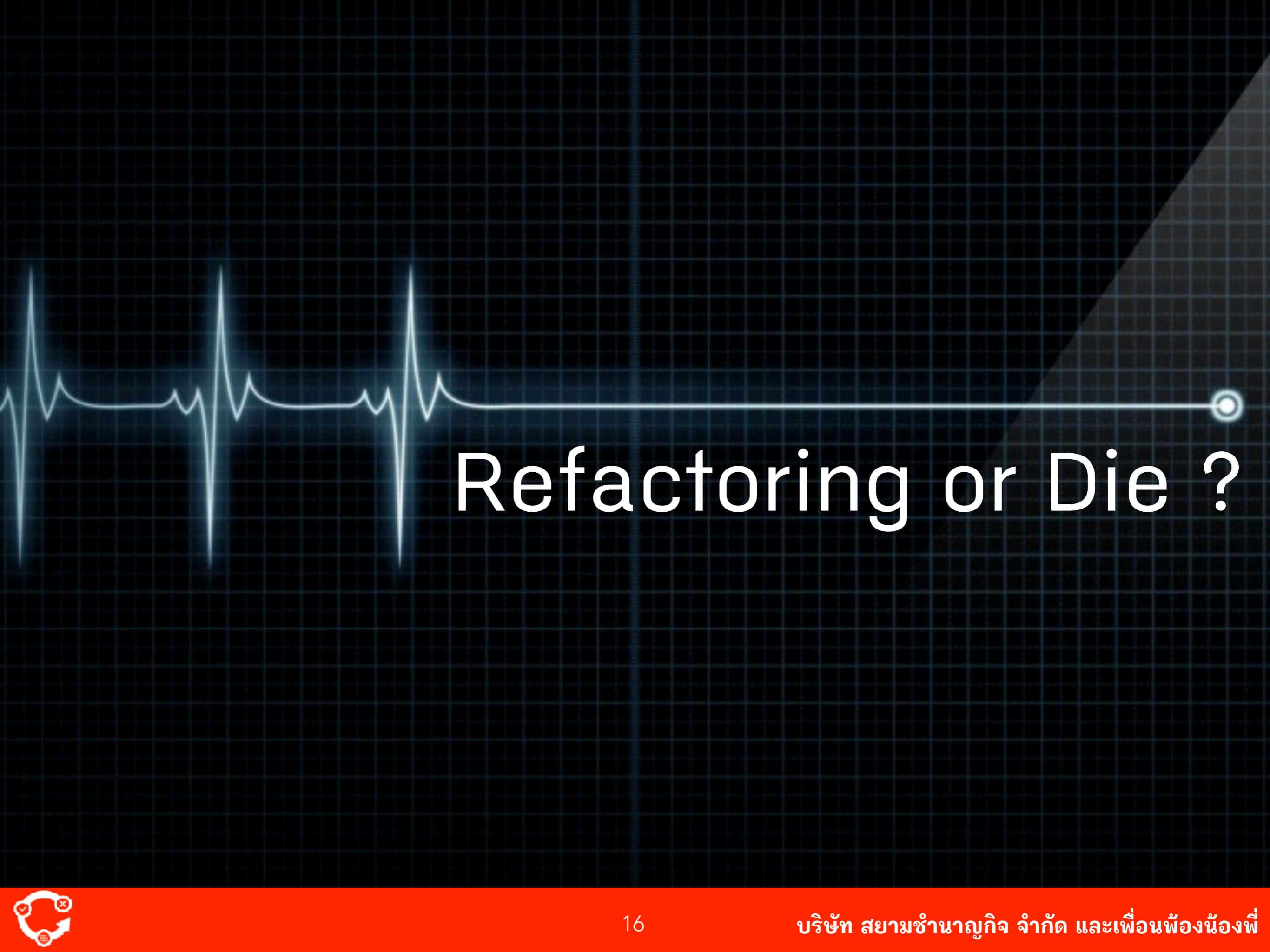


# Are you too busy to improve?



Håkan Forss @hakanforss <http://hakanforss.wordpress.com>





# Refactoring or Die ?

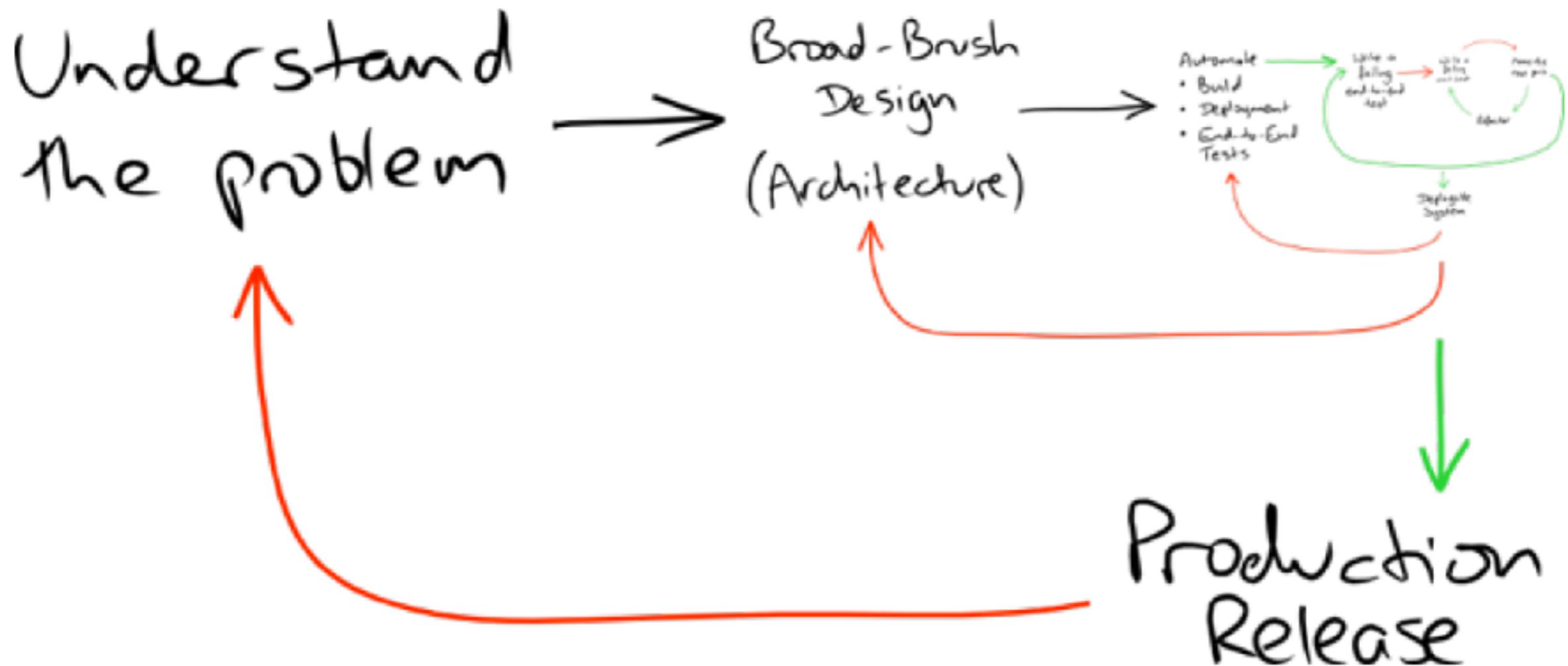


# Refactoring tests

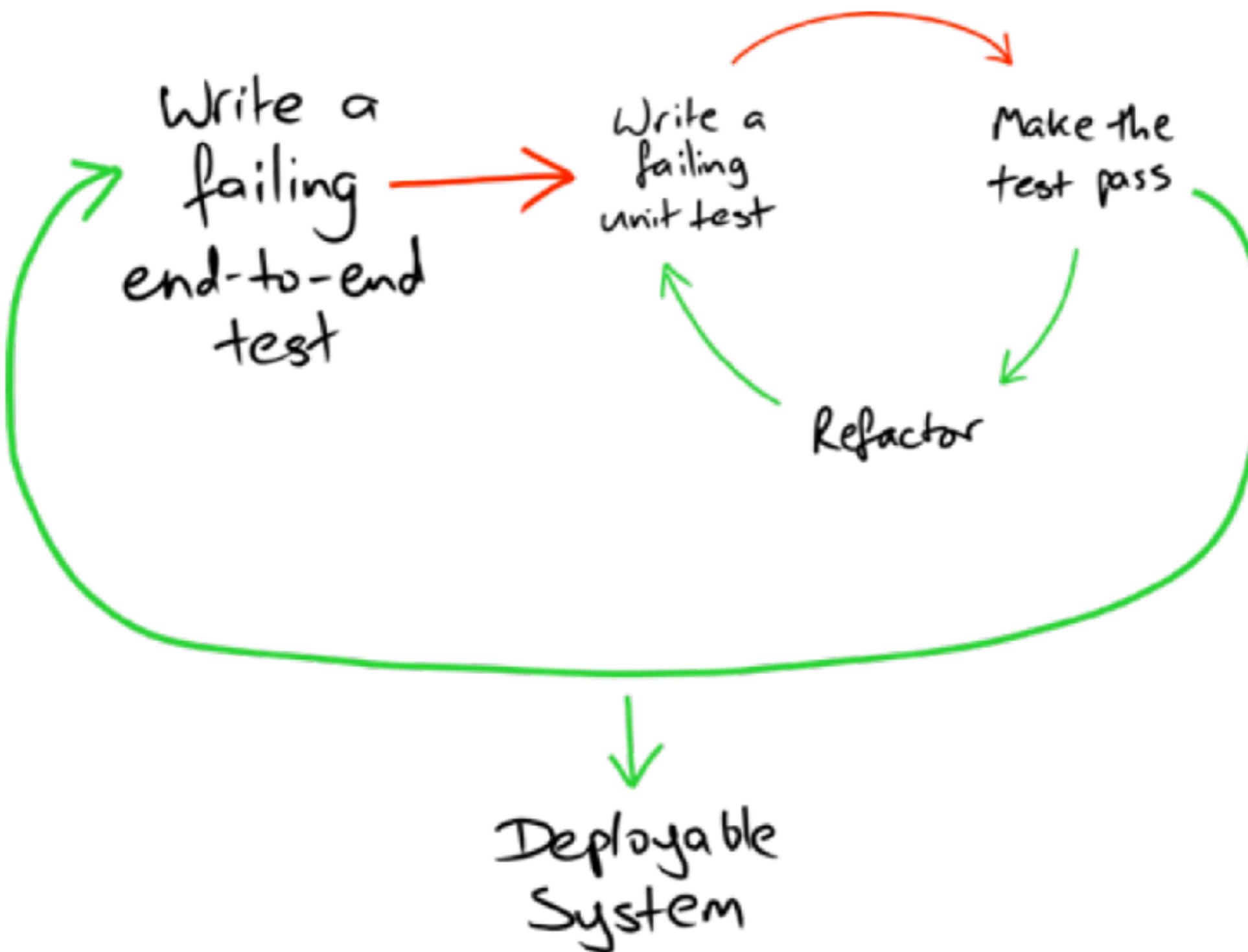
“The act of restructuring your code to improve the internal efficiency, stability and long-term maintainability without add/modify any of the functionality.”



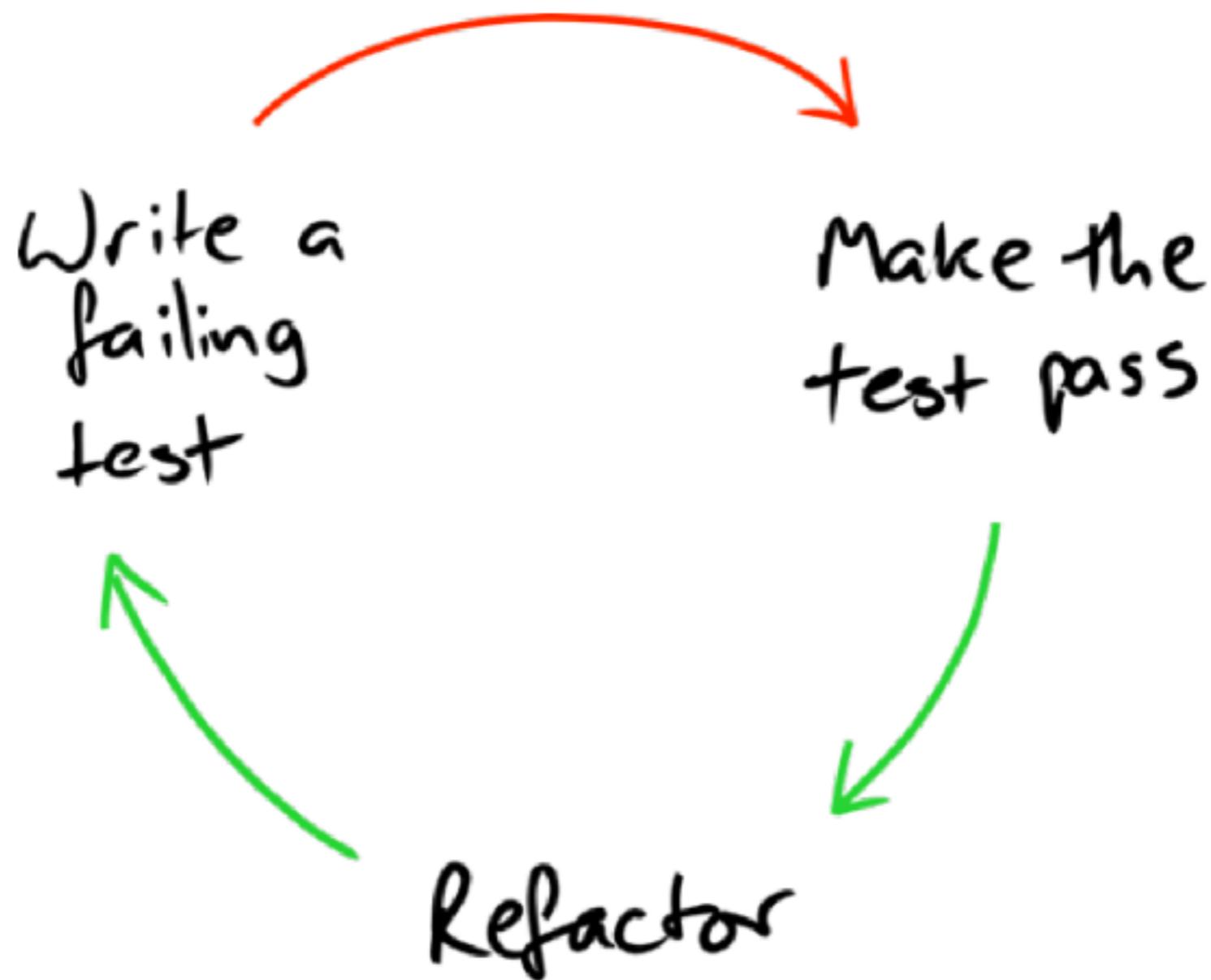
# Large feedback loop



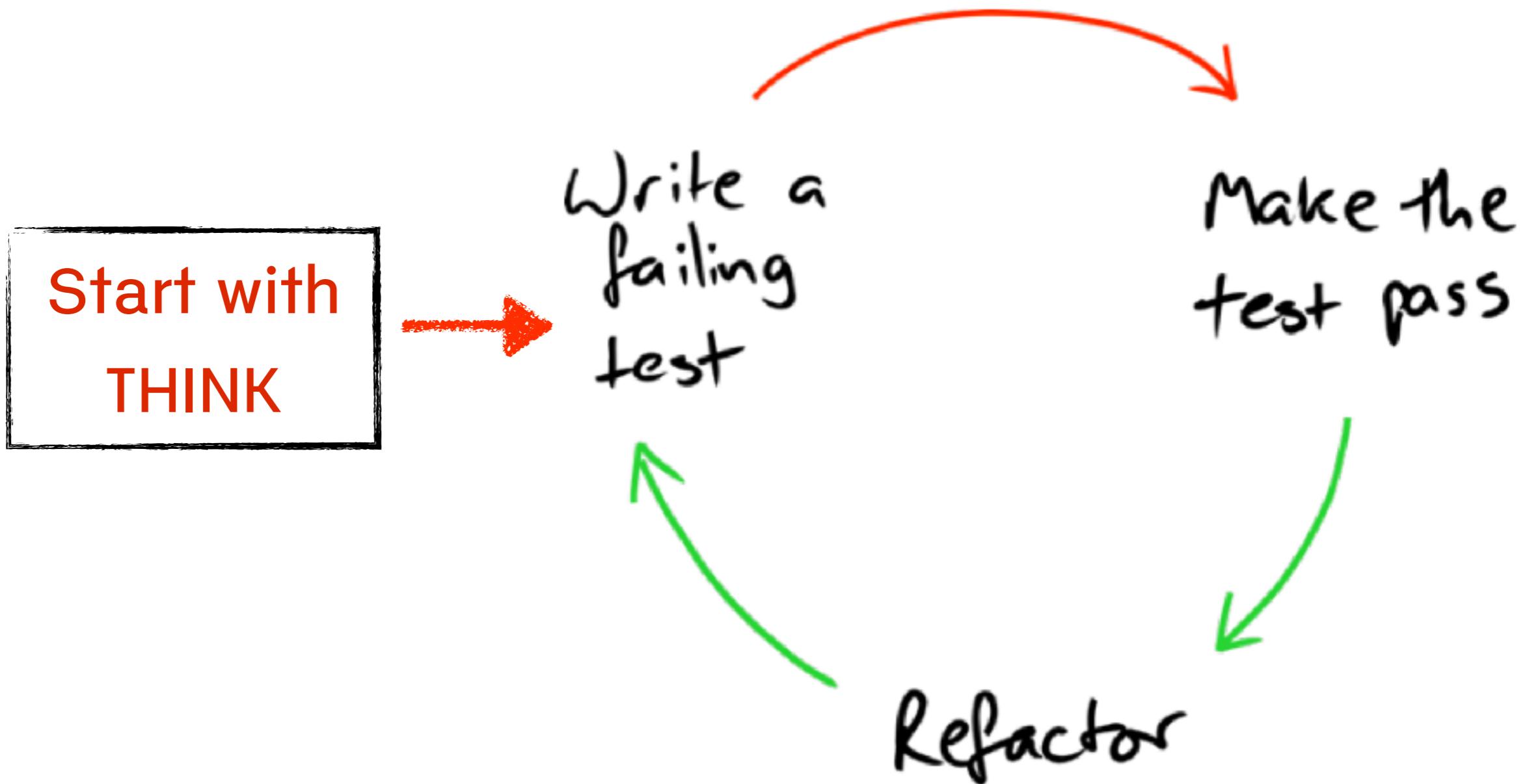
# Acceptance Test Driven



# TDD cycle



# Improve TDD cycle



# Life cycle of test framework



# Test frameworks

MS Test

NUnit



# Workshop



# Workshop

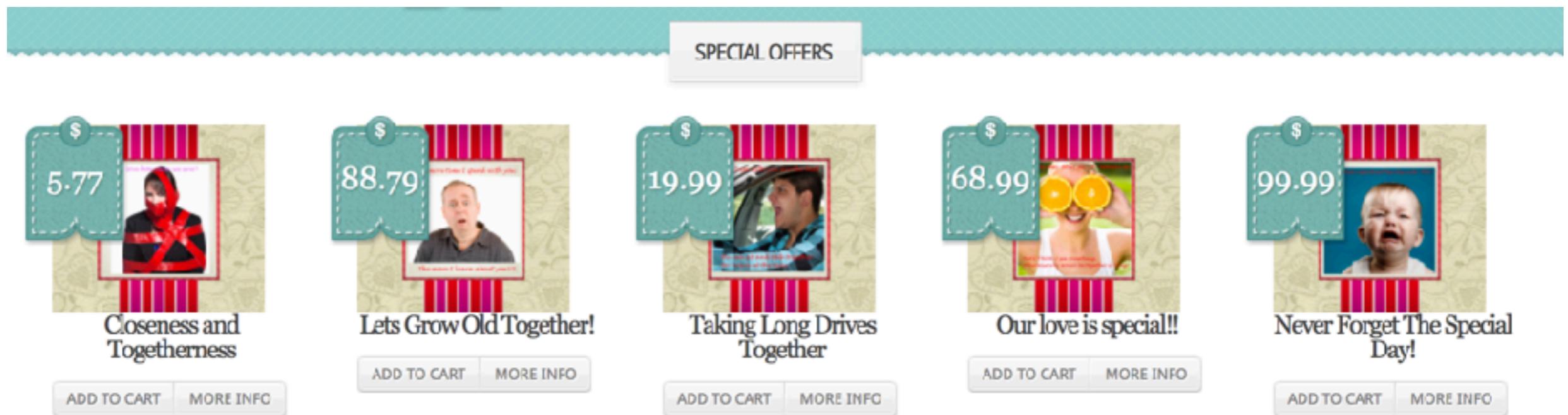
Comment product (Success and Fail)

<http://awful-valentine.com/>



# Flow of comment product

## 1. Choose a product



# Flow of comment product

## 2. Show product detail



# Flow of comment product

## 3. Submit comment

Your email address will not be published.

somkiat

somkiat.p@gmail.com

somkiat.cc

Rating 

แม่ทัพภาคการ

You may use these HTML tags and attributes: <a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code> <del datetime=""> <em> <i> <q cite=""> <strike> <strong>

SUBMIT COMMENT



# Flow of comment product

## 4. Read latest comment

① [awful-valentine.com/lets-grow-old-together/#comment-3497](http://awful-valentine.com/lets-grow-old-together/#comment-3497)

CONSUMER REVIEWS

 somkiat  
*December 9, 2016 at 2:46 pm*

แจ่มมากนาย

★★★☆☆



# Let's start coding



# Let's start to review



```

[Test()]
public void add_review_to_product()
{
    IWebDriver webdriver = new ChromeDriver(@"~/Users/somkiat/Projects/demo_ui_test/demo_ui_test/");
    webdriver.Url = "http://awful-valentine.com/";
    webdriver.FindElement(By.XPath("//*[@id='special-items']/div[4]/div[1]/a[2]")).Click();
    Assert.AreEqual("http://awful-valentine.com/our-love-is-special/", webdriver.Url);
    Assert.AreEqual("Our love is special!!", webdriver.FindElement(By.ClassName("category-title")).Text);

    webdriver.FindElement(By.Id("author")).SendKeys("Somkiat");
    webdriver.FindElement(By.Id("email")).SendKeys("Somkiat@gmail.com");
    webdriver.FindElement(By.Id("url")).SendKeys("http://www.somkiat.cc");
    webdriver.FindElement(By.XPath("//*[@id='et-rating']/div/span/div[8]/a")).Click();
    webdriver.FindElement(By.Id("comment")).Clear();
    webdriver.FindElement(By.Id("comment")).SendKeys("My comment naja 3");
    webdriver.FindElement(By.Id("submit")).Click();

    if (webdriver.Url.Contains("#"))
    {
        string[] paths = webdriver.Url.Split('#');
        string commentId = paths[1];
        IWebElement commentElement = webdriver.FindElement(By.Id(commentId));
        IWebElement infoElement = commentElement.FindElement(By.ClassName("comment-author-metainfo"));
        string name = infoElement.FindElement(By.ClassName("url")).Text;
        string commentContent = commentElement.FindElement(By.ClassName("comment-content")).Text;

        Assert.AreEqual("Somkiat", name);
        Assert.AreEqual("My comment naja 3", commentContent);
    }
}

```



```
[Test()]
public void add_review_to_product()
{
    IWebDriver webdriver = new ChromeDriver(@"~/Users/somkiat/Projects/demo_ui_test/demo_ui_test/");
    webdriver.Url = "http://awful-valentine.com/";
    webdriver.FindElement(By.XPath("//*[@id='special-items']/div[4]/div[1]/a[2]")).Click();
    Assert.AreEqual("http://awful-valentine.com/our-love-is-special/", webdriver.Url);
    Assert.AreEqual("Our love is special!!", webdriver.FindElement(By.ClassName("category-title")).Text);

    webdriver.FindElement(By.Id("author")).SendKeys("Somkiat");
    webdriver.FindElement(By.Id("email")).SendKeys("Somkiat@gmail.com");
    webdriver.FindElement(By.Id("url")).SendKeys("http://www.somkiat.cc");
    webdriver.FindElement(By.XPath("//*[@id='et-rating']/div/span/div[8]/a")).Click();
    webdriver.FindElement(By.Id("comment")).Clear();
    webdriver.FindElement(By.Id("comment")).SendKeys("My comment naja 3");
    webdriver.FindElement(By.Id("submit")).Click();

    if (webdriver.Url.Contains("#"))
    {
        string[] paths = webdriver.Url.Split('#');
        string commentId = paths[1];
        IWebElement commentElement = webdriver.FindElement(By.Id(commentId));
        IWebElement infoElement = commentElement.FindElement(By.ClassName("comment-author-metainfo"));
        string name = infoElement.FindElement(By.ClassName("url")).Text;
        string commentContent = commentElement.FindElement(By.ClassName("comment-content")).Text;

        Assert.AreEqual("Somkiat", name);
        Assert.AreEqual("My comment naja 3", commentContent);
    }
}
```

TM



# Refactoring tests

**DRY (Don't Repeat Yourself)**

Test independence

Hermetic test pattern

Sharing common functionalities between tests

Random run order practice



# DRY

**Reduce long-term maintenance costs  
by removing all unnecessary duplication**

**Need Single Point Of Truth (SPOT)**



# DRY's Pros

Modular tests

Reduced duplication

Fast updates

No junk code



# DRY's Cons

Complicated project structure

Constant improve

Programming skills



# Let's start to refactoring



# 1. Use setup and teardown

```
[TestFixture]
public class AddCommentProductDRY
{
    IWebDriver webdriver;

    [SetUp]
    public void setup()
    {
        webdriver = new ChromeDriver();
    }

    [TearDown]
    public void teardown()
    {
        webdriver.Close();
    }
}
```



# Test life cycle

Test Suite  
Setup

Test Case 1    Setup  
                    Teardown

Test Case 2    Setup  
                    Teardown

Teardown



# MS Test

[TestClass]

[ClassInitialize]

[TestMethod]

[TestInitialize]

[TestCleanup]

Test Case 2

[ClassCleanup]



# NUnit

[TestFixture]

[OneTimeSetup]

[Test]

[Setup]

[TearDown]

Test Case 2

[OneTimeTeardown]



## 2. Remove duplication with method

```
private void chooseProduct()
{
    webdriver.FindElement(By.XPath("//*[@id=\"special-items\"]//li"));
}

[Test()]
public void add_review_to_product()
{
    webdriver.Url = "http://awful-valentine.com/";
    chooseProduct();
    Assert.AreEqual("http://awful-valentine.com/our-love-is-special", webdriver.Title);
    Assert.AreEqual("Our love is special!!", webdriver.FindElement(By.XPath("//p")));
}
```



# 3. Test Goal ?

Remove unnecessary assertions

```
[Test()]
public void add_review_to_product()
{
    webdriver.Url = "http://awful-valentine.com/";
    chooseProduct();

    Assert.AreEqual("http://awful-valentine.com/our-love-is-special/", webdriver.
    Assert.AreEqual("Our love is special!!", webdriver.FindElement(By.ClassName('

```



# 4. Extract to new method

## Fill in comment form

```
private void fillInCommentForm()
{
    webdriver.FindElement(By.Id("author")).SendKeys("Somkiat");
    webdriver.FindElement(By.Id("email")).SendKeys("Somkiat@gmail.co");
    webdriver.FindElement(By.Id("url")).SendKeys("http://www.somkiat");
    webdriver.FindElement(By.XPath("//*[@id='et-rating']/div/span/"));
    webdriver.FindElement(By.Id("comment")).Clear();
    webdriver.FindElement(By.Id("comment")).SendKeys("My comment naj");
    webdriver.FindElement(By.Id("submit")).Click();

}

[Test()]
public void add_review_to_product()
{
    webdriver.Url = "http://awful-valentine.com/";
    chooseProduct();
    fillInCommentForm();
```



# 5. Remove test-on-test dependence

```
private void fillInCommentForm()
{
    webdriver.FindElement(By.Id("author")).SendKeys("Somkiat");
    webdriver.FindElement(By.Id("email")).SendKeys("Somkiat@gmail.com");
    webdriver.FindElement(By.Id("url")).SendKeys("http://www.somkiat.cc");
    webdriver.FindElement(By.XPath("//*[@id='et-rating']/div/span/div[8]/a"));
    webdriver.FindElement(By.Id("comment")).Clear();
    webdriver.FindElement(By.Id("comment")).SendKeys("My comment naja 3");
    webdriver.FindElement(By.Id("submit")).Click();
}
```



# 5. Remove test-on-test dependence

Pass parameters to method

```
private void fillInCommentForm(Comment comment)
{
    webdriver.FindElement(By.Id("author")).SendKeys(comment.name);
    webdriver.FindElement(By.Id("email")).SendKeys(comment.email);
    webdriver.FindElement(By.Id("url")).SendKeys(comment.link);
    webdriver.FindElement(By.XPath("//*[@id='et-rating']/div/span/div"));
    webdriver.FindElement(By.Id("comment")).Clear();
    webdriver.FindElement(By.Id("comment")).SendKeys(comment.content);
    webdriver.FindElement(By.Id("submit")).Click();
}
```

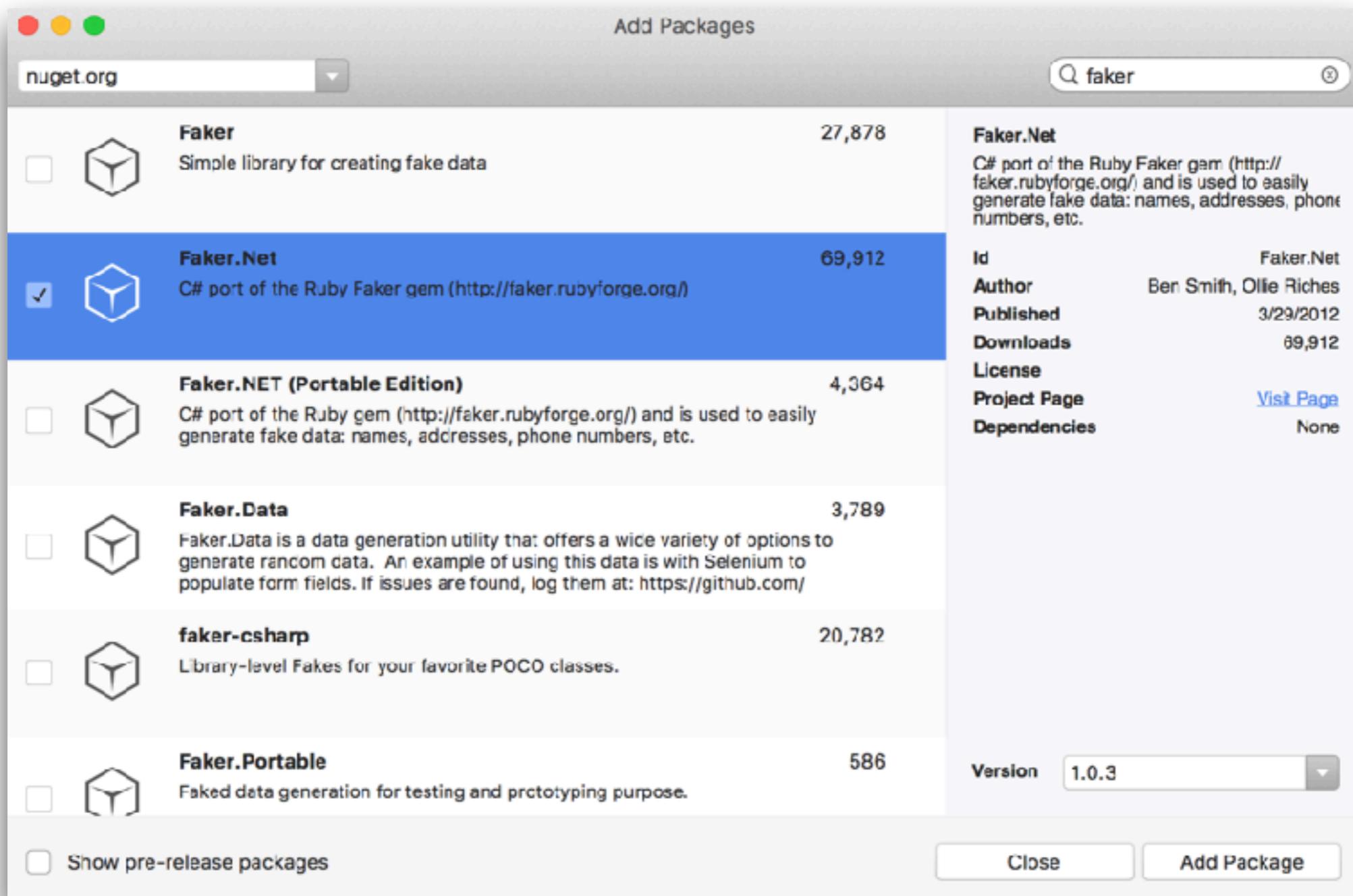


# 6. Make unique data with time

```
[Test()]
public void add_review_to_product()
{
    webdriver.Url = "http://awful-valentine.com/";
    chooseProduct();
    Comment comment = new Comment();
    comment.name = "Somkiat";
    comment.email = "somkiat.p@gmail.com";
    comment.link = "http://www.somkiat.cc";
    comment.content = "My comment " + DateTime.Now.ToString("hh.mm.ss.fffff");
    fillInCommentForm(comment);
}
```



# Tips :: Fake data



<https://github.com/slashdotdash/faker-cs>



# Tips :: Fake data

```
private Comment generateUniqueCommentWithFaker()
{
    Comment comment = new Comment();
    comment.name = Faker.Name.FullName();
    comment.email = Faker.Internet.Email();
    comment.link = Faker.Internet.DomainName();
    comment.content = Faker.Lorem.Sentence() + DataGenerator.generateRandomString();
    return comment;
}
```



# 7. Extract to new method

## for generate unique comment

```
private Comment generateUniqueComment()
{
    Comment comment = new Comment();
    ...
    comment.name = "Somkiat";
    comment.email = "somkiat.p@gmail.com";
    comment.link = "http://www.somkiat.cc";
    comment.content = "My comment " + DateTime.Now.ToString("hh.mm.ss.ffffff");
    return comment;
}

[Test()]
public void add_review_to_product()
{
    webdriver.Url = "http://awful-valentine.com/";
    chooseProduct();
    Comment comment = generateUniqueComment();
    ...
    fillInCommentForm(comment);
}
```



# 7. Extract to new method

## for get new comment id

```
private string getNewCommentId()
{
    string[] paths = webdriver.Url.Split('#');
    string commentId = paths[1];
    return commentId;
}

[Test()]
public void add_review_to_product()
{
    webdriver.Url = "http://awful-valentine.com/";
    chooseProduct();
    Comment comment = generateUniqueComment();
    fillInCommentForm(comment);
    string newCommentId = getNewCommentId();
```



# What's next ?

```
[Test()]
public void add_review_to_product()
{
    webdriver.Url = "http://awful-valentine.com/";
    chooseProduct();
    Comment comment = generateUniqueComment();
    fillInCommentForm(comment);
    string newCommentId = getNewCommentId();

    IWebElement commentElement = webdriver.FindElement(By.Id(newCommentId));
    IWebElement infoElement = commentElement.FindElement(By.ClassName("comment-au
    string name = infoElement.FindElement(By.ClassName("url")).Text;
    string commentContent = commentElement.FindElement(By.ClassName("comment-cont
    Assert.AreEqual(comment.name, name);
    Assert.AreEqual(comment.content, commentContent);
}
```



# What's next ?

Extract to new method

```
[Test()]
public void add_review_to_product()
{
    webdriver.Url = "http://awful-valentine.com/";
    chooseProduct();
    Comment comment = generateUniqueComment();
    fillInCommentForm(comment);
    string newCommentId = getNewCommentId();

    IWebElement commentElement = webdriver.FindElement(By.Id(newCommentId));
    IWebElement infoElement = commentElement.FindElement(By.ClassName("comment-au
    string name = infoElement.FindElement(By.ClassName("url")).Text;
    string commentContent = commentElement.FindElement(By.ClassName("comment-cont
    Assert.AreEqual(comment.name, name);
    Assert.AreEqual(comment.content, commentContent);
}
```



# What's next ?

```
[Test()]
public void add_review_to_product()
{
    gotoHomepage();
    chooseProduct();
    Comment comment = generateUniqueComment();
    fillInCommentForm(comment);
    string newCommentId = getNewCommentId();

    IWebElement commentElement = webdriver.FindElement(By.Id(newCommentId));
    IWebElement infoElement = commentElement.FindElement(By.ClassName("comment"));
    string name = infoElement.FindElement(By.ClassName("url")).Text;
    string commentContent = commentElement.FindElement(By.ClassName("comment-content"));
    Assert.AreEqual(comment.name, name);
    Assert.AreEqual(comment.content, commentContent);
}
```



# What's next ?

```
[Test()]
public void add_review_to_product()
{
    gotoHomepage();
    chooseProduct();
    Comment comment = generateUniqueComment();
    fillInCommentForm(comment);
    string newCommentId = getNewCommentId();

    IWebElement commentElement = webdriver.FindElement(By.Id(newCommentId));
    IWebElement infoElement = commentElement.FindElement(By.ClassName("comment"));
    string name = infoElement.FindElement(By.ClassName("url")).Text;
    string commentContent = commentElement.FindElement(By.ClassName("comment-content"));
    Assert.AreEqual(comment.name, name);
    Assert.AreEqual(comment.content, commentContent);
}
```



# What's next ?

Extract to new method and rearrange !!

```
private void generateNewComment(Comment comment)
{
    gotoHomepage();
    chooseProduct();
    fillInCommentForm(comment);
}

[Test()]
public void add_review_to_product()
{
    Comment comment = generateUniqueComment();
    generateNewComment(comment);
    string newCommentId = getNewCommentId();

    IWebElement commentElement = webdriver.FindElement(By.Id(newCommentId));
    IWebElement infoElement = commentElement.FindElement(By.ClassName("comment-a"));
    string name = infoElement.FindElement(By.ClassName("url")).Text;
    string commentContent = commentElement.FindElement(By.ClassName("comment-con"));
    Assert.AreEqual(comment.name, name);
    Assert.AreEqual(comment.content, commentContent);
}
```



# Improve another test case

```
[Test()]
public void duplicate_review_to_product()
{
    Comment comment = generateUniqueComment();
    generateNewComment(comment);
    generateNewComment(comment);

    IWebElement errorElement = webdriver.FindElement(By.Id("error-page"));
    string error = errorElement.Text;
    Assert.AreEqual("Duplicate comment detected; it looks as though you've

}
```



# Random run order principle

Very importance on **Continuous Integration (CI)**  
environment



# Pros

Prevent test interdependence

Flushes out data pollution

Build-in



# Cons

A lot of refactoring

Random run audits

Team frustration

No build-in support



# Stable Testing



# Culture of stability

Running fast and failing fast

Running as often as possible

Keep a clean and consistent environment

Discard bad code changes

**Maintain a stable test suite**



# Unstable tests !!



# Unstable tests !!

Asynchronous process

AJAX

Latency of response from server

Animation

Slow internet connection

Slow browser

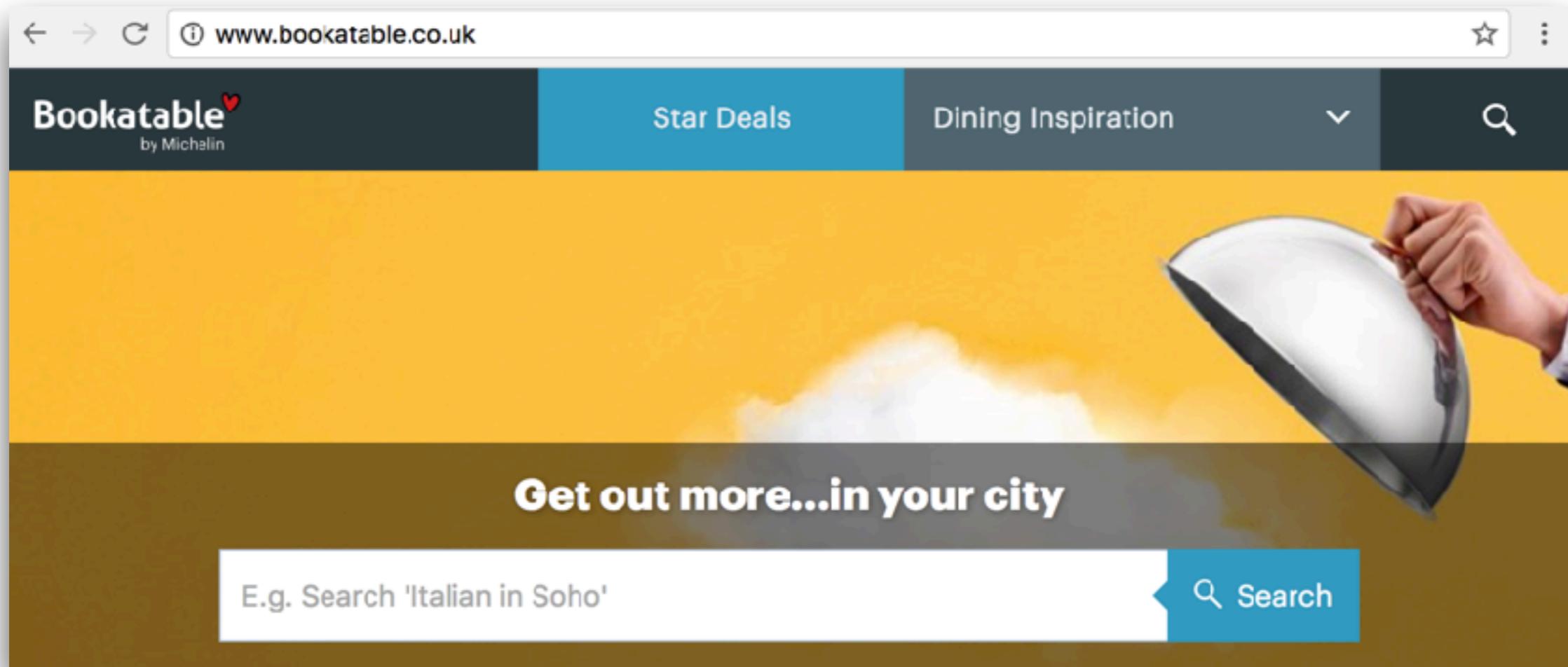


# Workshop



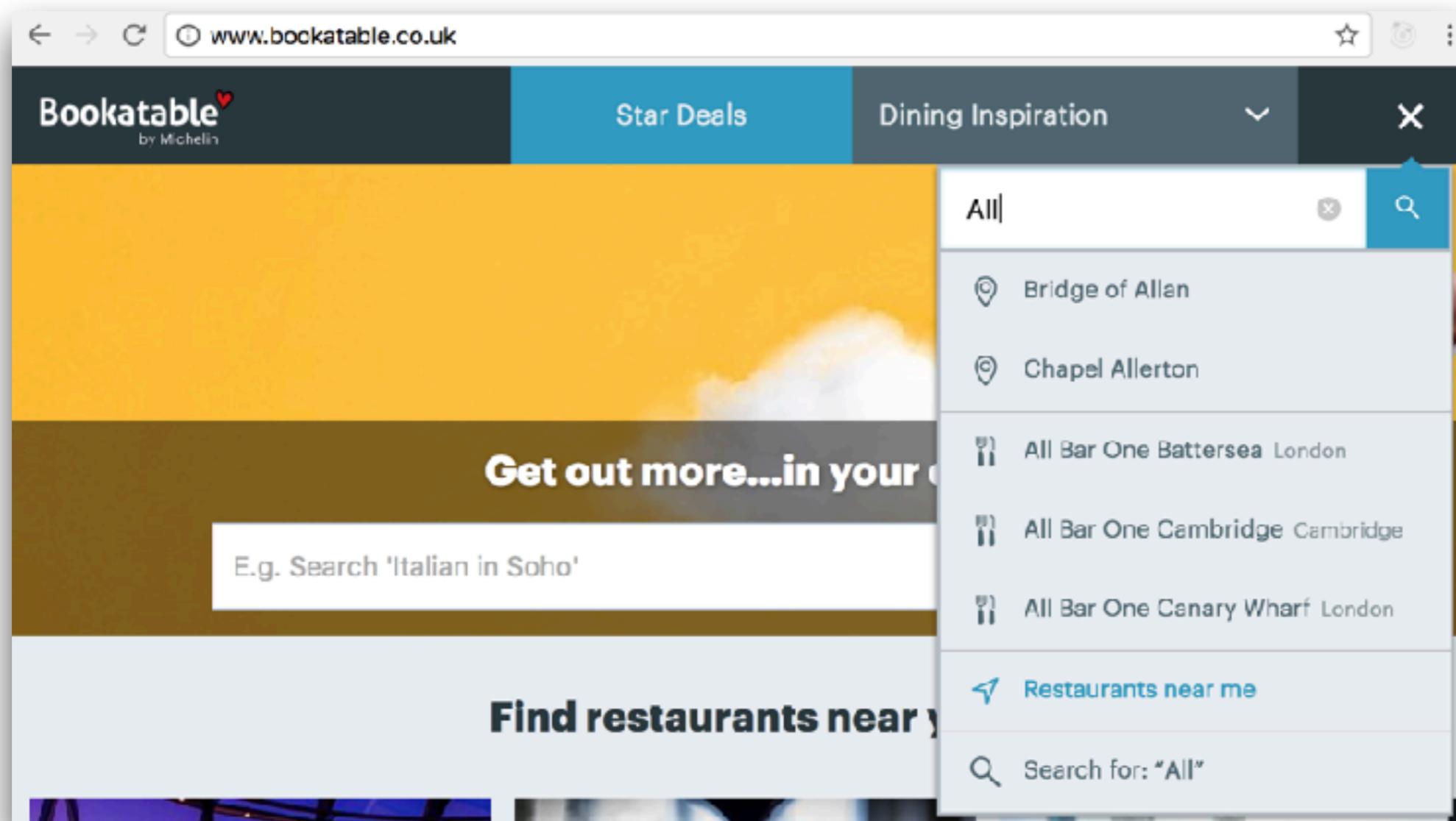
# Asynchronous process

<http://www.bookatable.co.uk/>



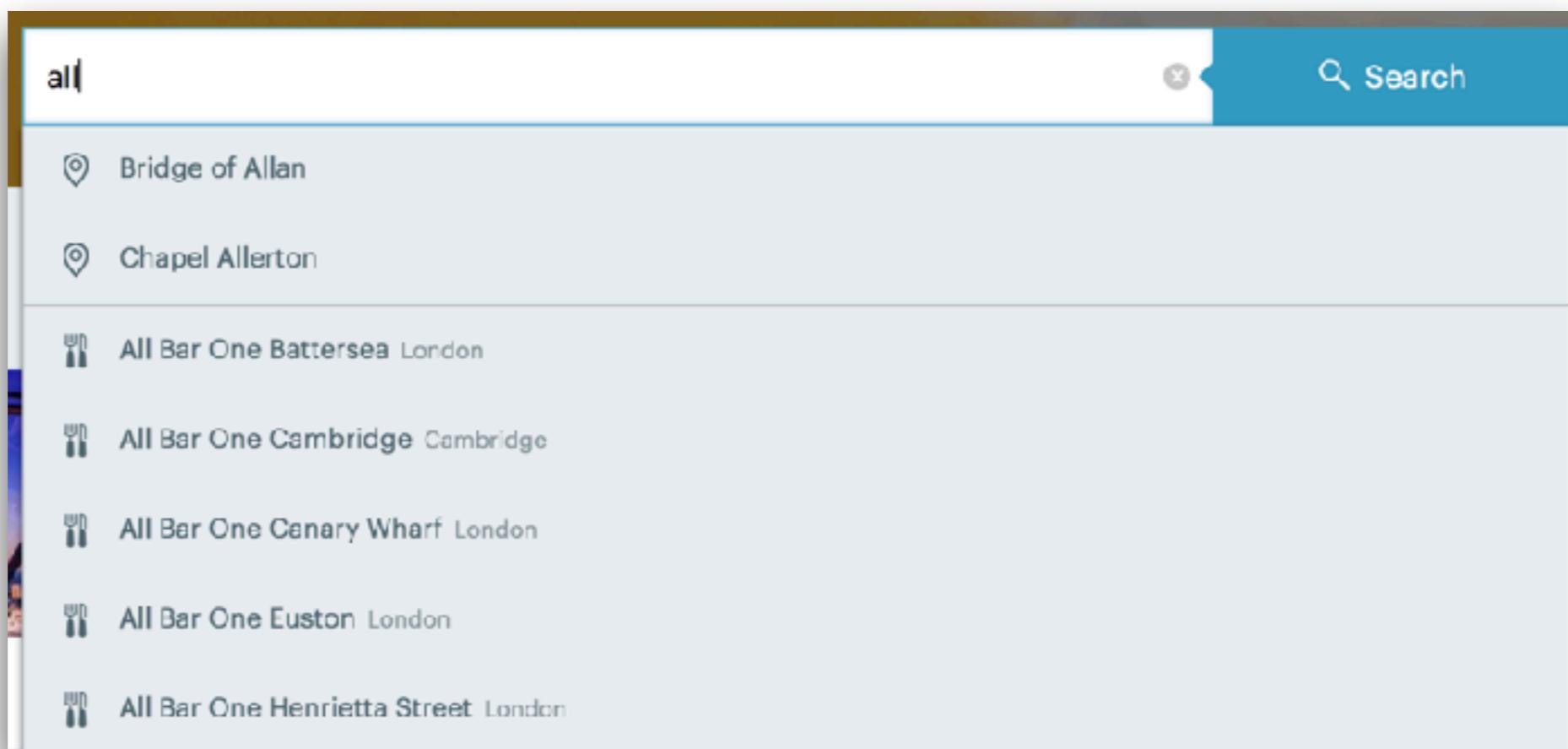
# Asynchronous process

<http://www.bookatable.co.uk/>



# Asynchronous process

<http://www.bookatable.co.uk/>



# Without waiting

Unstable testing !!

```
[Test]
public void search_restaurant()
{
    webdriver.Url = "http://www.bookatable.com/";

    //Action
    IWebElement searchLink = webdriver.FindElement(By.ClassName("global-r
    searchLink.Click();
    IWebElement searchBox = webdriver.FindElement(By.Id("global-navigatio
    searchBox.SendKeys("All");

    //Assert
    IWebElement searchResult = webdriver.FindElement(By.XPath("//html/body
    Assert.AreEqual("", searchResult.Text);
}
```



# Thread.sleep()

## Unstable testing !!

```
[Test]
public void search_restaurant_with_thread_sleep()
{
    webdriver.Url = "http://www.bookatable.com/";

    //Action
    IWebElement searchLink = webdriver.FindElement(By.ClassName("glob
    searchLink.Click();
    IWebElement searchBox = webdriver.FindElement(By.Id("global-navig
    searchBox.SendKeys("All"));

    Thread.Sleep(5000);

    //Assert
    IWebElement searchResult = webdriver.FindElement(By.XPath("/html/
    Assert.AreEqual("", searchResult.Text);
}
```



# Waiting with selenium

Implicit waiting

Explicit waiting



# Implicit waiting

```
[Test]
public void search_restaurant_with_implicit_wait()
{
    webdriver.Url = "http://www.bookatable.com/";
    webdriver.Manage().Timeouts().ImplicitlyWait(TimeSpan.FromSeconds(5));

    //Action
    IWebElement searchLink = webdriver.FindElement(By.ClassName("global-navi
    searchLink.Click();
    IWebElement searchBox = webdriver.FindElement(By.Id("global-navigation_
    searchBox.SendKeys("All"));

    //Assert
    IWebElement searchResult = webdriver.FindElement(By.XPath("/html/body/di
    Assert.AreEqual("", searchResult.Text);
}
```



# Explicit waiting

```
[Test]
public void search_restaurant_with_explicit_wait()
{
    webdriver.Url = "http://www.bookatable.com/";

    //Action
    IWebElement searchLink = webdriver.FindElement(By.ClassName("glob
    searchLink.Click();
    IWebElement searchBox = webdriver.FindElement(By.Id("global-navig
    searchBox.SendKeys("All"));

    //Assert
    WebDriverWait webdriverWait =
        new WebDriverWait(webdriver, TimeSpan.FromSeconds(5));
    IWebElement searchResult =
        webdriverWait.Until(x=>x.FindElement(By.XPath("/html/body/div

    Assert.AreEqual("", searchResult.Text);
}
```



# Tips :: Page load and Script

```
[Test]
public void withPageLoadAndScriptTimeout()
{
    webdriver.Manage().Timeouts()
        .SetPageLoadTimeout(TimeSpan.FromSeconds(5));

    webdriver.Manage().Timeouts()
        .SetScriptTimeout(TimeSpan.FromSeconds(5));
}
```



# Black Hole Proxy Pattern

<https://engineering.groupon.com/2013/misc/black-hole-proxy-for-selenium-testing/>



# Black Hole Proxy Pattern

Try to reduce test instability  
Get rid of as 3-parties uncertain

- Image
- Ads script
- Analytic script
- Social network script



# Setup proxy to browser

```
[SetUp]
public void setup()
{
    var chromeOptions = new ChromeOptions();
    var proxy = new Proxy();
    proxy.Kind = ProxyKind.Manual;
    proxy.IsAutoDetect = false;
    proxy.HttpProxy = "localhost:8888";
    proxy.NoProxy = "localhost,127.0.0.1,*tarad.com";

    chromeOptions.Proxy = proxy;
    webdriver =
        new ChromeDriver(chromeOptions);
}
```



# Need proxy server

```
[SetUp]
public void setup()
{
    var chromeOptions = new ChromeOptions();
    var proxy = new Proxy();
    proxy.Kind = ProxyKind.Manual;
    proxy.IsAutoDetect = false;
    proxy.HttpProxy = "localhost:8888";
    proxy.NoProxy = "localhost,127.0.0.1,*tarad.com";

    chromeOptions.Proxy = proxy;
    webdriver =
        new ChromeDriver(chromeOptions);
}
```



# By pass proxy

```
[SetUp]
public void setup()
{
    var chromeOptions = new ChromeOptions();
    var proxy = new Proxy();
    proxy.Kind = ProxyKind.Manual;
    proxy.IsAutoDetect = false;
    proxy.HttpProxy = "localhost:8888";
    proxy.NoProxy = "localhost,127.0.0.1,*tarad.com";

    chromeOptions.Proxy = proxy;
    webdriver =
        new ChromeDriver(chromeOptions);
}
```



# Proxy server

Use BrowserMob Proxy

<http://bmp.lightbody.net/>

**\$browsermob-proxy -port 8888**



# Run and see result

◆ เข็มขัดของขวัญปีใหม่ กระดังงาเขี้ด (Cadanga Series)

ชื่อสินค้า : เข็มขัดของขวัญปีใหม่ กระดังงาเขี้ด (Cadanga Series)  
สินค้าในสต็อก : มีสินค้า  
ระยะเวลาการขาย : จนกว่าของจะหมด  
ค่าขนส่งขั้นต่ำ : จัดส่งฟรี

---

ราคาปกติ : 3,429 บาท  
ราคายี่ห้อเดียวกัน : **2,999 บาท** (สั่งซื้อออนไลน์ รับแฉ้มคืน 29 บาท(แต้ม))  
ประหยัด : 430 บาท  
จำนวนสินค้า :  (ระบุจำนวนสินค้าที่สั่งซื้อ)

บันทึกลงรายการโปรด

**สั่งซื้อ**

แนะนำ! กดณาชาระเงินผ่านระบบคลาวด์ รับแฉ้มแทนเงินสด มั่นใจได้ของขวาร์ 100%



# Pros

Improve speed of testing

Improve stability

Reduce external environments



# Cons

Broken layout

External content tests are broken

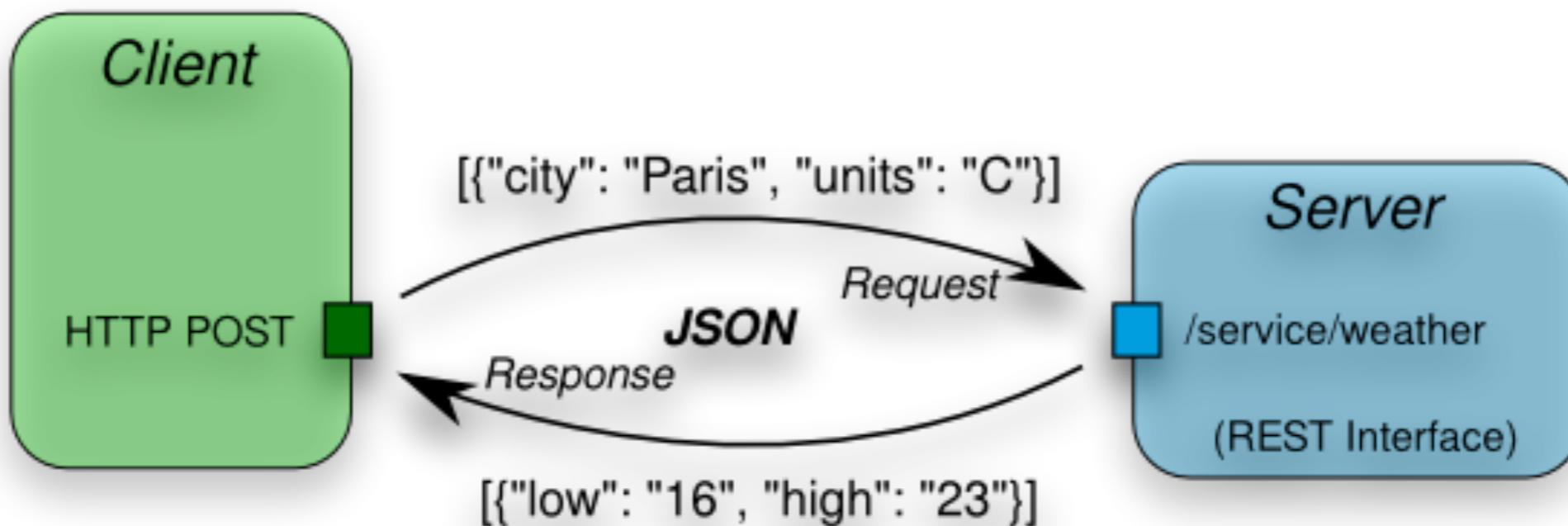


# Stub APIs



# Stub APIs

**JSON / REST / HTTP**



# Stubby with .NET

Stubby4net

<https://github.com/mrak/stubby4net>



# Stubby with .NET

```
$stubby -d <yaml>
```



# Stubby with Java

Stubby4j

<https://github.com/azagniotov/stubby4j>

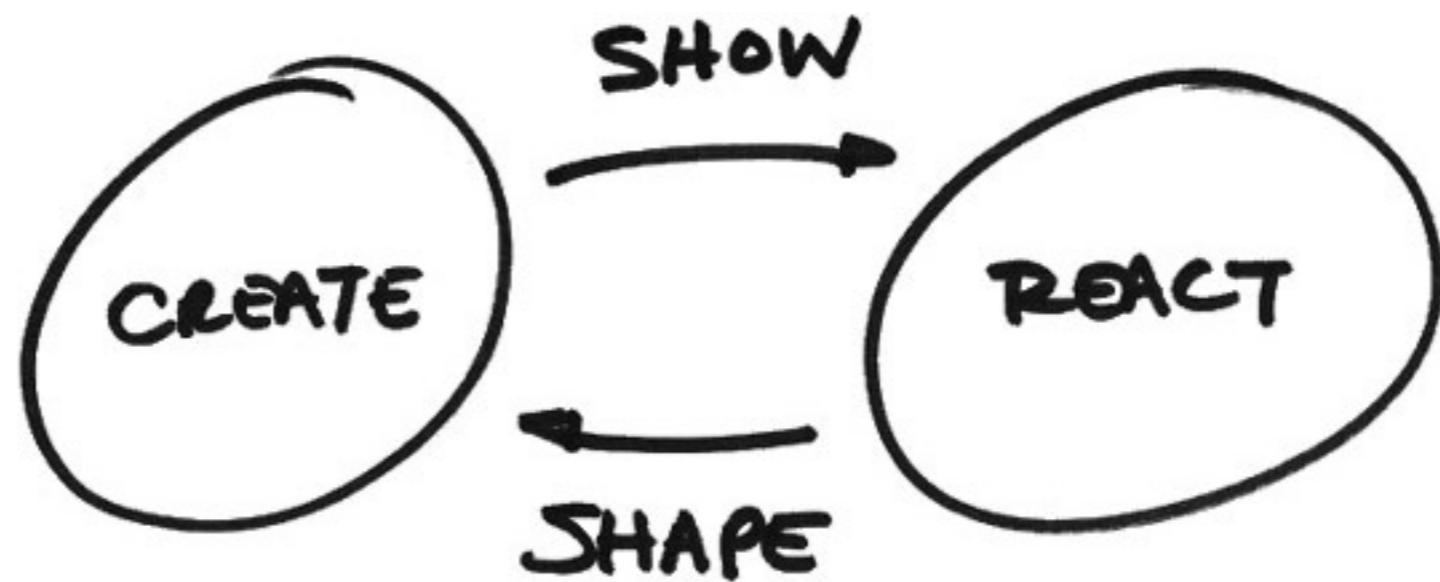


# Stubby with Java

```
$java -jar stubby4j<version> -d <yaml>
```



# Testing the behavior



# Page object pattern



## Contact Us

Home » *Contact Us*

Your Name (required)

Your Email (required)

Subject

Your Message

Send

### Cart

You have 0 items in your shopping

cart. [View Cart](#)

### Advertisement



## Contact Us

Home » *Contact Us*

Your Name (required)

Your Email (required)

Subject

Your Message

Send

Cart

You have 0 items in your shopping

cart. [View Cart](#)

Advertisement



# HEADER

## Contact Us

Home » Contact Us

Your Name (required)

Your Email (required)

Subject

Your Message

Send

## BODY

Cart

You have 0 items in your shopping

[View Cart](#)

Adv.

## SIDE BAR

## FOOTER



# Web page as object

page = ContactUsPage

page.header = PageHeader

page.body = PageBody

page.sidebar = PageSideBar

page.footer = PageFooter



# Page object pattern

Domain Specific Language (DSL)

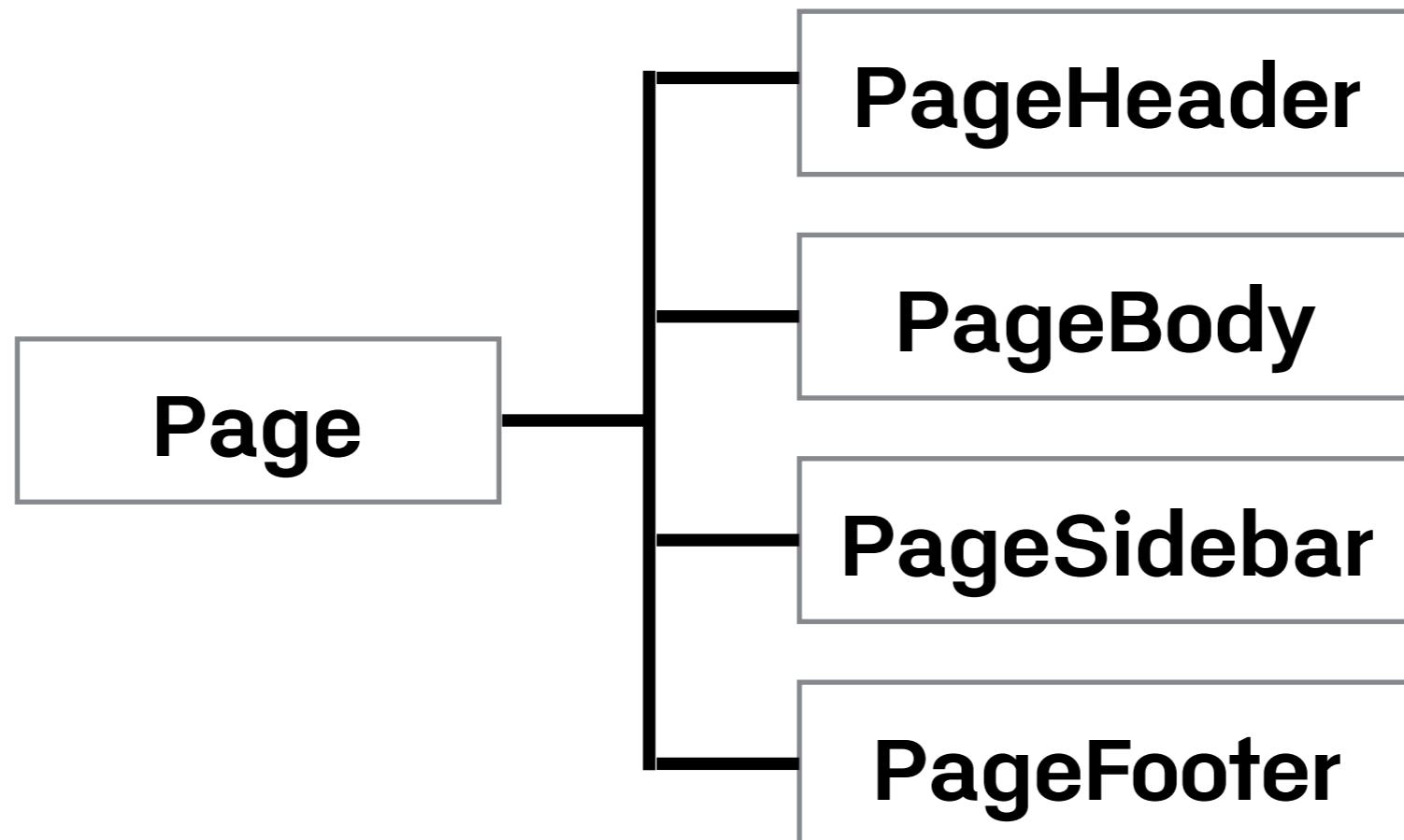
Abstracts the implementation details



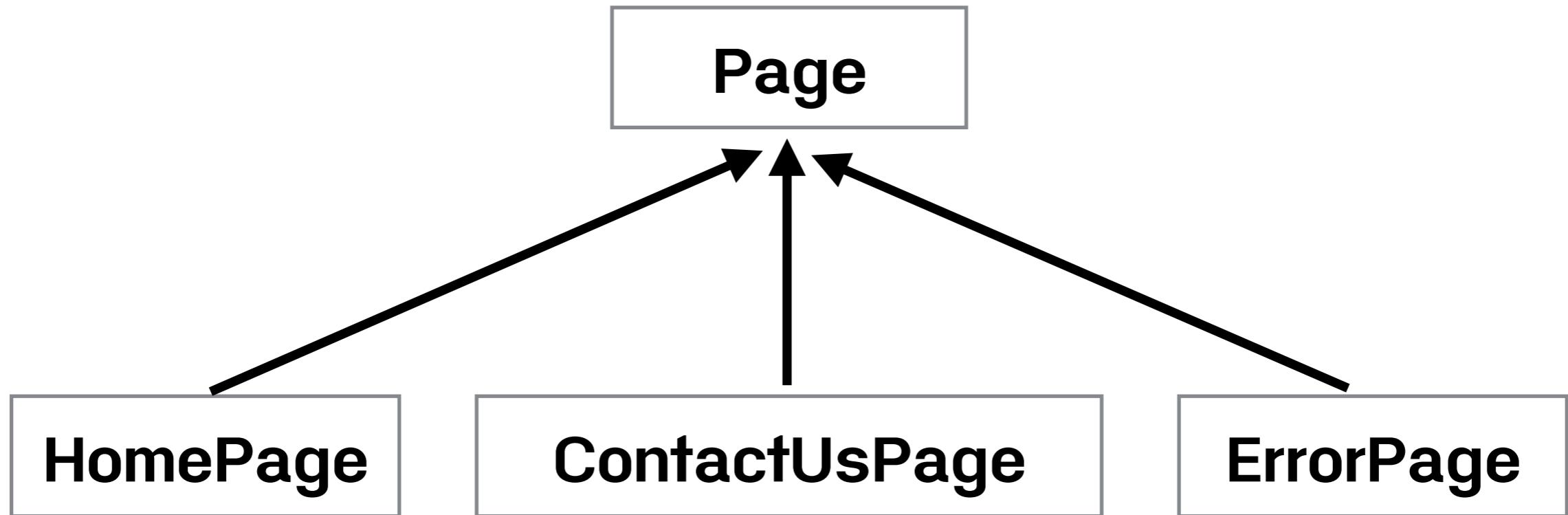
Let's start  
to create Page object



# Class diagram



# Class diagram



# Page object in the right way



# Page object in the right way

Make pages smarter than tests

Make tests smarter than pages

Use module instead of inheritance

Placing logic in Page object



# Workshop page object



# Homepage

① [awful-valentine.com](http://awful-valentine.com)

Home Code » Contact Us Purchase Forms » Store »

The screenshot shows a website for Valentine's Day products. At the top, there's a navigation bar with links to Home, Code, Contact Us, Purchase Forms, and Store. Below the navigation is a main banner featuring a product titled "Closeness and Togetherness" which shows a person with their face obscured by a red ribbon. The price is \$5.77. A red box highlights a section of the page below the banner, which contains five more product cards:

- Closeness and Togetherness** (\$5.77) - A person with a red ribbon obscuring their face.
- Lets Grow Old Together!** (\$88.79) - A man with a surprised expression.
- Taking Long Drives Together** (\$19.99) - A man driving a car.
- Our love is special!!** (\$68.99) - A woman holding two orange slices over her eyes.
- Never Forget The Special Day!** (\$99.99) - A baby crying.

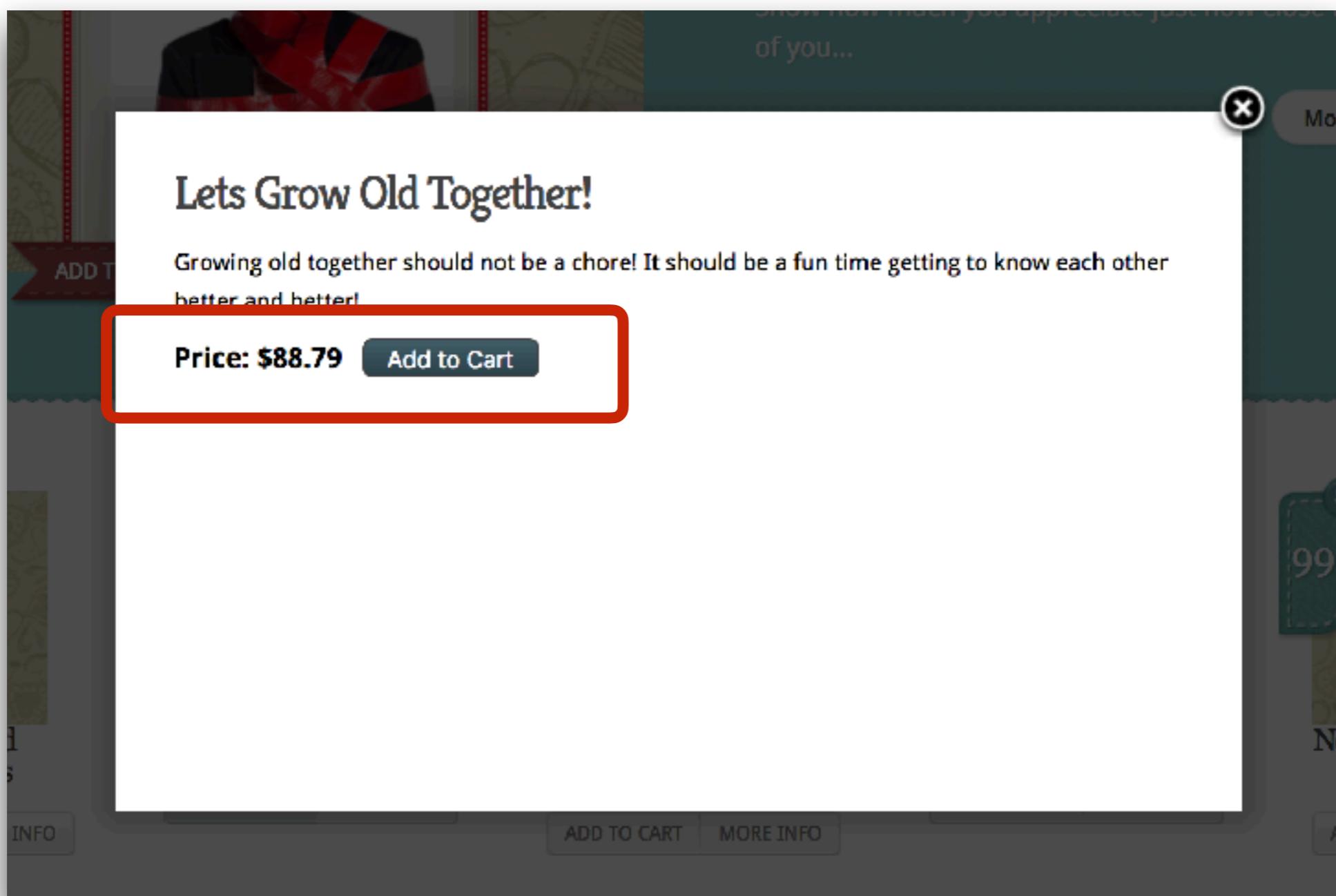
Below this section is a "RECENT PRODUCTS" row with four smaller items:

- Closeness and Togetherness** (\$5.77)
- Lets Grow Old Together!** (\$88.79)
- Taking Long Drives Together** (\$19.99)
- Our love is special!!** (\$68.99)

To the right, there's a "Cart" section indicating 0 items in the shopping cart.



# Add to cart



# Contact Us page

The screenshot shows a "Contact Us" page with a navigation bar at the top. The navigation bar includes links for Home, Code, Contact Us, Purchase Forms, and Store. Below the navigation, the main title "Contact Us" is displayed. A breadcrumb trail "Home » Contact Us" is shown. On the left side, there are four input fields labeled "Your Name (required)", "Your Email (required)", "Subject", and "Your Message", each with a corresponding text input box. At the bottom left is a "Send" button. On the right side, there is a "Cart" section enclosed in a red box. The cart summary states: "You have 2 items (\$94.56) in your shopping cart." It lists two items: "Closeness and Togetherness" priced at \$5.77 (1 x \$5.77) and "Lets Grow Old Together!" priced at \$88.79 (1 x \$88.79). The subtotal is \$94.56. Below the cart summary are "View Cart" and "Checkout" links. At the very bottom of the page is an "Advertisement" section.

Home Code » **Contact Us** Purchase Forms » Store »

## Contact Us

Home » Contact Us

Your Name (required)

Your Email (required)

Subject

Your Message

Send

Cart

You have 2 items (\$94.56) in your shopping cart.

Closeness and Togetherness	\$5.77
1 x \$5.77	
Lets Grow Old Together!	\$88.79
1 x \$88.79	
<b>Subtotal:</b>	<b>\$94.56</b>

[View Cart](#) | [Checkout](#)

Advertisement



# Sidebar cart

Cart

You have 2 items (\$94.56) in your shopping cart.

<b>Closeness and Togetherness</b>	<b>\$5.77</b>
1 x \$5.77	
<b>Lets Grow Old Together!</b>	<b>\$88.79</b>
1 x \$88.79	
<b>Subtotal: \$94.56</b>	

[View Cart | Checkout](#)



# Let's coding + demo



# Page

```
public class Page
{
    protected IWebDriver webdriver;
    public Page(IWebDriver webdriver)
    {
        this.webdriver = webdriver;
        verify();
    }

    virtual public PageSidebar getPageSidebar()
    {
        return new PageSidebar(webdriver);
    }

    virtual public string getPagePath()
    {
        throw new NotImplementedException();
    }

    private void verify()
    {
        if (!currentPath().Equals(getPagePath()))
        {
            throw new NotImplementedException();
        }
    }
}
```



# HomePage

```
public class HomePage: Page
{
    public HomePage(IWebDriver webdriver): base(webdriver)
    {
    }

    public override string getPagePath()
    {
        return "/";
    }

    public SpecialItems getItems()
    {
        return new SpecialItems(webdriver);
    }
}
```



# List of special items

```
public class SpecialItems
{
    private IWebDriver webdriver;
    private List<SpecialItem> items = new List<SpecialItem>();

    public SpecialItems(IWebDriver webdriver)
    {
        this.webdriver = webdriver;

        foreach (IWebElement element in
            webdriver.FindElements(By.ClassName("special-item")))
        {
            this.items.Add(new SpecialItem(webdriver, element));
        }
    }

    public SpecialItem get(int index)
    {
        return this.items[index];
    }
}
```



# Special item

```
public class SpecialItems
{
    private IWebDriver webdriver;
    private List<SpecialItem> items = new List<SpecialItem>();

    public SpecialItems(IWebDriver webdriver)
    {
        this.webdriver = webdriver;

        foreach (IWebElement element in
            webdriver.FindElements(By.ClassName("special-item")))
        {
            this.items.Add(new SpecialItem(webdriver, element));
        }
    }

    public SpecialItem get(int index)
    {
        return this.items[index];
    }
}
```



# Contact Us Page

```
public class ContactUsPage: Page
{
    public ContactUsPage(IWebDriver webdriver) : base(webdriver)
    {
    }

    public override string getPagePath()
    {
        return "/contact-us/";
    }
}
```



# Page Sidebar

```
public class PageSidebar
{
    private IWebDriver webdriver;
    public PageSidebar(IWebDriver webdriver)
    {
        this.webdriver = webdriver;
    }

    public SidebarCart getSidebarCart()
    {
        return new SidebarCart(this.webdriver);
    }
}
```



# Page Sidebar

```
public class PageSidebar
{
    private IWebDriver webdriver;
    public PageSidebar(IWebDriver webdriver)
    {
        this.webdriver = webdriver;
    }

    public SidebarCart getSidebarCart()
    {
        return new SidebarCart(this.webdriver);
    }
}
```



# Sidebar cart

```
public class SidebarCart
{
    IWebDriver webdriver;

    public SidebarCart(IWebDriver webdriver)
    {
        this.webdriver = webdriver;
    }

    public string getSummary()
    {
        return webdriver.FindElement(By.Id("Cart66WidgetCartEmptyAdvanced")).Text;
    }

    public string getSubTotal()
    {
        return webdriver.FindElement(By.ClassName("Cart66Subtotal")).Text;
    }
}
```



# Write first test with page object



# Add product to cart

```
[Test]
public void addProductToCart()
{
    webdriver.Url = "http://awful-valentine.com/";

    HomePage homepage = new HomePage(webdriver);
    homepage.getItems().get(0).addToCart();

    webdriver.Url = "http://awful-valentine.com/contact-us/";
    ContactUsPage contactUsPage = new ContactUsPage(webdriver);
    Assert.AreEqual("You have 1 item ($5.77) in your shopping cart.",
                   contactUsPage.getPageSidebar().getSidebarCart().getSummary());
}
```



# Back to comment product !!



# Let's coding



# Pros

DSL framework

Testing behavior

DRY

Modular and reusable

Clear intention



# Cons

Increase complexity

Complexity to use and maintain



# Growing the test suites



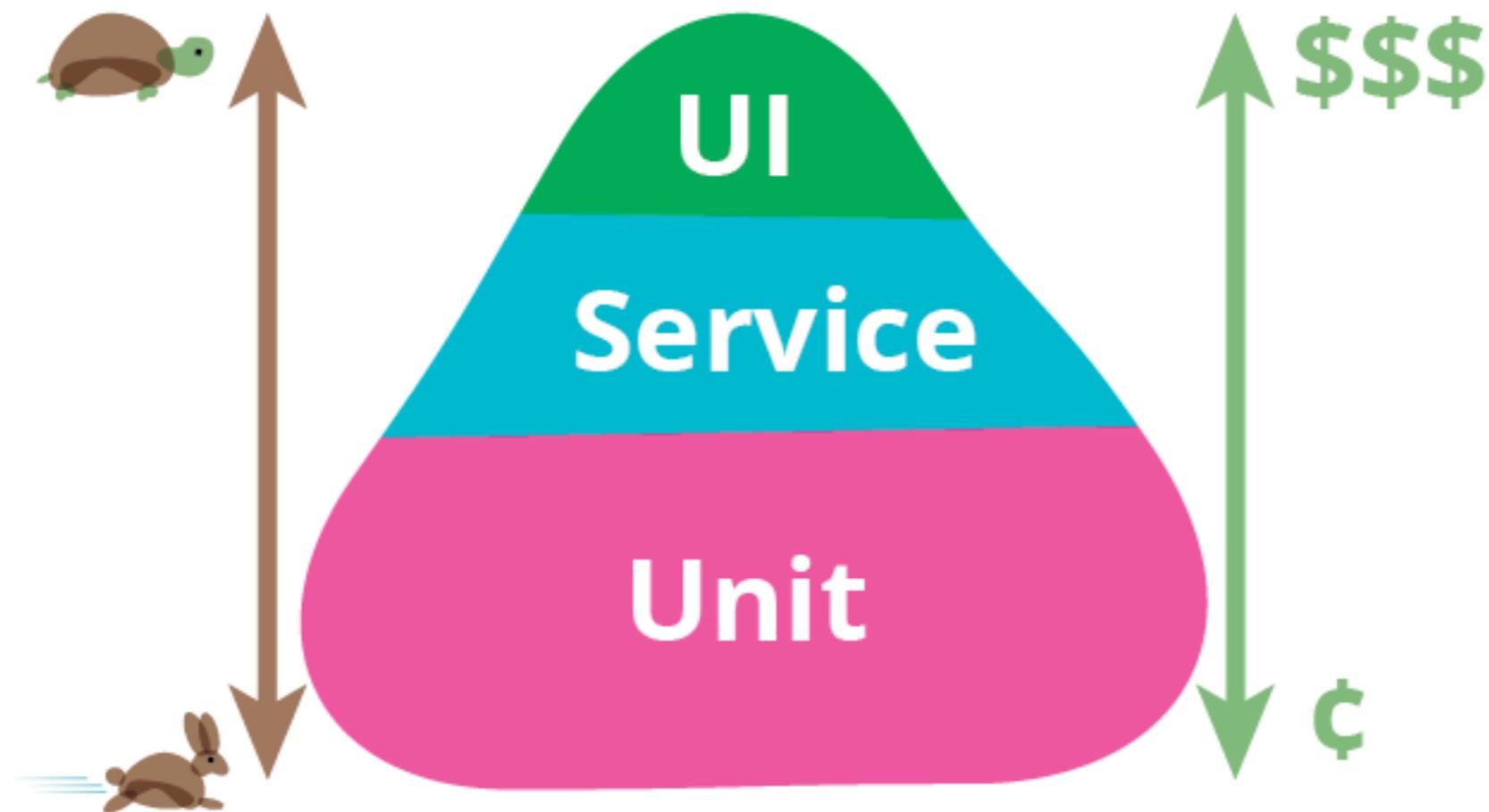
# Strategy for writing test ?

100% test coverage

Prioritize the growth of the test suites



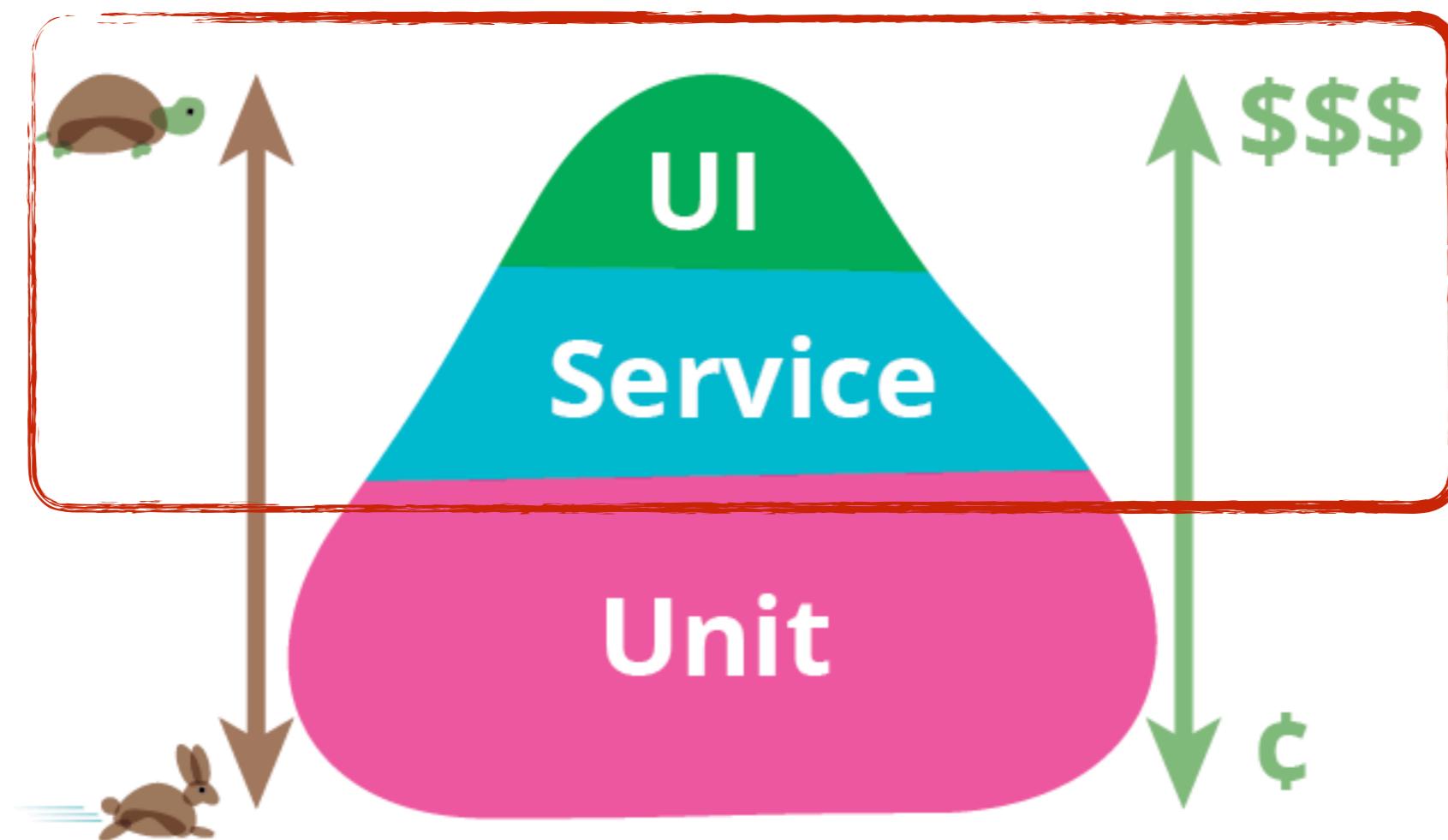
# Test Pyramid



<http://martinfowler.com/bliki/TestPyramid.html>



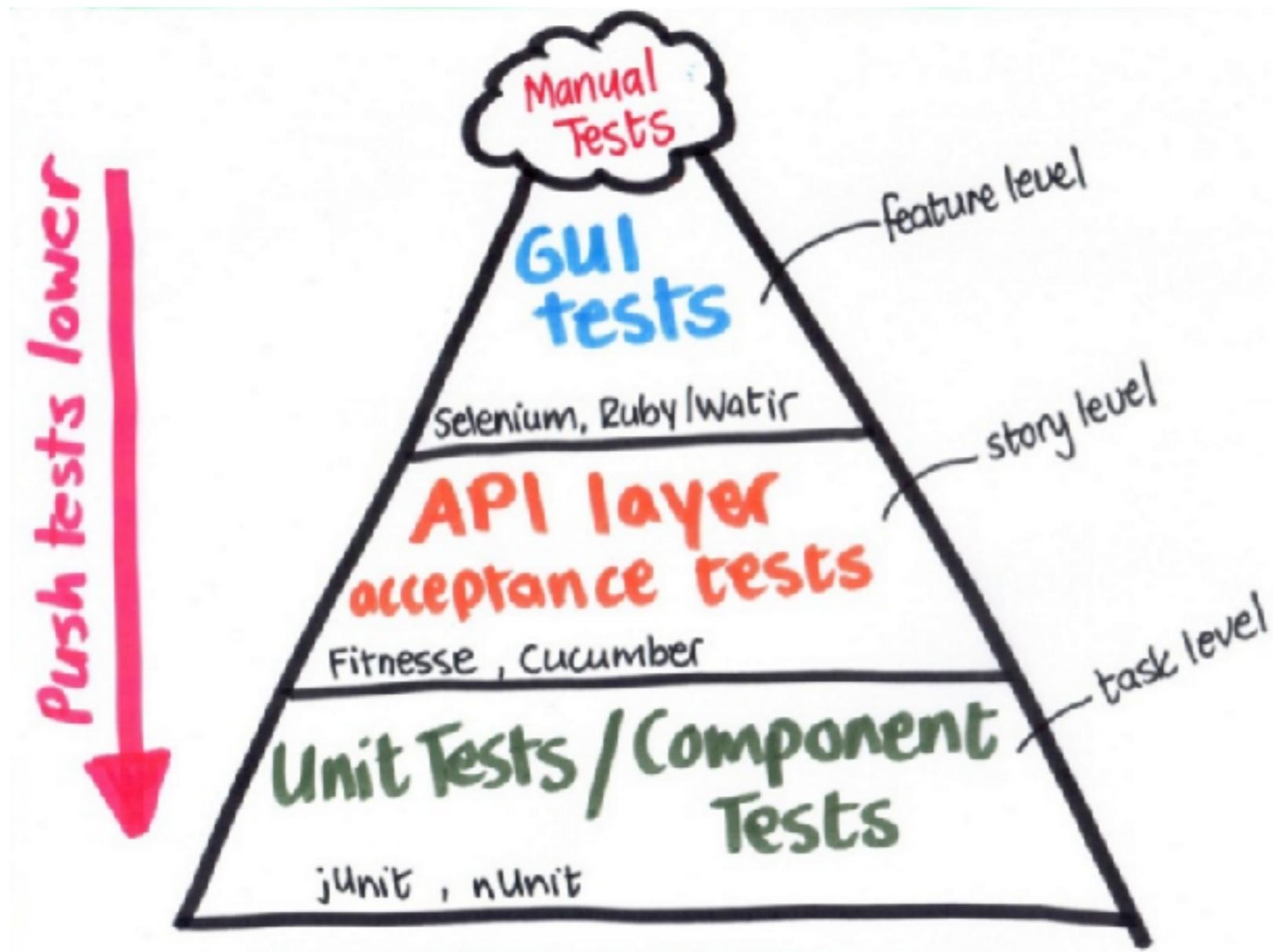
# Selenium



<http://martinfowler.com/bliki/TestPyramid.html>



# Test Pyramid



<http://www.slideshare.net/growingagile/expoqa-tutorial-agile-testing-techniques-for-the-whole-team>



# The smoke test suite



# The smoke test suite

Test the new code

With **small test suite** to make sure

Production environment is up and running !!



# To make sure ?

Running application (ERROR 500)

Database connection

Abnormal amount of execution



# Keep in mind

Avoid writing to database, if can't clean !!

Don't test too much



# Strategy for writing test



Money path suite  
New feature growth  
Bug-driven growth  
Regression suite  
99% coverage suite

# Continuous Integration (CI)



# Continuous Integration



Jenkins



Bamboo



TeamCity

> go™



Hudson



# Continuous Integration



Jenkins

The Bamboo logo features a circular icon with four arrows pointing clockwise, followed by the word "Bamboo" in a large, bold, sans-serif font.

Bamboo



TeamCity

CI is about **what people do**  
not about what tools they use

The Hudson logo features a large, light gray arrow pointing to the right, with the word "Hudson" in a large, light gray sans-serif font to its right.

Hudson



The Wercker logo features a green shield icon with three vertical bars inside, followed by the word "wercker" in a green, lowercase, sans-serif font.

The CircleCI logo features a gray circle with a white 'C' icon, followed by the word "circleci" in a gray, lowercase, sans-serif font.



# CI Practice

Discipline to integrate frequently



# CI Practice

**Strive to make small change**

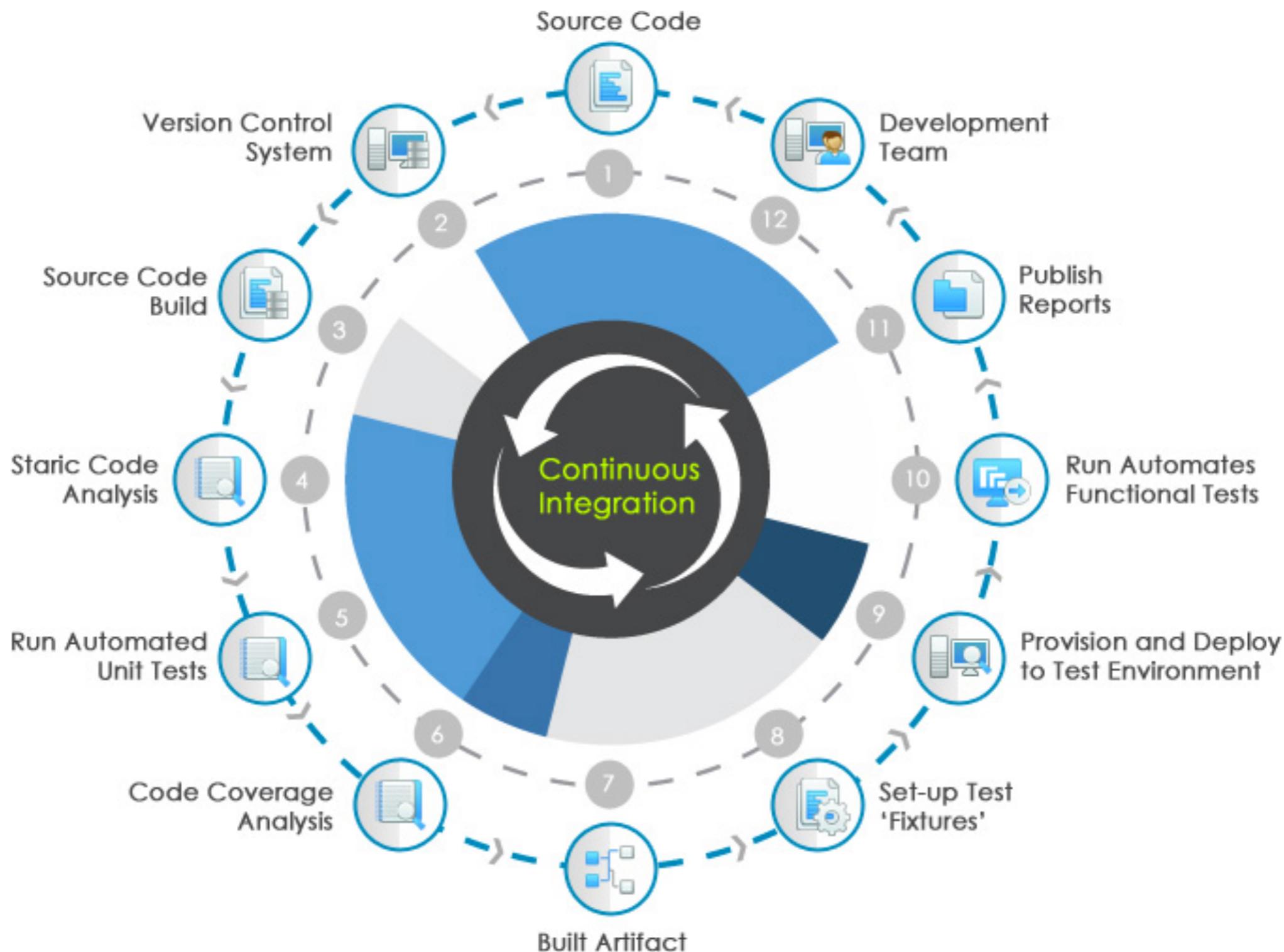


# CI Practice

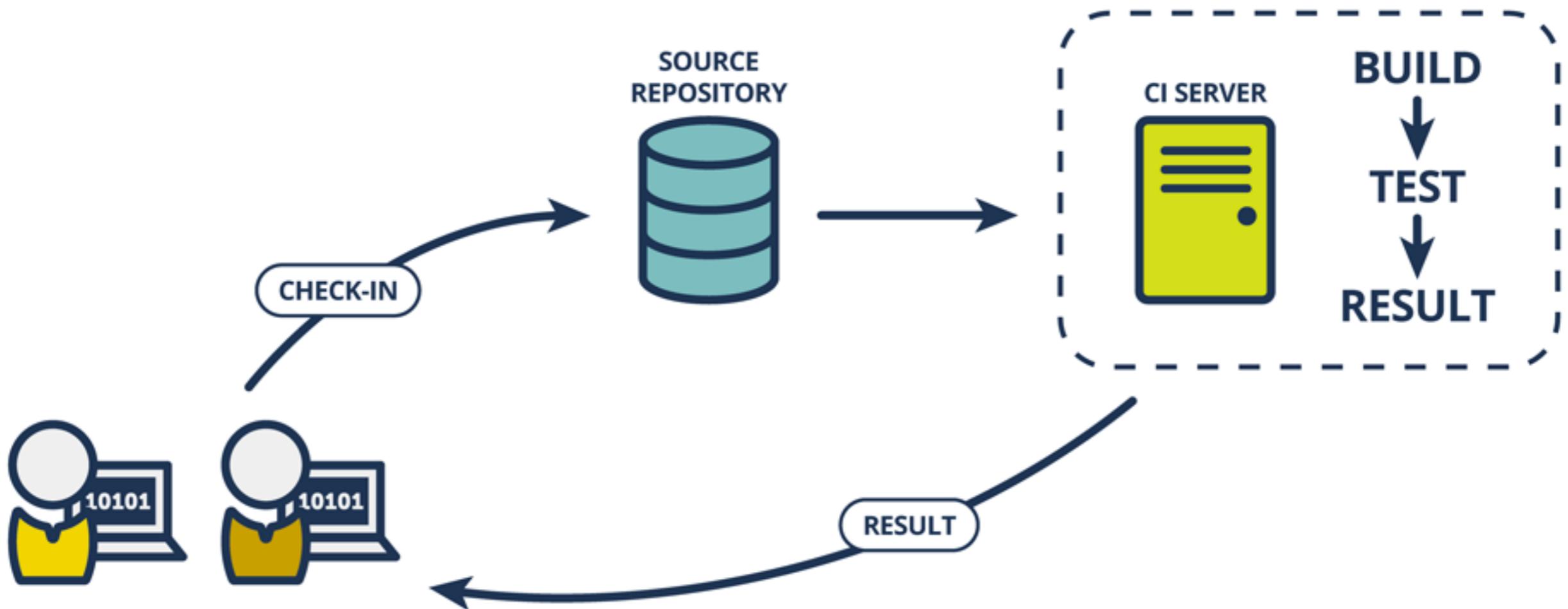
Strive for **fast feedback**



# Continuous Integration



# Continuous Integration



# Components in CI system

Test environment

Test data

Tests and test stability

**Test nodes ?**

**CI server ?**



# Components



Jenkins +



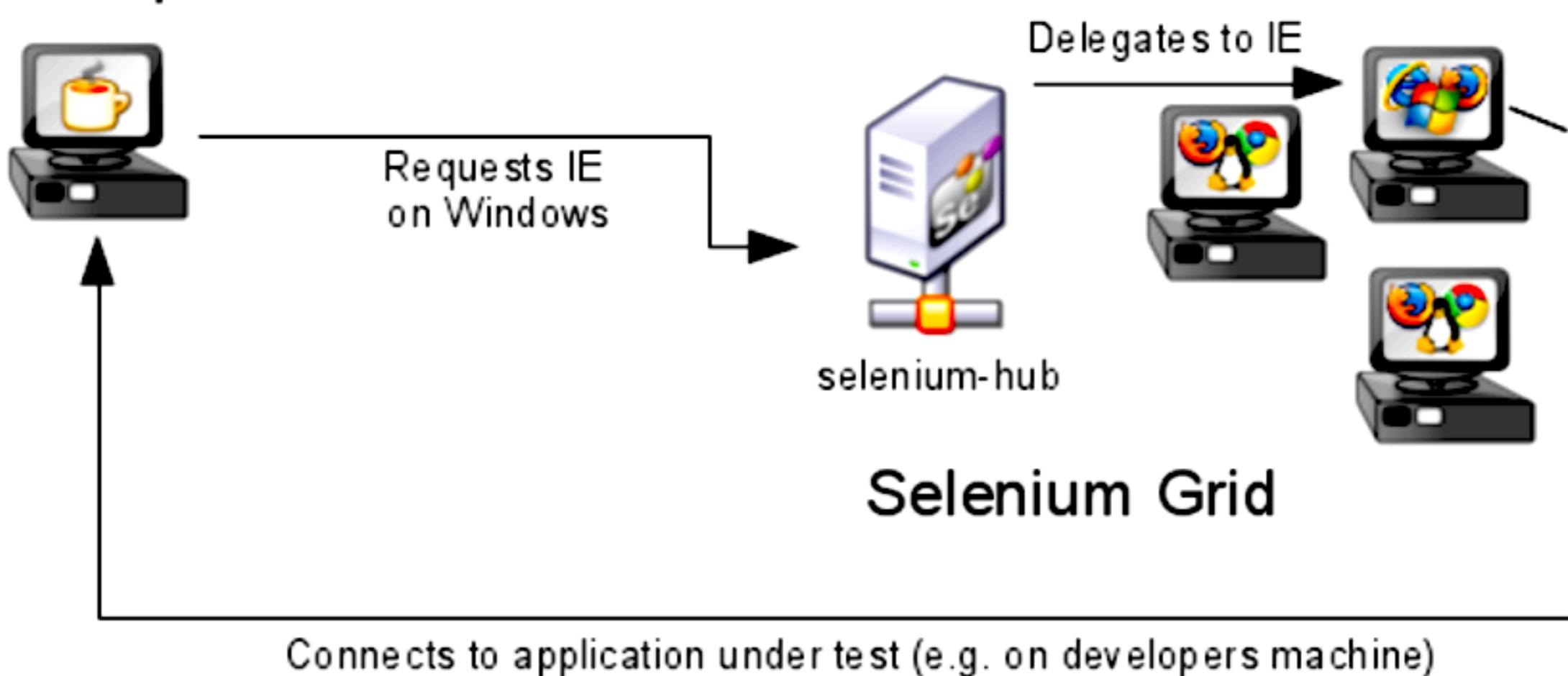
Grid

# Selenium grid



# Selenium Grid

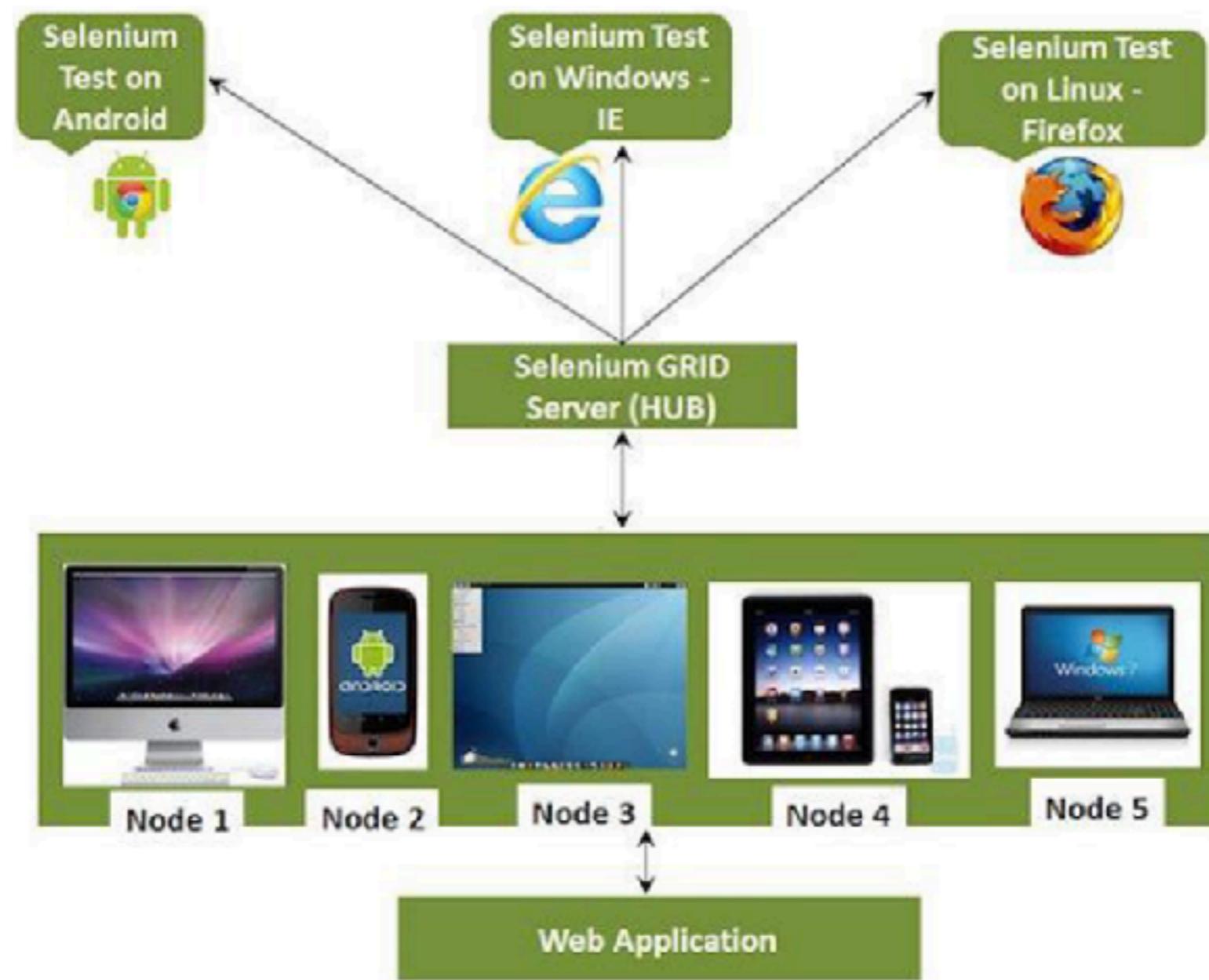
Developer



<http://www.serveridol.com/2013/05/04/selenium-grid-hub-and-node-communication/>



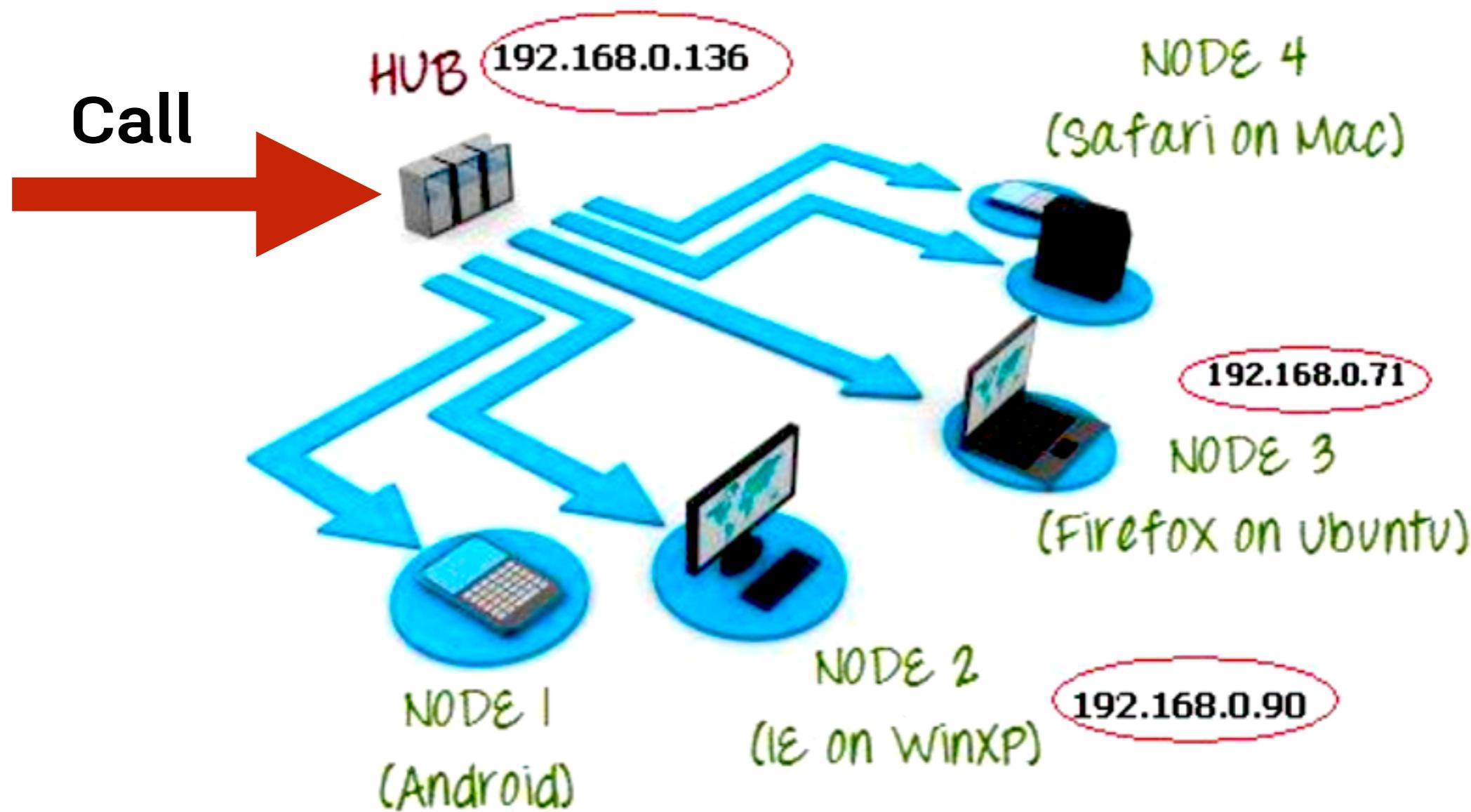
# Selenium Grid



<http://www.hub4tech.com/selenium/selenium-grid>



# Hub and Nodes



<http://www.serveridol.com/2013/05/04/selenium-grid-hub-and-node-communication/>



# Installation

Download selenium standalone server

<http://docs.seleniumhq.org/download/>

## Selenium Standalone Server

The Selenium Server is needed in order to run Remote Selenium capable of running Selenium RC directly, rather it does it through WebDriverBackedSelenium interface.

Download version [3.0.1](#)



# Start selenium hub

```
$java -jar selenium-server-standalone-3.0.1.jar  
-role hub
```

```
11:46:38.894 INFO - Selenium build info: version: '3.0.1', revision: '1969d75'  
11:46:38.895 INFO - Launching Selenium Grid hub  
2016-12-12 11:46:39.543:INFO::main: Logging initialized @1047ms  
11:46:39.559 INFO - Will listen on 4444  
2016-12-12 11:46:39.628:INFO:osjs.Server:main: jetty-9.2.15.v20160210  
2016-12-12 11:46:39.671:INFO:osjsh.ContextHandler:main: Started o.s.j.s.ServletContextHandler@62fdb4a6{/null,AVAILABLE}  
2016-12-12 11:46:39.700:INFO:osjs.ServerConnector:main: Started ServerConnector@49e202ad{HTTP/1.1}{0.0.0.0:4444}  
2016-12-12 11:46:39.700:INFO:osjs.Server:main: Started @1205ms  
11:46:39.701 INFO - Nodes should register to http://192.168.1.12:4444/grid/register/  
11:46:39.701 INFO - Selenium Grid hub is up and running
```



# Start selenium node

```
$java -jar selenium-server-standalone-3.0.1.jar  
-role wd -hub http://localhost:4444 -port 5555
```

Default port

```
$java -jar selenium-server-standalone-3.0.1.jar  
-role wd -hub http://localhost:4444 -port 6666
```



# Selenium grid console

The screenshot shows the Selenium Grid Console interface running on localhost:4444/grid/console. It displays two nodes, each with a title bar indicating they are DefaultRemoteProxy (version : 3.0.1) with id : http://192.168.1.12:5555 and mixed OS. Each node has tabs for 'Browsers' and 'Configuration'. Under 'Browsers', there are sections for WebDriver (with icons for Firefox, Chrome, and Edge) and v: (with icons for Firefox, Chrome, and Edge). A 'view config' link is located at the bottom left of the left node's panel.

Grid Console v.3.0.1

Help

DefaultRemoteProxy (version : 3.0.1)  
id : http://192.168.1.12:5555, OS : mixed OS

DefaultRemoteProxy (version : 3.0.1)  
id : http://192.168.1.12:5556, OS : mixed OS

Browsers Configuration

WebDriver

v:

v:

Browsers Configuration

WebDriver

v:

v:

v:

[view config](#)



# Configuration selenium grid



# Configuration hub

```
$java -jar selenium-server-standalone-3.0.1.jar  
      -role hub  
      -hubConfig <json file>
```



# Configuration hub

```
{"_comment" : "Configuration for Hub - hubConfig.json",
"host": ip,
"maxSessions": 5,
"port": 4444,
"cleanupCycle": 5000,
"timeout": 300000,
"newSessionWaitTimeout": -1,
" servlets": [],
"prioritizer": null,
"capabilityMatcher": "org.openqa.grid.internal.utils.DefaultCapabilityMatcher",
"throwOnCapabilityNotPresent": true,
"nodePolling": 180000,
"platform": "WINDOWS"}
```

<https://github.com/SeleniumHQ/selenium/blob/master/java/server/src/org/openqa/grid/common/defaults/DefaultHub.json>



# Configuration node

**-Dwebdriver.chrome.driver=chromedriver.exe**



# Configuration node with json

**-nodeConfig <json file>**

<https://github.com/SeleniumHQ/selenium/blob/master/java/server/src/org/openqa/grid/common/defaults/DefaultNodeWebDriver.json>

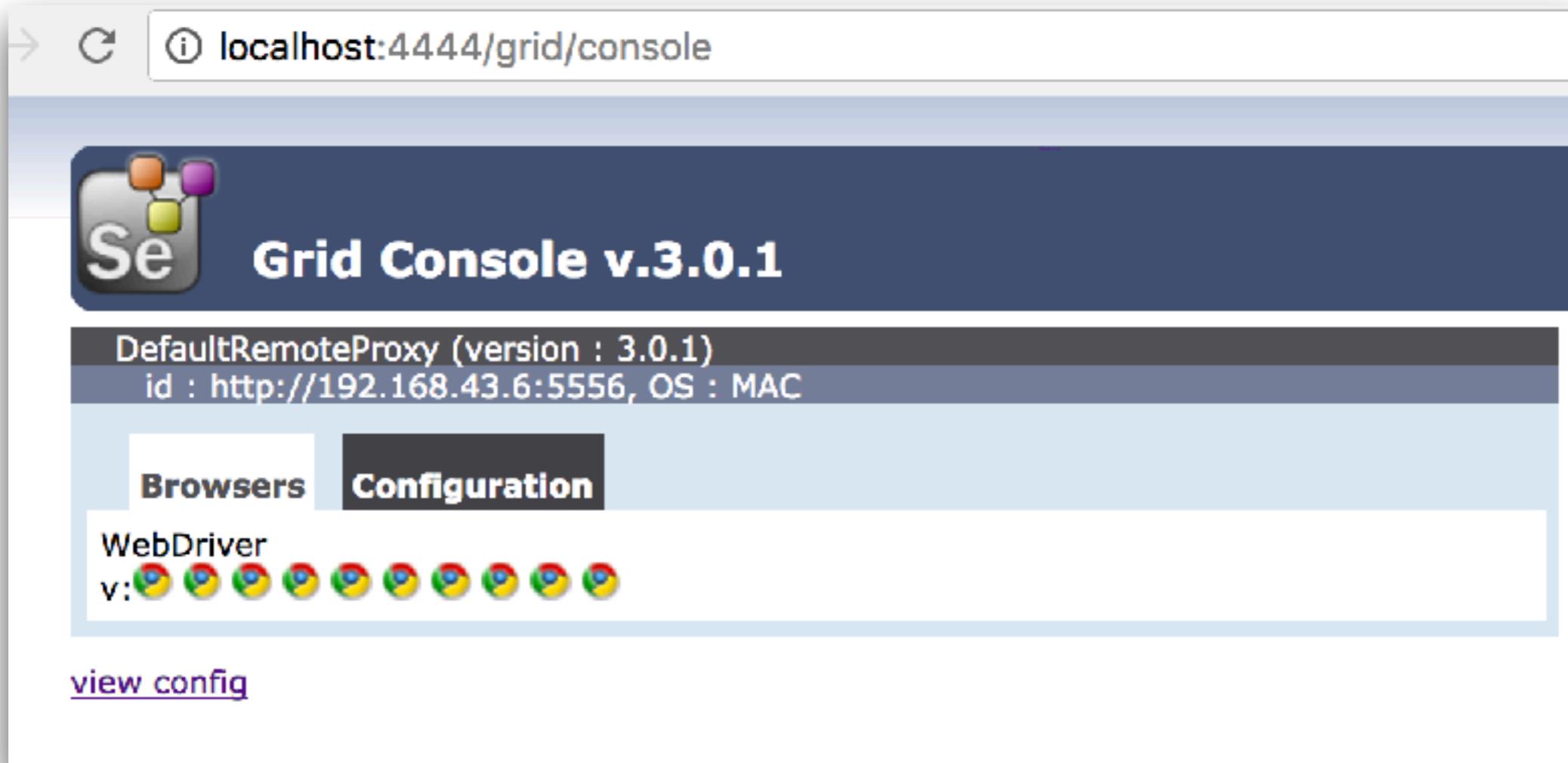


# Node for Google Chrome

```
{  
  "capabilities":  
  [  
    {  
      "browserName": "chrome",  
      "maxInstances": 10,  
      "seleniumProtocol": "WebDriver"  
    }  
  ],  
}
```



# Node for Google Chrome



# Workshop

The screenshot shows the Grid Console v.3.0.1 interface running in a web browser at localhost:4444/grid/console. The interface is divided into four main sections, each representing a different browser instance:

- Top Left Section:** DefaultRemoteProxy (version : 3.0.1) id : http://192.168.1.34:5556, OS : MAC. Contains a "Browsers" tab and a "Configuration" tab. Under "Browsers", there are icons for various browsers including Internet Explorer, Firefox, and Chrome. Below the tabs, it lists "v:" followed by a row of browser icons.
- Top Right Section:** DefaultRemoteProxy (version : 3.0.1) id : http://192.168.1.34:5558, OS : MAC. Contains a "Browsers" tab and a "Configuration" tab. Under "Browsers", there are icons for various browsers including Internet Explorer, Firefox, and Chrome. Below the tabs, it lists "v:" followed by a row of browser icons.
- Bottom Left Section:** DefaultRemoteProxy (version : 3.0.1) id : http://192.168.1.34:5557, OS : MAC. Contains a "Browsers" tab and a "Configuration" tab. Under "Browsers", there are icons for various browsers including Internet Explorer, Firefox, and Chrome. Below the tabs, it lists "v:" followed by a row of browser icons.
- Bottom Right Section:** DefaultRemoteProxy (version : 3.0.1) id : http://192.168.1.34:5559, OS : MAC. Contains a "Browsers" tab and a "Configuration" tab. Under "Browsers", there are icons for various browsers including Internet Explorer, Firefox, and Chrome. Below the tabs, it lists "v:" followed by a row of browser icons.

At the bottom left of the interface, there is a link labeled "view config".



# Working with selenium grid



# Hello selenium grid

```
IWebDriver webdriver;

[SetUp]
public void setup()
{
    DesiredCapabilities capabilities = new DesiredCapabilities();
    capabilities = DesiredCapabilities.Chrome();
    capabilities.SetCapability(
        CapabilityType.BrowserName, "chrome");
    capabilities.SetCapability(
        CapabilityType.Platform, new Platform(PlatformType.Mac));

    webdriver = new RemoteWebDriver(
        new Uri("http://localhost:4444/wd/hub"), capabilities);
    webdriver.Manage().Window.Maximize();
}
```



# Hello selenium grid

```
IWebDriver webdriver;  
  
[SetUp]  
public void setup()  
{  
    DesiredCapabilities capabilities = new DesiredCapabilities();  
    capabilities = DesiredCapabilities.Chrome();  
    capabilities.SetCapability(  
        CapabilityType.BrowserName, "chrome");  
    capabilities.SetCapability(  
        CapabilityType.Platform, new Platform(PlatformType.Mac));  
  
    webdriver = new RemoteWebDriver(  
        new Uri("http://localhost:4444/wd/hub"), capabilities);  
    webdriver.Manage().Window.Maximize();  
}
```

Browser



# Hello selenium grid

```
IWebDriver webdriver;  
  
[SetUp]  
public void setup()  
{  
    DesiredCapabilities capabilities = new DesiredCapabilities();  
    capabilities = DesiredCapabilities.Chrome();  
    capabilities.SetCapability(  
        CapabilityType.BrowserName, "chrome");  
    capabilities.SetCapability(  
        CapabilityType.Platform, new Platform(PlatformType.Mac));  
  
    webdriver = new RemoteWebDriver(  
        new Uri("http://localhost:4444/wd/hub"), capabilities);  
    webdriver.Manage().Window.Maximize();  
}
```

OS



# Hello selenium grid

```
IWebDriver webdriver;  
  
[SetUp]  
public void setup()  
{  
    DesiredCapabilities capabilities = new DesiredCapabilities();  
    capabilities = DesiredCapabilities.Chrome();  
    capabilities.SetCapability(  
        CapabilityType.BrowserName, "chrome");  
    capabilities.SetCapability(  
        CapabilityType.Platform, new Platform(CapabilityType.Mac));  
  
    webdriver = new RemoteWebDriver(  
        new Uri("http://localhost:4444/wd/hub"), capabilities);  
    webdriver.Manage().Window.Maximize();  
}
```

Selenium  
hub



# Pros

Scale by distributed test to another machines

Parallel testing

Cross-platform



# Cons

Maintainability

Complexity to use and maintain



# Selenium grid extras

<https://github.com/groupon/Selenium-Grid-Extras>



# Selenium on cloud



# Selenium on cloud



BrowserStack

 SAUCE LABS



 CrossBrowserTesting  
A SMARTBEAR COMPANY



# Demo



**BrowserStack**

<https://www.browserstack.com/automate/c-sharp>



# Choose CI Server



**Jenkins**



**Jenkins**



**Jenkins**



**Jenkins**



**Jenkins**



**Jenkins**



**Jenkins**

# Installation

Download WAR file

<https://jenkins-ci.org/>



# Jenkins

Build great things at any scale

The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.

[Download Jenkins](#)

Get 2.19.4 LTS .war or the latest 2.36 weekly release



# Download

## LTS Release

LTS (Long-Term Support) releases are chosen every 12 weeks from the stream of regular releases as the stable release for that time period.

 2.19.4 .war ▾

[Changelog](#) | [Upgrade Guide](#) | [Past Releases](#)

## Weekly Release

A new release is produced weekly to deliver bug fixes and features to users and plugin developers.

 2.36 .war ▾

[Changelog](#) | [Past Releases](#)

Download Jenkins

Get 2.19.4 LTS .war or the latest 2.36 weekly release



# Start server

```
$java -jar jenkins.war
```



# Configuration

<http://localhost:8080>

The screenshot shows the Jenkins 'Getting Started' configuration screen. At the top, it says 'Unlock Jenkins'. Below that, a message states: 'To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server: /Users/somkiat/data/slide/selenium/demo/secrets/initialAdminPassword'. It then instructs the user to 'Please copy the password from either location and paste it below.' A red box highlights the 'Administrator password' input field, which is currently empty. In the bottom right corner of the form, there is a blue 'Continue' button.



# Fill in admin password

```
*****  
*****  
*****
```

Jenkins initial setup is required. An admin user has been created.  
Please use the following password to proceed to installation:

2d68139030734c5b85aa655f1726c845

This may also be found at: /Users/somkiat/data/slide/selenium/de

```
*****  
*****  
*****
```



# Choose plug-ins

Getting Started

## Customize Jenkins

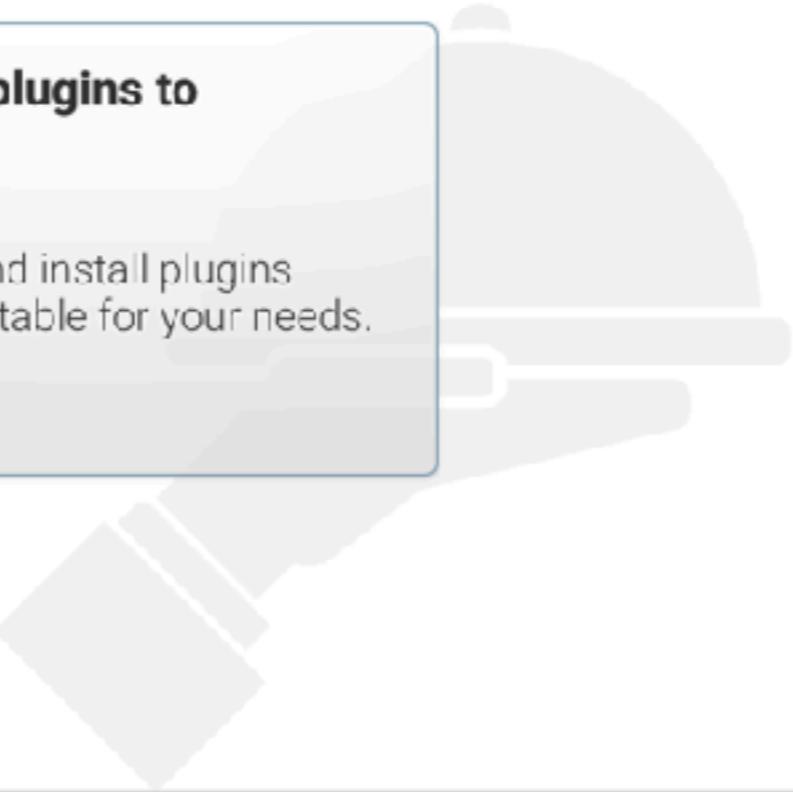
Plugins extend Jenkins with additional features to support many different needs.

**Install suggested plugins**

Install plugins the Jenkins community finds most useful.

**Select plugins to install**

Select and install plugins most suitable for your needs.



Jenkins 2.19.4



# Installing plug-ins

Getting Started

## Getting Started

Progress bar: 10%

<input checked="" type="checkbox"/> Folders Plugin	<input type="checkbox"/> OWASP Markup Formatter Plugin	<input type="checkbox"/> build timeout plugin	<input type="checkbox"/> Credentials Binding Plugin	** bouncycastle API Plugin Folders Plugin
<input type="checkbox"/> Timestamper	<input type="checkbox"/> Workspace Cleanup Plugin	<input type="checkbox"/> Ant Plugin	<input type="checkbox"/> Gradle Plugin	
<input type="checkbox"/> Pipeline	<input type="checkbox"/> GitHub Organization Folder Plugin	<input type="checkbox"/> Pipeline: Stage View Plugin	<input type="checkbox"/> Git plugin	
<input type="checkbox"/> Subversion Plug in	<input type="checkbox"/> SSH Slaves plugin	<input type="checkbox"/> Matrix Authorization Strategy Plugin	<input type="checkbox"/> PAM Authentication plugin	
<input type="checkbox"/> LDAP Plugin	<input type="checkbox"/> Email Extension Plugin	<input type="checkbox"/> Mailer Plugin		

\*\* - required dependency

Jenkins 2.19.4



# Create admin user

Getting Started

## Create First Admin User

Username:

Password:

Confirm password:

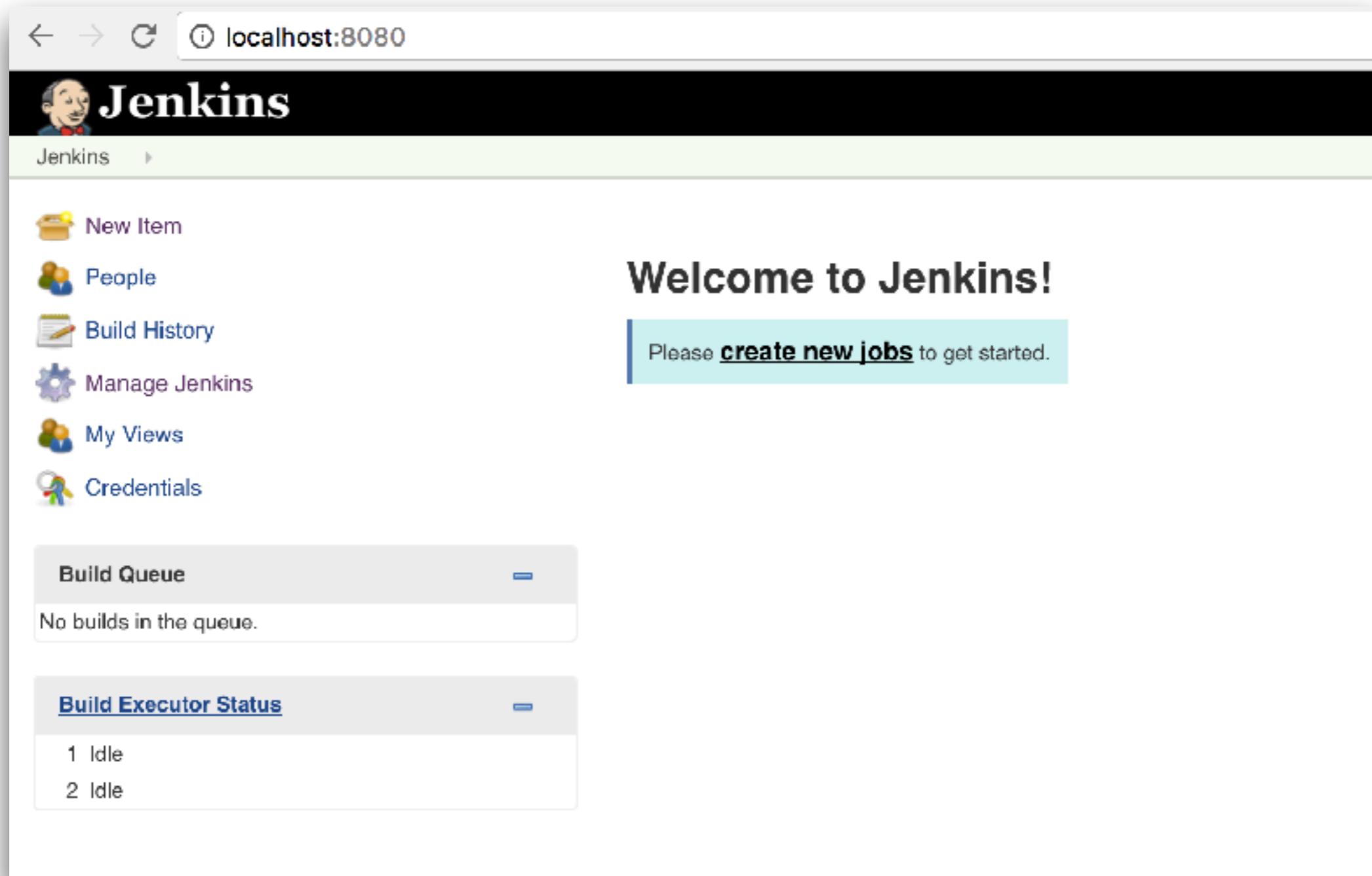
Full name:

Jenkins 2.19.4

[Continue as admin](#) [Save and Finish](#)



# Hello Jenkins

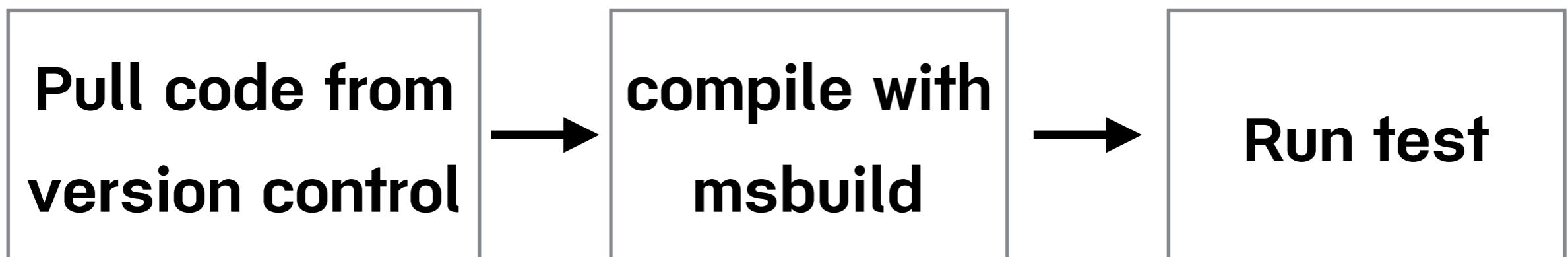


# Workshop Jenkins + C#



# Let's start with design

build process / build pipeline



# Create process with Jenkins

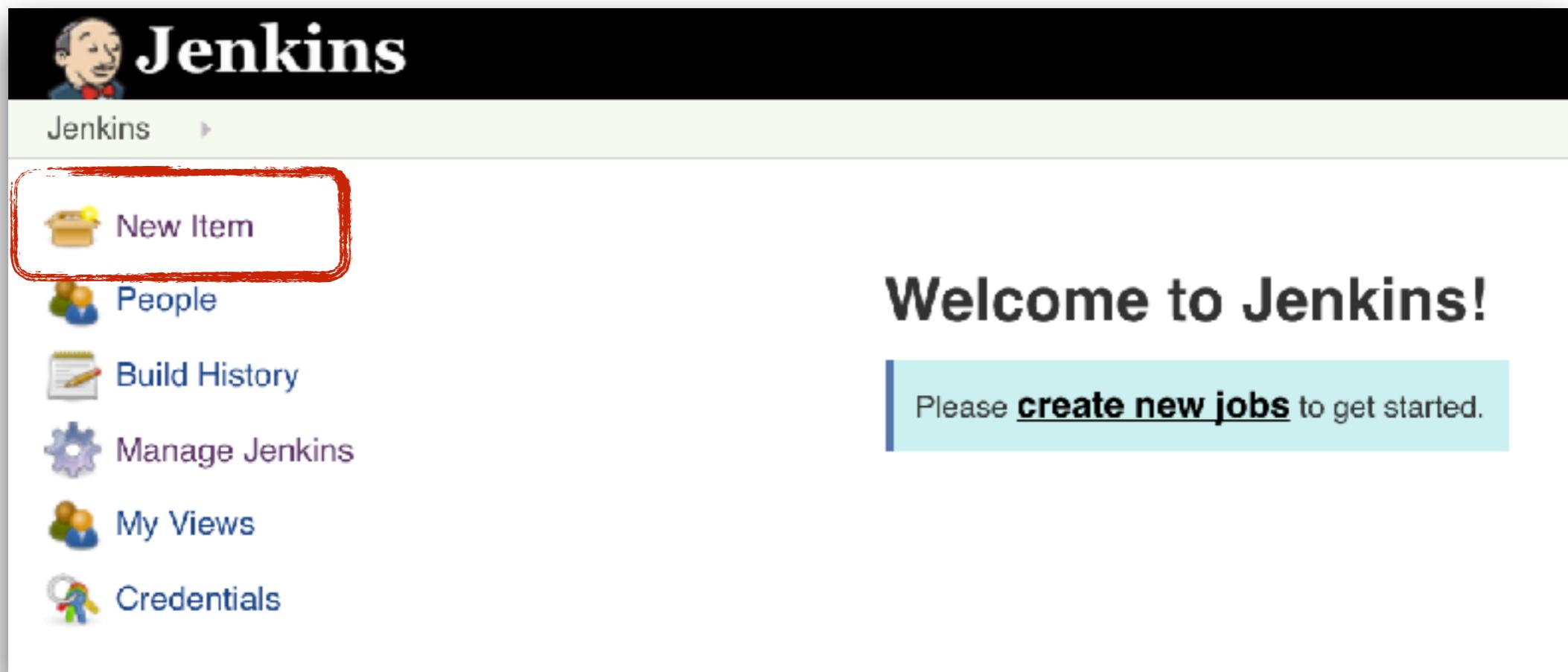


# 1. Pull latest code



# 1. pull latest code

create new item/job/project



# 1. pull latest code

Name of process

Enter an item name

PULL-CODE

» Required field

 **Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

 **External Job**  
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

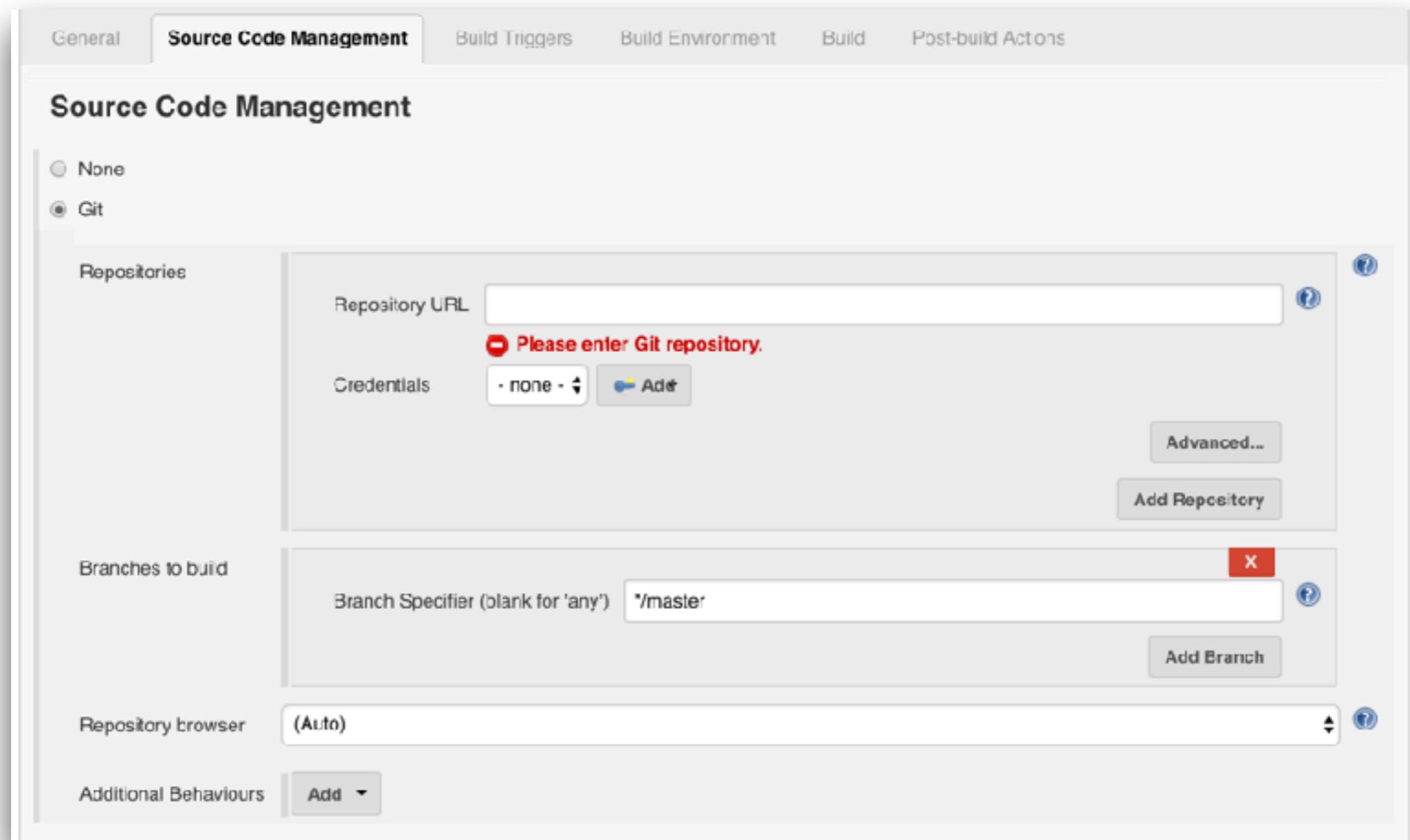
 **Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**OK**



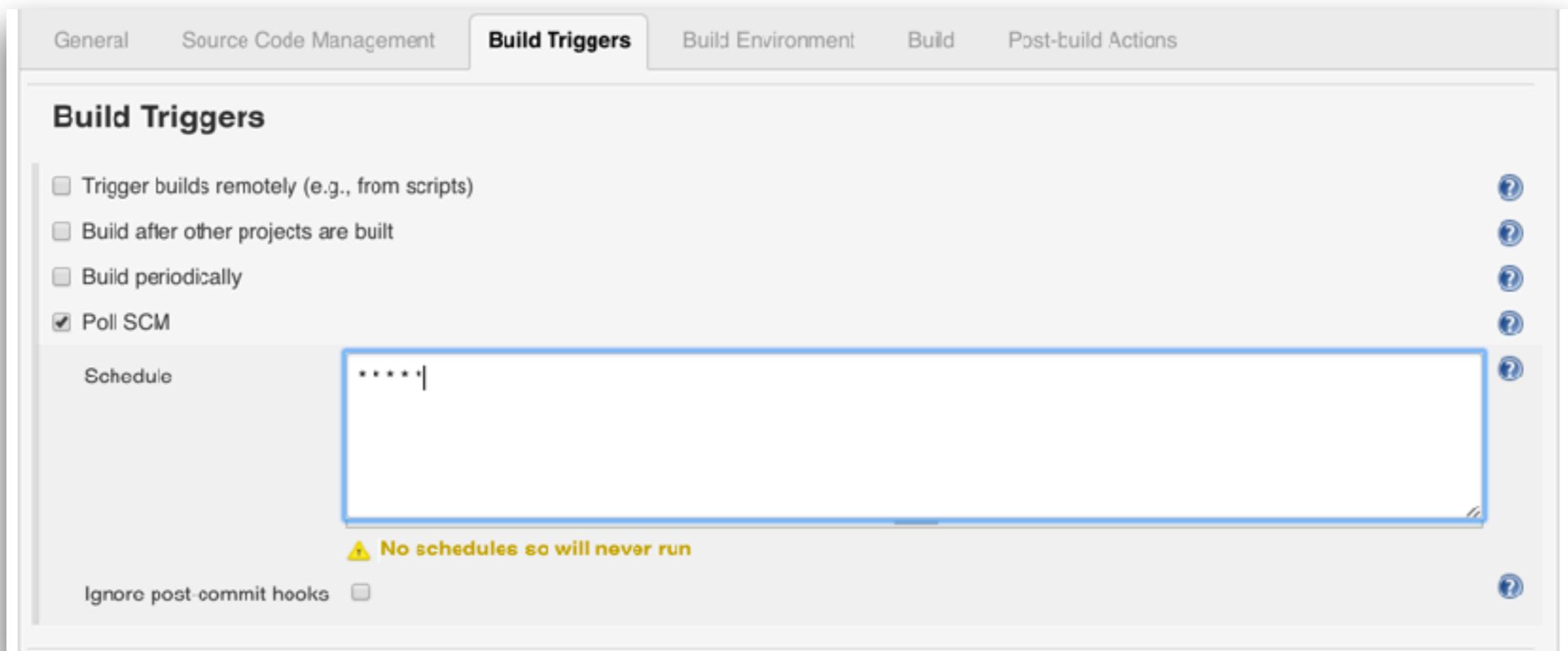
# 1. pull latest code

Choose version control (git)



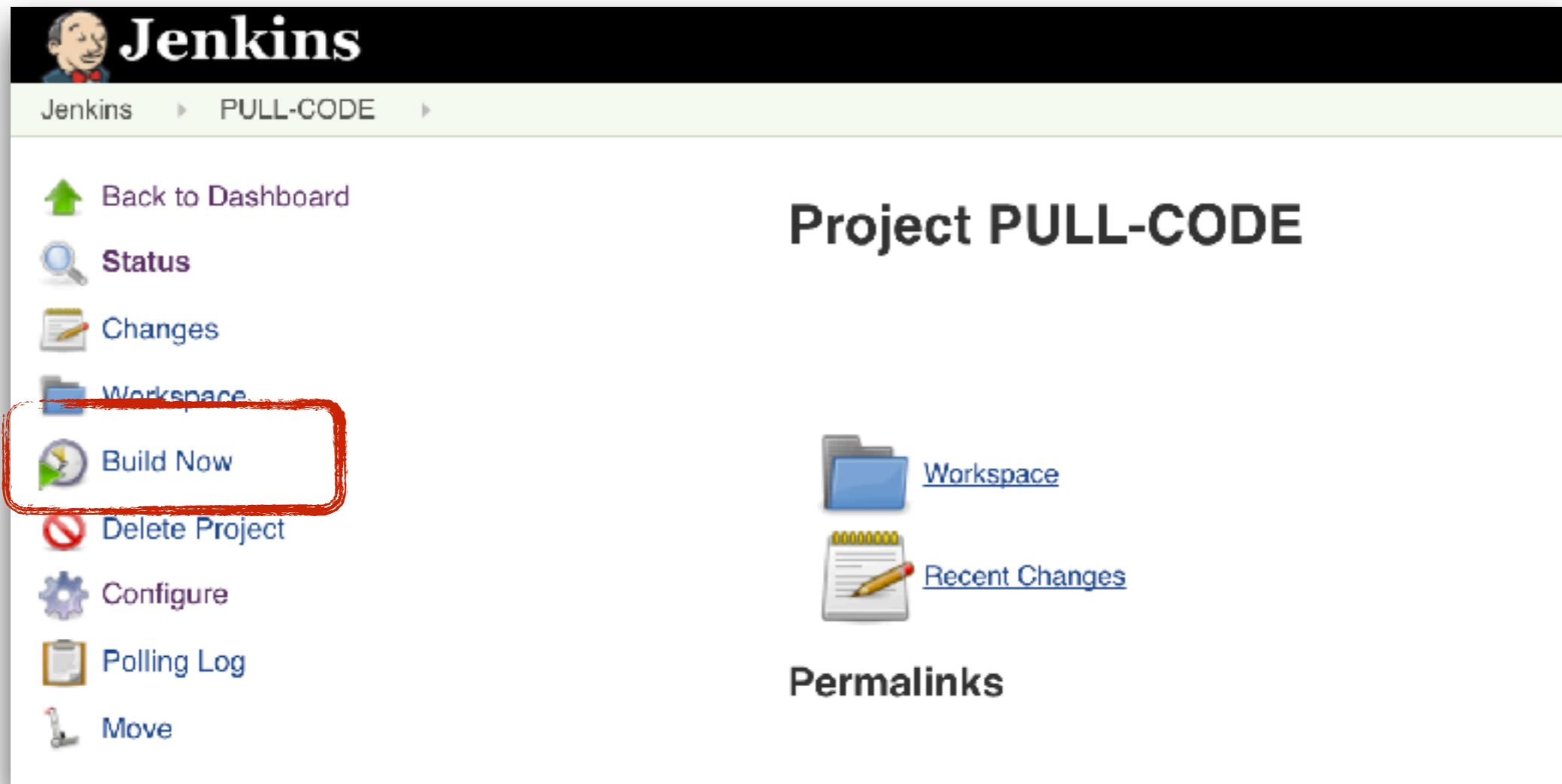
# 1. pull latest code

Config build trigger and save



# 1. pull latest code

Save and test configuration



# 1. pull latest code

See success result at workspace

The screenshot shows the Jenkins interface for the 'PULL-CODE' project. On the left, a sidebar menu includes 'Back to Dashboard', 'Status', 'Changes', 'Workspace' (which is highlighted with a red box), and 'Wipe Out Current Workspace'. Other options like 'Build Now', 'Delete Project', 'Configure', 'Polling Log', and 'Move' are also listed. Below this is a 'Build History' section showing a single build (#1) from Dec 12, 2016, at 9:21 PM. At the bottom are links for 'RSS for all' and 'RSS for failures'. The main content area is titled 'Workspace of PULL-CODE on master' and displays a file tree with various files and their details:

File/Folder	Size	Action
.git	1.73 KB	<a href="#">view</a>
bin/Debug	2.87 KB	<a href="#">view</a>
obj/Debug	502 B	<a href="#">view</a>
AddCommentProduct.cs	10.28 MB	<a href="#">view</a>
AddCommentProductDRY.cs	1.53 KB	<a href="#">view</a>
App.config	105 B	<a href="#">view</a>
chromedriver	2.74 KB	<a href="#">view</a>
chromedriver-license.txt	449 B	<a href="#">view</a>
ContactUsPage.cs	3.30 MB	<a href="#">view</a>
demo_ui_test.csproj	827 B	<a href="#">view</a>
demo_ui_test.csproj.user	7.00 KB	<a href="#">view</a>
geckodriver	747 B	<a href="#">view</a>
HelloSeleniumGrid.cs	392 B	<a href="#">view</a>
nunit-result.xml		
packages.config		
Test.cs		

At the bottom right of the workspace view, there is a link '(all files in zip)'.

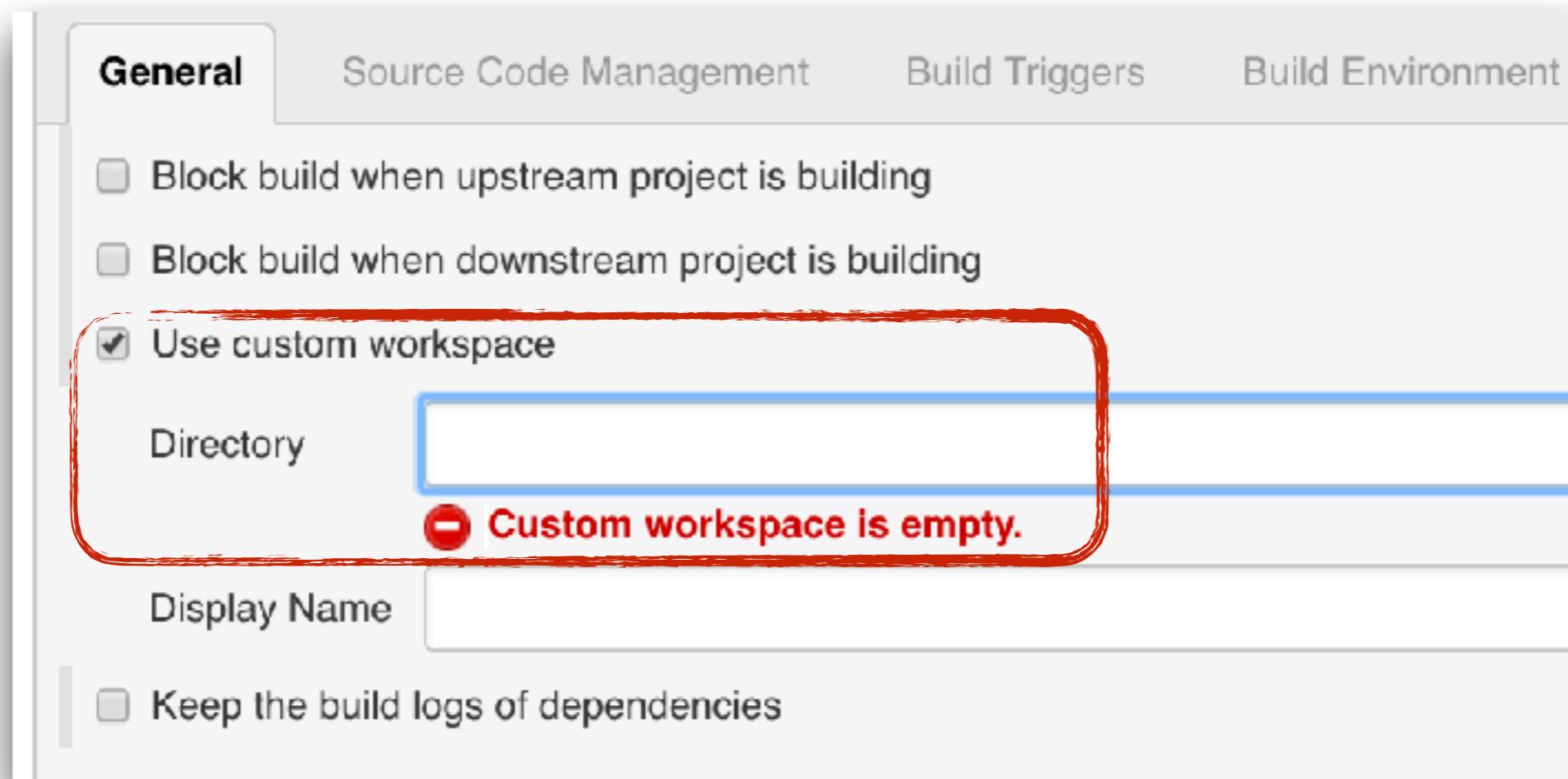


# Build and Testing



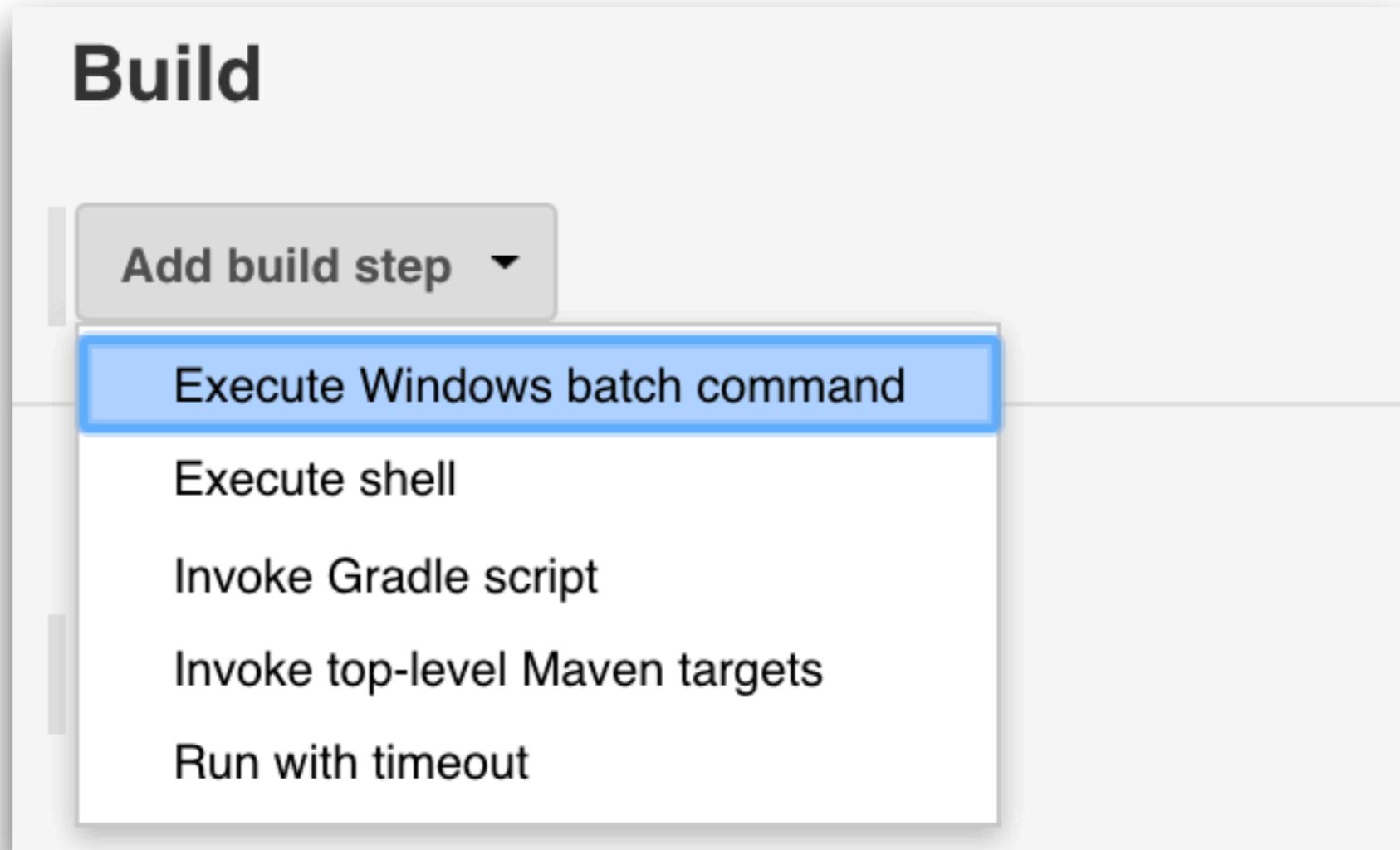
# 2. Build

## Use custom workspace



# 2. Build

Build with windows batch command



# 2. Build

## Build with msbuild command

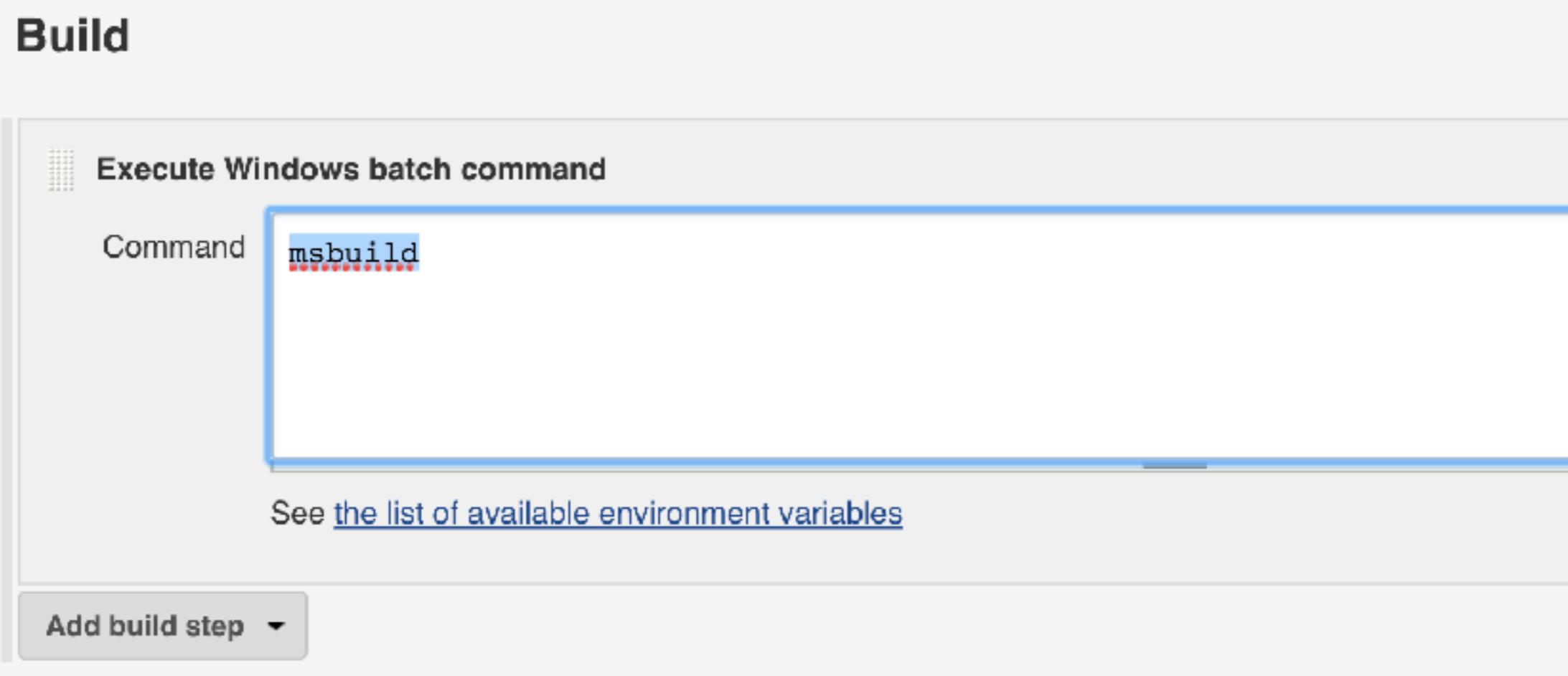
**Build**

Execute Windows batch command

Command `msbuild`

See [the list of available environment variables](#)

Add build step ▾

A screenshot of a software interface for building projects. The title bar says 'Build'. Below it, there's a section titled 'Execute Windows batch command'. A text input field contains the command 'msbuild', which is highlighted with a blue selection border. Below the input field, there's a link 'See the list of available environment variables'. At the bottom left, there's a button labeled 'Add build step' with a dropdown arrow.

# 3.1 Testing

## Testing with NUnit

**Build**

**Execute shell**

Command `nunit-console bin/Debug/demo_ui_test.dll -xml=nunit-result.xml`

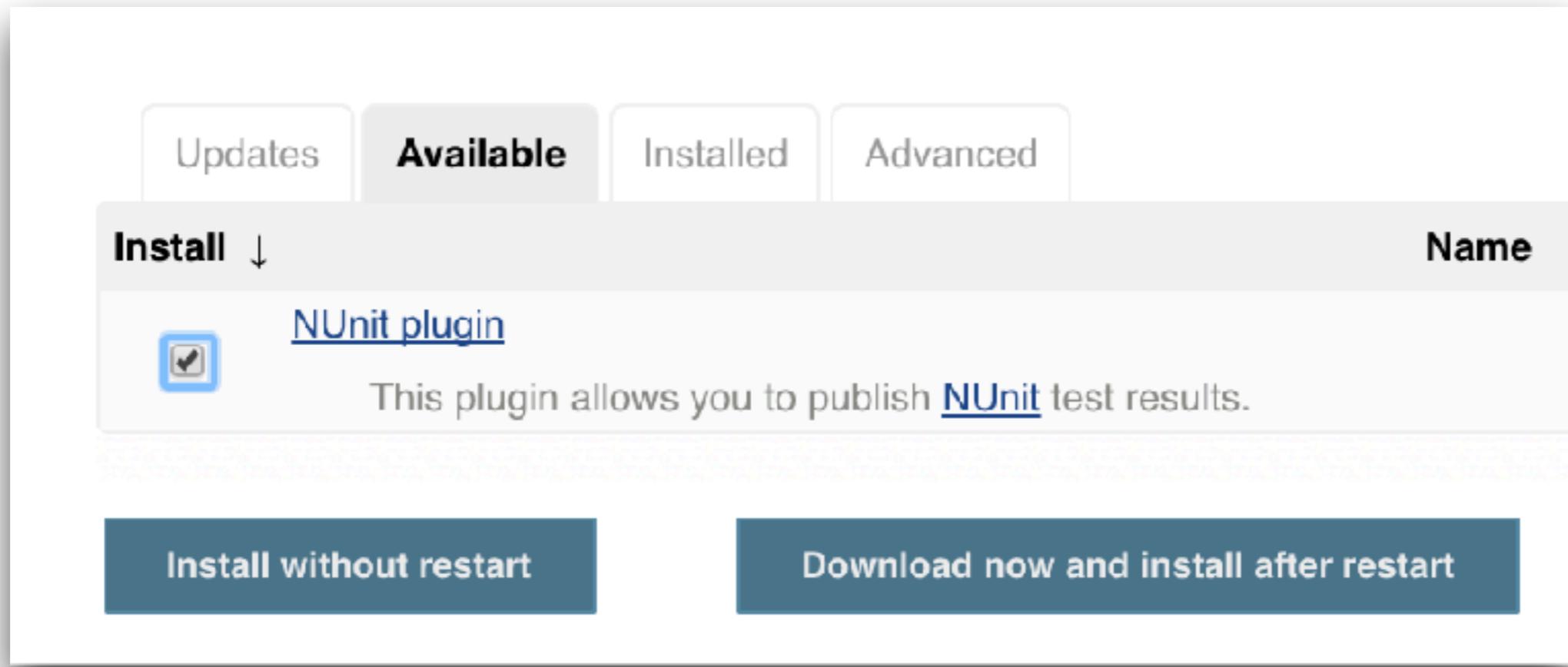
See [the list of available environment variables](#)

Add build step ▾



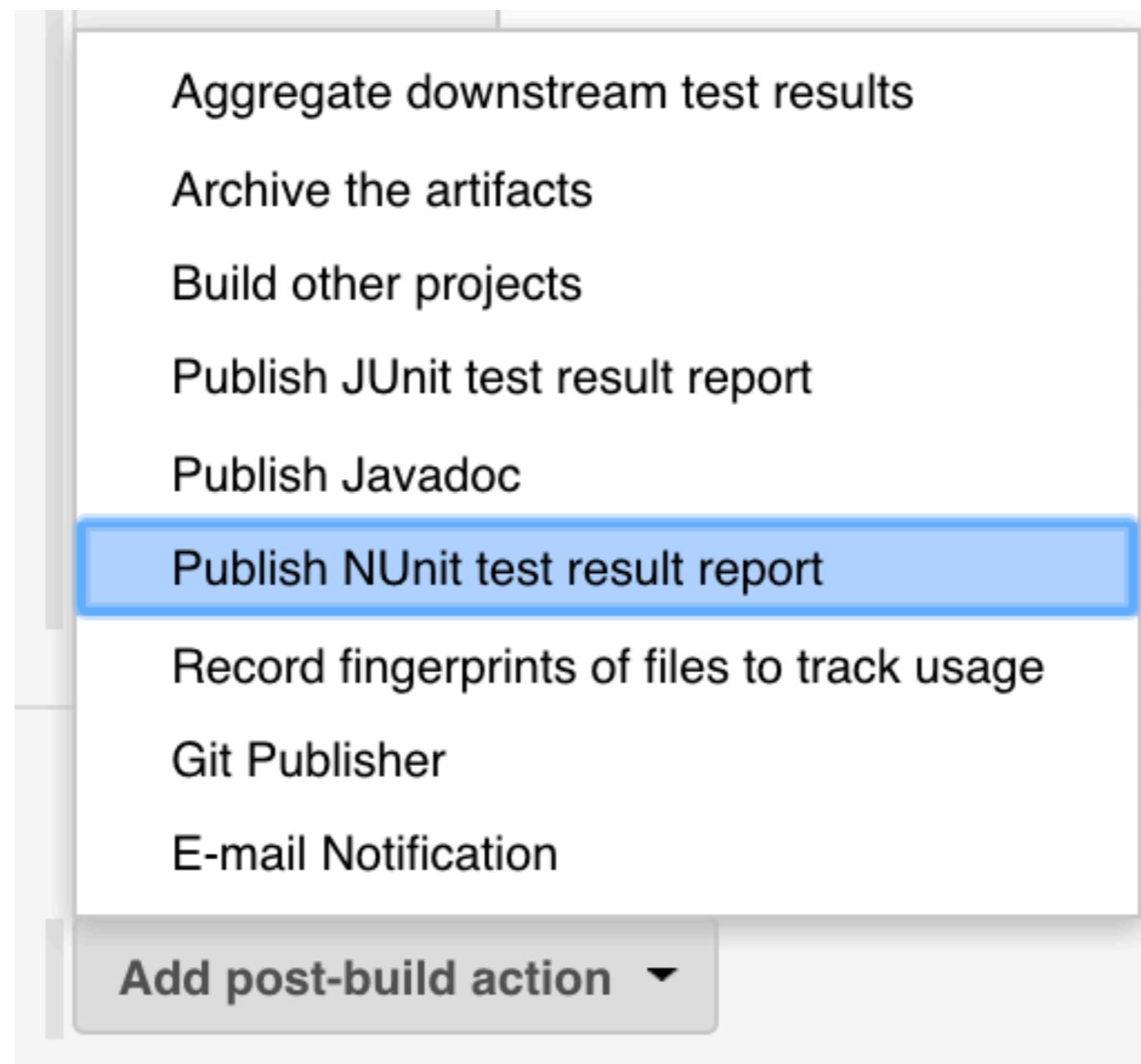
# 3.1 Testing

## Install nUnit plug-in



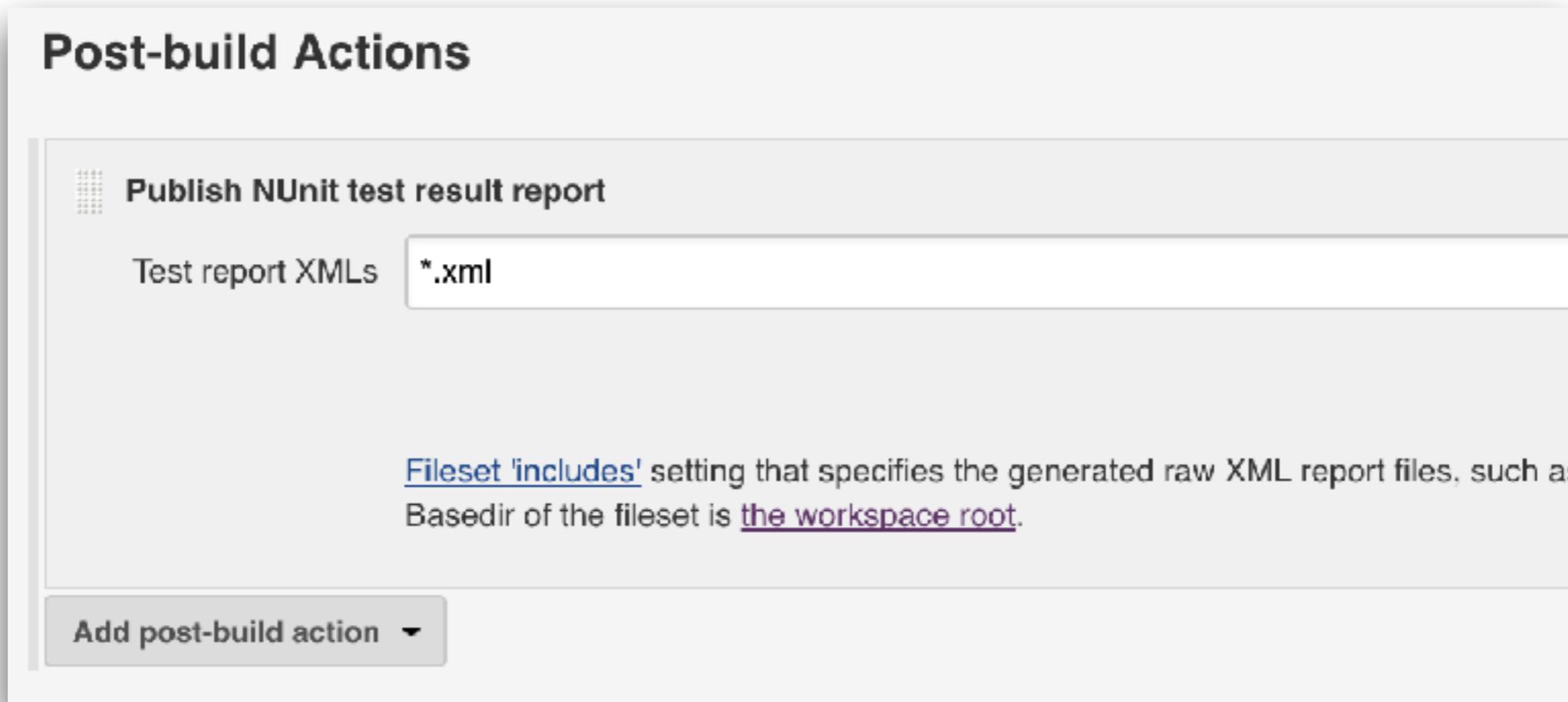
# 3.1 Testing

## Testing with NUnit + report



# 3.1 Testing

## Testing with NUnit + report



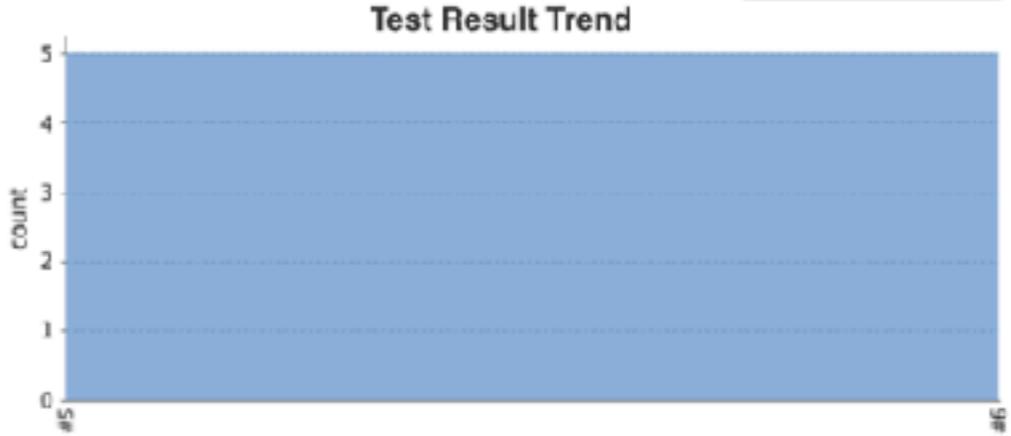
# 3.1 Testing

Save, Run and See report

**Project TESTING-NUNIT**

[add description](#)

[Disable Project](#)

 **Test Result Trend**  
count  
5  
4  
3  
2  
1  
0  
#5 (just show failures) enlarge

[Workspace](#)

[Recent Changes](#)

[Latest Test Result \(no failures\)](#)

**Upstream Projects**

[BUILD](#)

**Permalinks**

- [Last build \(#6\), 3 min 33 sec ago](#)
- [Last stable build \(#6\), 3 min 33 sec ago](#)
- [Last successful build \(#6\), 3 min 33 sec ago](#)
- [Last failed build \(#4\), 16 min ago](#)
- [Last unsuccessful build \(#4\), 16 min ago](#)
- [Last completed build \(#5\), 3 min 33 sec ago](#)



# 3.1 Testing

See detail report

**Test Result : demo\_ui\_test**

0 failures (±0)

5 tests (±0)  
Took 1 min 46 sec.  
[Add description](#)

**All Tests**

Class	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
<a href="#">AddCommentProduct</a>	45 sec	0		0		1		1	
<a href="#">AddCommentProductDRY</a>	52 sec	0		0		2		2	
<a href="#">HelloSeleniumGrid</a>	5.1 sec	0		0		1		1	
<a href="#">Test</a>	3.8 sec	0		0		1		1	



# 3.2 Testing

## Install VsTest Runner plug-in

The screenshot shows the Jenkins plugin manager interface. At the top, there are four tabs: Updates, Available (which is selected and highlighted in black), Installed, and Advanced. Below the tabs, there is a heading 'Install ↓' followed by a table.

Name
<a href="#">MSTest plugin</a>
<input type="checkbox"/> This plugin converts <a href="#">MSTest</a> TRX test reports into JUnit XML reports so it can be integ plugin converts the .coveragexml files found in the project workspace to the EMMA for <a href="#">plugin</a> or <a href="#">VsTestRunner plugin</a> to run the test and use this plugin to proce
<a href="#">VSTest Runner plugin</a>
<input checked="" type="checkbox"/> This plugin allow you to execute test using <a href="#">VsTest</a> command line tool.



# 3.2 Testing

## Testing with MS Test and VsTest.console

**Build**

Run unit tests with VSTest.console

VsTest Version (Default)

Test Files

Settings File

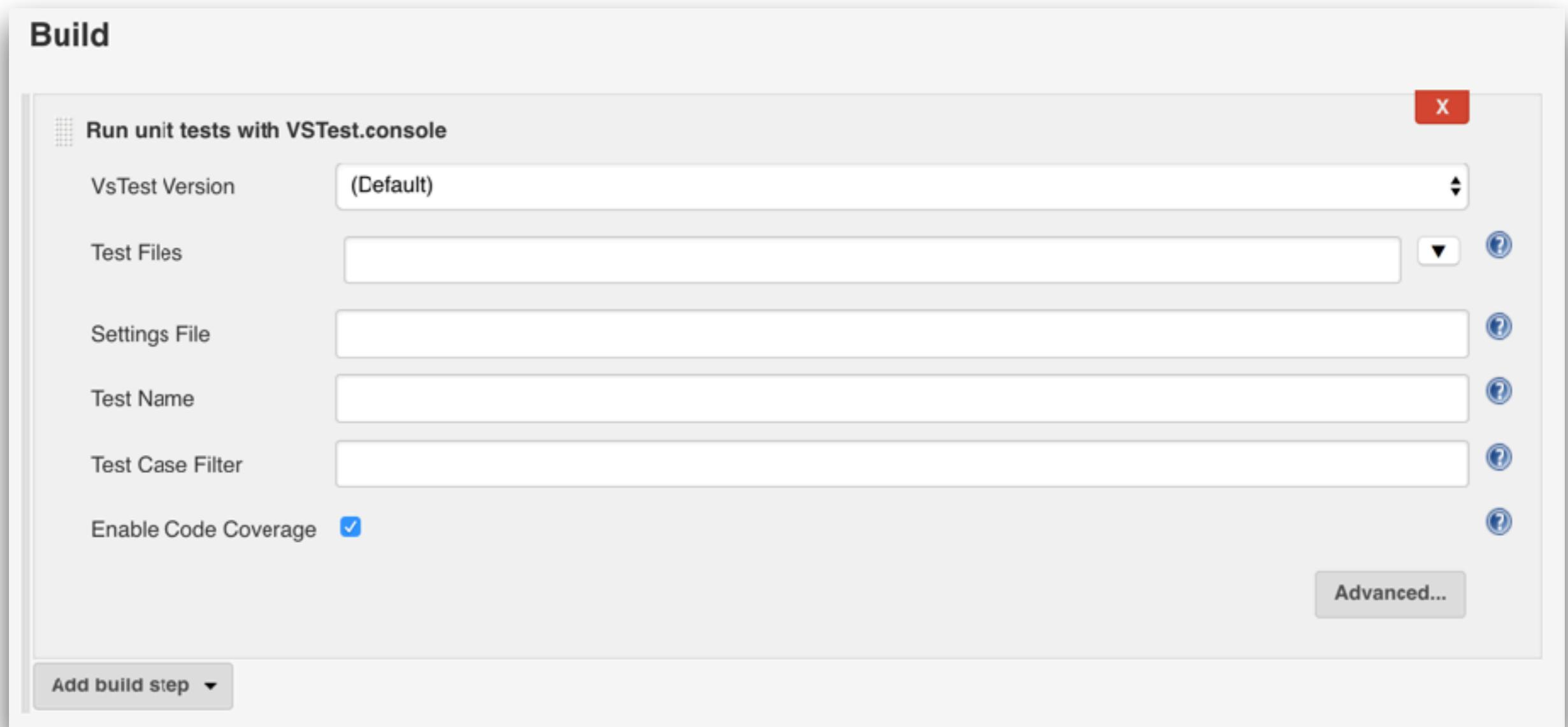
Test Name

Test Case Filter

Enable Code Coverage

Advanced...

Add build step ▾

A screenshot of a software interface showing a 'Build' configuration screen. At the top, it says 'Run unit tests with VSTest.console'. Below that, there are several input fields: 'VsTest Version' set to '(Default)', 'Test Files' (empty), 'Settings File' (empty), 'Test Name' (empty), 'Test Case Filter' (empty), and 'Enable Code Coverage' with a checked checkbox. To the right of each input field is a blue question mark icon. At the bottom right is a grey 'Advanced...' button. At the very bottom left is a button labeled 'Add build step ▾'.

# 3.2 Testing

## Install MS Test plug-in

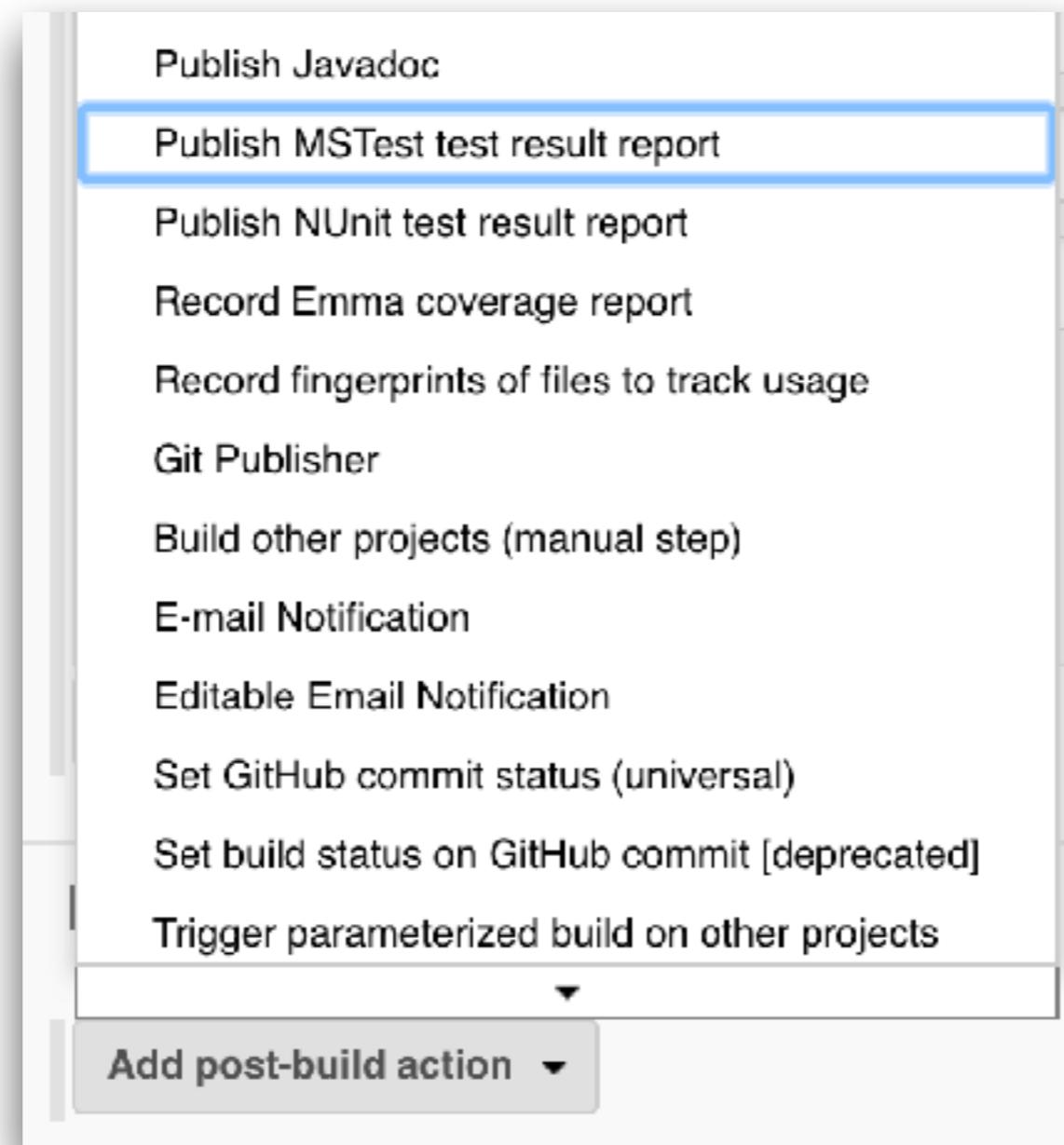
The screenshot shows the Jenkins plugin manager interface. At the top, there are four tabs: Updates, Available (which is selected and highlighted in black), Installed, and Advanced. Below the tabs, there is a heading 'Install' with a downward arrow. Two plugin entries are listed:

Name
<a href="#">MSTest plugin</a> This plugin converts <a href="#">MSTest</a> TRX test reports into JUnit XML reports so it can be integrated with . plugin converts the .coveragexml files found in the project workspace to the EMMA format. You can <a href="#">plugin</a> or <a href="#">VsTestRunner plugin</a> to run the test and use this plugin to process the results.
<a href="#">MSTestRunner plugin</a> This plugin allow you to execute test using MSTest command line tool.



# 3.2 Testing

## Testing with MS Test + report



# 3.2 Testing

## Testing with MS Test + report

**Post-build Actions**

**Publish MSTest test result report**

Test report TRX file  A path relative to [the workspace root](#), an Ant fileset pattern, or an environment variable.

Fail build if no files are found

Retain long standard output/error

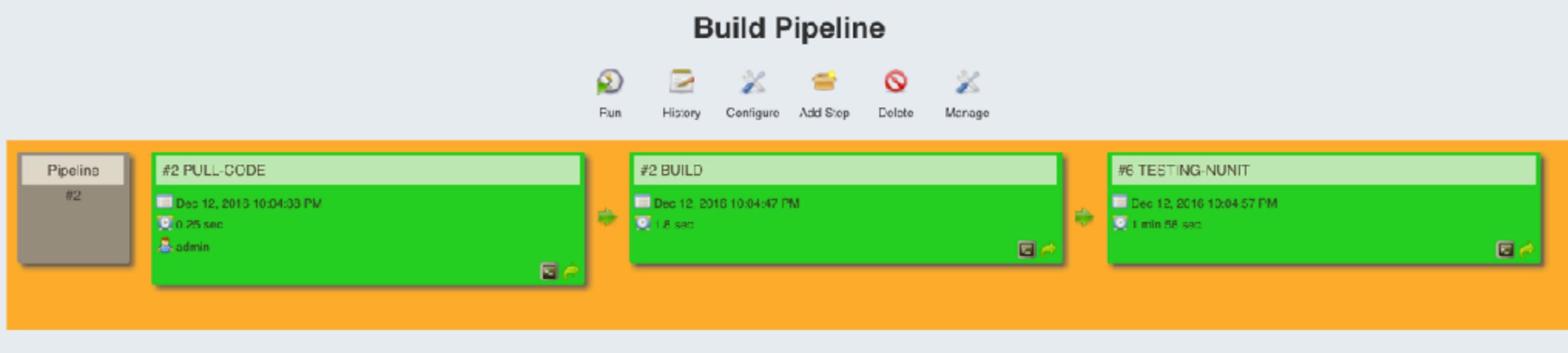
**Add post-build action ▾**



# Create build pipeline



# Create build pipeline



# Create build pipeline

View name **BUILD-PIPELINE**

**Build Pipeline View**  
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version prop

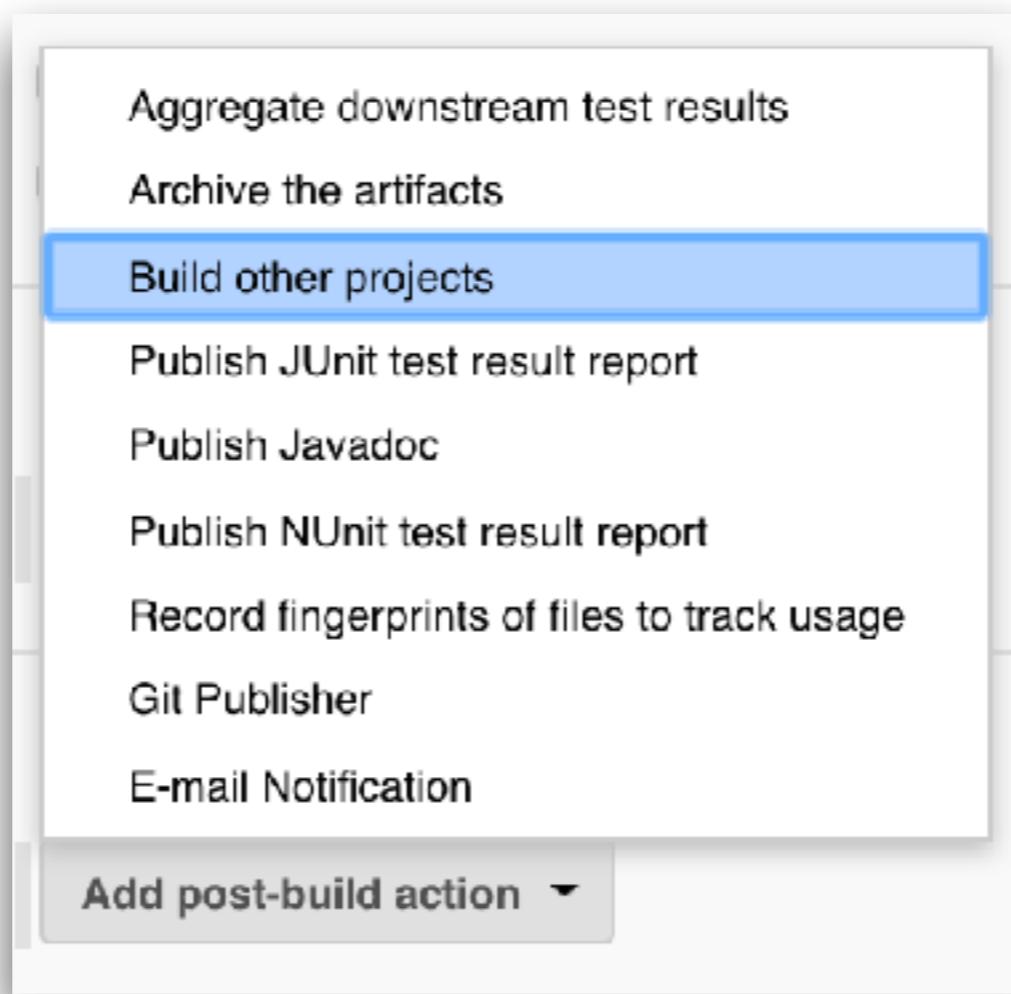
**List View**  
Shows items in a simple list format. You can choose which jobs are to be displayed in wh

**My View**  
This view automatically displays all the jobs that the current user has an access to.

**OK**



# Create build pipeline



# Create build pipeline

**Post-build Actions**

**Build other projects**

Projects to build **BUILD,** |

**No project specified**

- Trigger only if build is stable
- Trigger even if the build is unstable
- Trigger even if the build fails



# Create build pipeline

View name **BUILD-PIPELINE**

**Build Pipeline View**  
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version prop

**List View**  
Shows items in a simple list format. You can choose which jobs are to be displayed in wh

**My View**  
This view automatically displays all the jobs that the current user has an access to.

**OK**



# Create build pipeline

Name

Description

Filter build queue  [?](#)

Filter build executors  [?](#)

Build Pipeline View Title

Pipeline Flow

Layout  [?](#)

This is your main choice for pipeline structure based on the upstream/downstream relationship. It's the most out-of-the-box supported layout mode, but is open for extension.

Upstream / downstream  [?](#)

Select Initial Job  **PULL-CODE** [?](#)

TESTING-NUNIT [?](#)

Trigger Options

Restrict triggers to most recent successful builds  Yes  No [?](#)

Always allow manual trigger on pipeline steps  Yes  No [?](#)

Display Options

No Of Displayed Builds  [?](#)

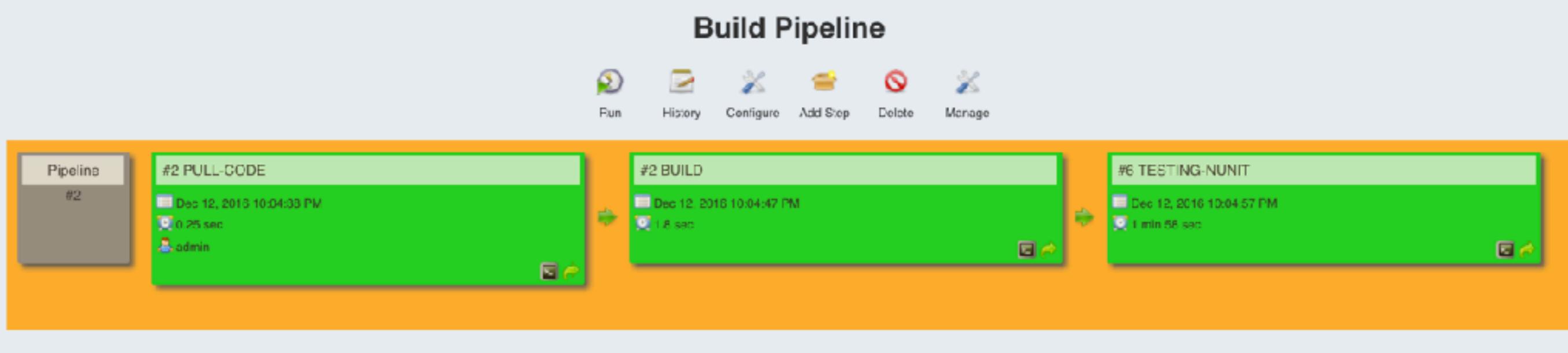
Row Headers  [?](#)  
Show just the build pipeline number

Column Headers  [?](#)  
Do not show any column headers

**OK** **Apply**



# Create build pipeline



# Back to basic



“What 's in a name ?”

That which we call a rose  
by any other name would smell  
as sweet

Romeo and Juliet



The ratio of time spent (code)  
versus writing is  
over 10 to 1

Robert C. Martin, Clean Code



# Naming convention

Namespace  
Class and method



# Guide to test writing

Don't say "**test**", say "**should**"



# Don't use the word “test”



# Use the word “should”

```
transferShouldDeductSumFromSourceAccountBalance()
```

```
transferShouldAddSumLessFeesToDestinationAccountBalance()
```

```
depositShouldAddAmountToAccountBalance()
```



# Guide to test writing

Don't test your class, test **behaviour**



# Guide to test writing

**Test class names** are important too

**Structure** your test well

Tests are **deliverable** too



**Now, go write some tests !!**





**“Quality is the responsibility of everybody.”**

