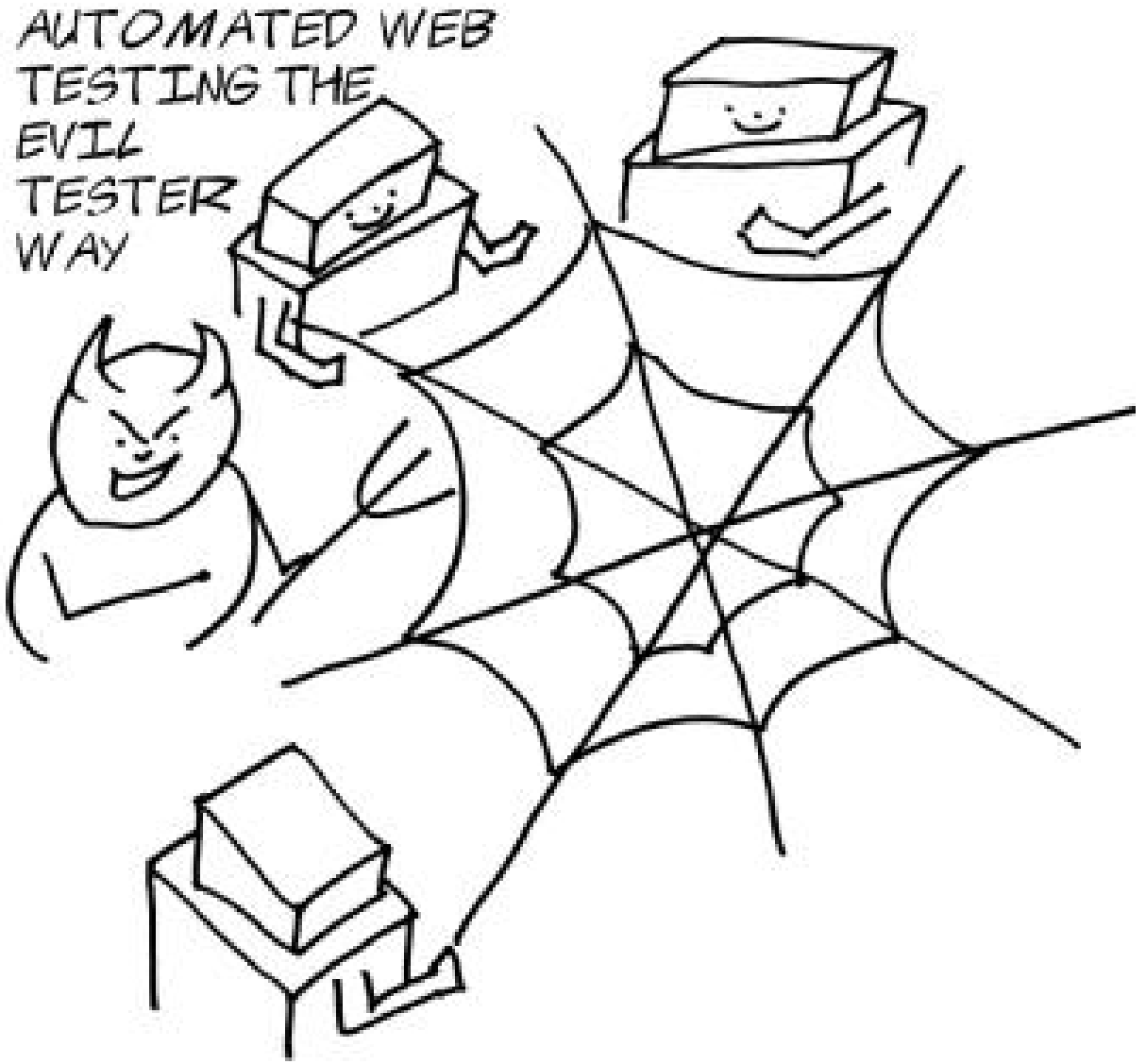
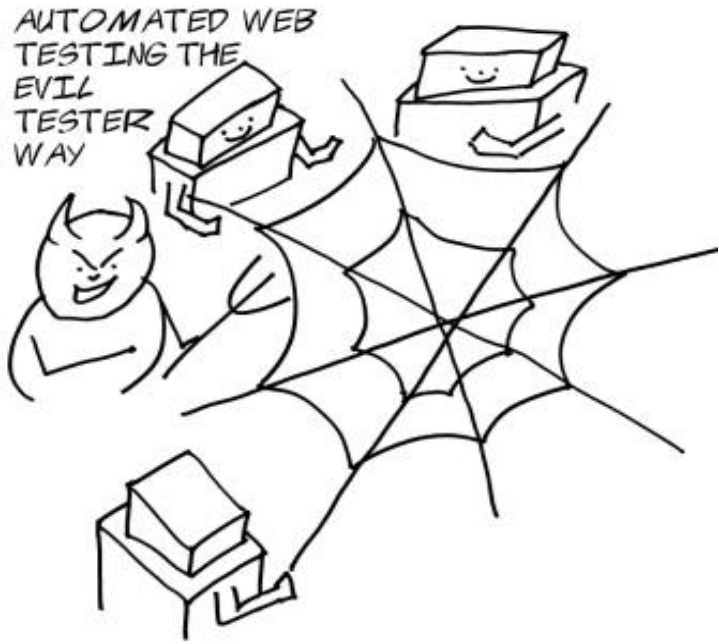
Selenium 2 : WebDriver Basics

Alan Richardson



Copyright

Continuous Integration



CI Described

- Continuously monitor the code base and when it changes
  - Build it, Test it, Report on Status
- Periodically build the code e.g. Nightly

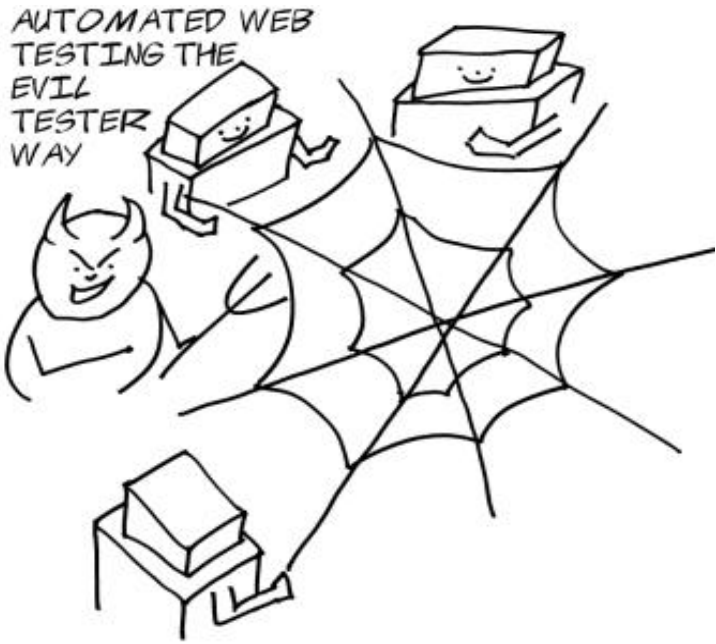
CI References

- References
  - http://martinfowler.com/articles/continuousIntegration.html
  - book Continuous Delivery https://continuousdelivery.com/
  - http://en.wikipedia.org/wiki/Continuous_integration

Continuous Integration Tools

- Lots of Continuous Integration Tools
  - https://xebialabs.com/the-ultimate-devops-tool-chest/continuous-integration/
  - e.g.
    - Jenkins - http://jenkins-ci.org/
    - JoliCI - https://github.com/jolicode/JoliCi
    - Travis-ci - https://travis-ci.org/
    - https://www.go.cd/

CI In Action Examples



Example Code for CI

- https://github.com/eviltester/wdci
  - A subset of the course source code
  - Freely available on github
  - Amended to run without error on HtmlUnit, Firefox, IE, etc.
  - You will need to configure paths to run against IE, ChromeDriver, Marionette etc.

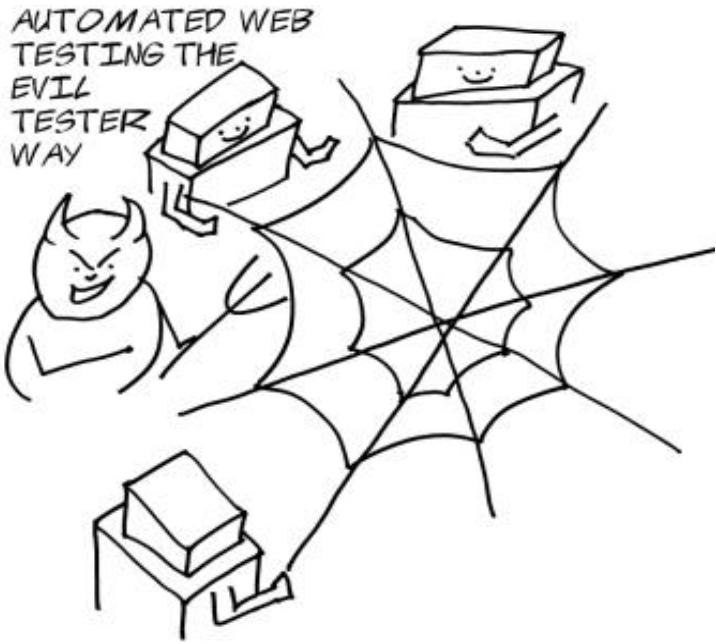
What does CI look like?

- Demo:
  - Manually trigger a build using Jenkins
- Using
  - https://github.com/eviltester/wdci

What does CI look like?

- Demo:
  - Automatically trigger a build using travis-ci
- Using
  - https://github.com/eviltester/wdci

Run from Command Line First



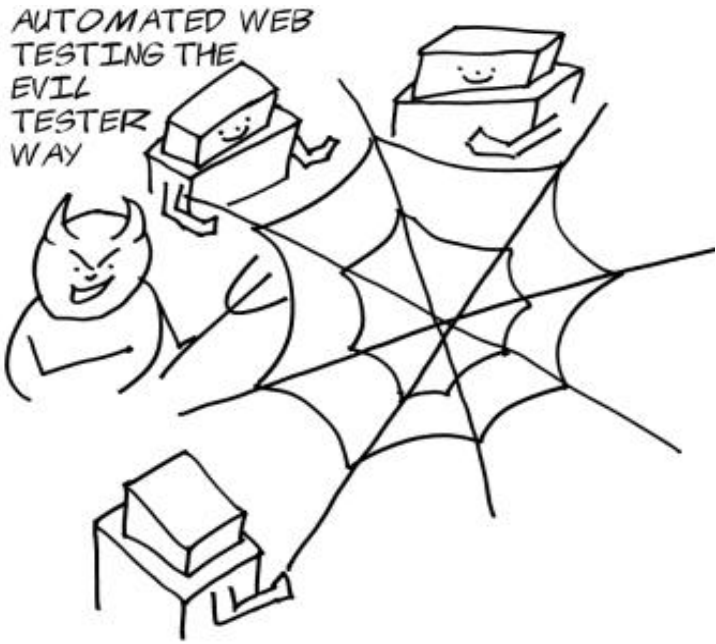
Secret to Using CI...

- If you can run your tests from the command line, you can run them from CI
  - e.g. `mvn test`
  - e.g. `mvn test -Ddriver=IE`
- Run from command line on build machine to ensure %PATH% setup correctly.

To use wdci from CLI

- Your commands run from the checkout root
- `pom.xml` is in the code folder, not root
- `mvn -f code\pom.xml test`
- `mvn -f code\pom.xml test -Dselenium2basics.webdriver=FIREFOX`
- `mvn -f code\pom.xml test -Dselenium2basics.webdriver=IE -Dwebdriver.ie.driver=<pathToYour>\IEDriverServer.exe`

Jenkins



Jenkins

- Keep it simple to start with
  - https://jenkins.io
  - Download war file
    - Also has native package for most operating systems
  - Copy to directory we want to run from
  - Run
    - `java -jar jenkins.war`

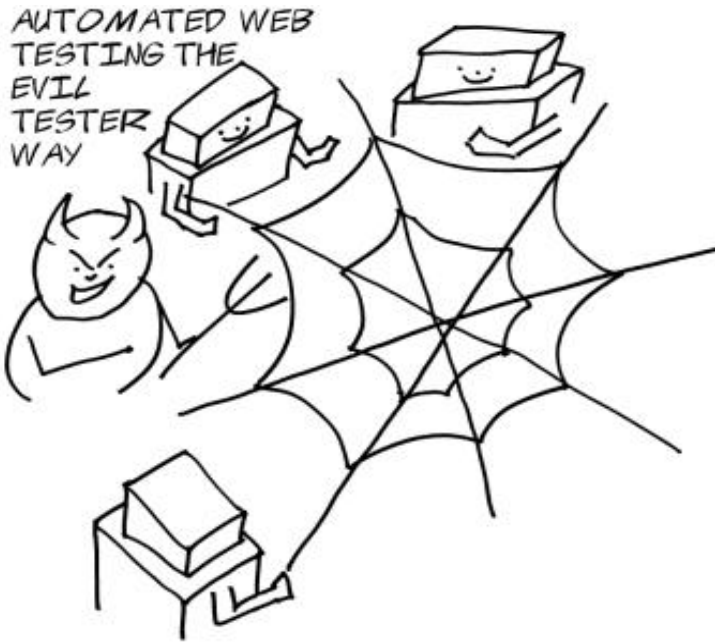
Install Recommended Plugins

- During initial startup allow Jenkins to install recommended plugins
  - This will provide git, & svn integration
- Use `Manage Jenkins` \ `Manage Plugins` to install
  - Maven Integration Plugin

Configure Global Tool Configuration

- Use `Manage Jenkins` \ `Global Tool Configuration` to setup
  - JDK
  - Maven
  - Git
- It might pick these up automatically if you have JAVA_HOME, MAVEN_HOME and if git.exe is on your path

Version Control: Git



Multiple ways to install

```
C:\Windows\System32>git --version
git version 2.6.3.windows.1
```

- https://git-scm.com/
  - Official Git site
  - https://git-scm.com/downloads
- https://help.github.com/
  - https://help.github.com/articles/set-up-git/
  - Install 'GitHub Desktop'
    - You might still have to add git.exe to your path
- https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html

Exercise: Create a job in Jenkins

- Create `New Item`
  - Name: `githhub wdci firefox`
  - Use `Maven Project`
- From Git repo
  - `https://github.com/eviltester/wdci`
- Build
  - Root Pom: `code/pom.xml`
  - Goals and Options: `test`
- Save
- Trigger a build manually

Exercise: Create a job using HTMLUnit

- Create `New Item` based on `github wdci firefox`
- Goals and Options:
  - `test -Dselenium2basics.webdriver=HTMLUNIT`

Exercise: Create a job using a different browser

- Create `New Item` based on `github wdci firefox`
- Goals and Options:

```
`test -Dselenium2basics.webdriver=<D>` -D<P>=<PATH>
```

<D> Driver Name	<P> Property	<PATH> to .exe
FIREFOXMARIONETTE	webdriver.gecko.driver	…/wires.exe
IE	webdriver.ie.driver	.../IEDriverServer.exe
GOOGLECHROME	webdriver.chrome.driver	.../chromedriver.exe

Exercise: Create your own repo

- Choose Github or Bitbucket
  - Bitbucket allows free 'private' repos
- Follow the instructions to create an account
- Create your own repo
- Add your code to the repo
- Check in your code
- Run your repo code from your local jenkins

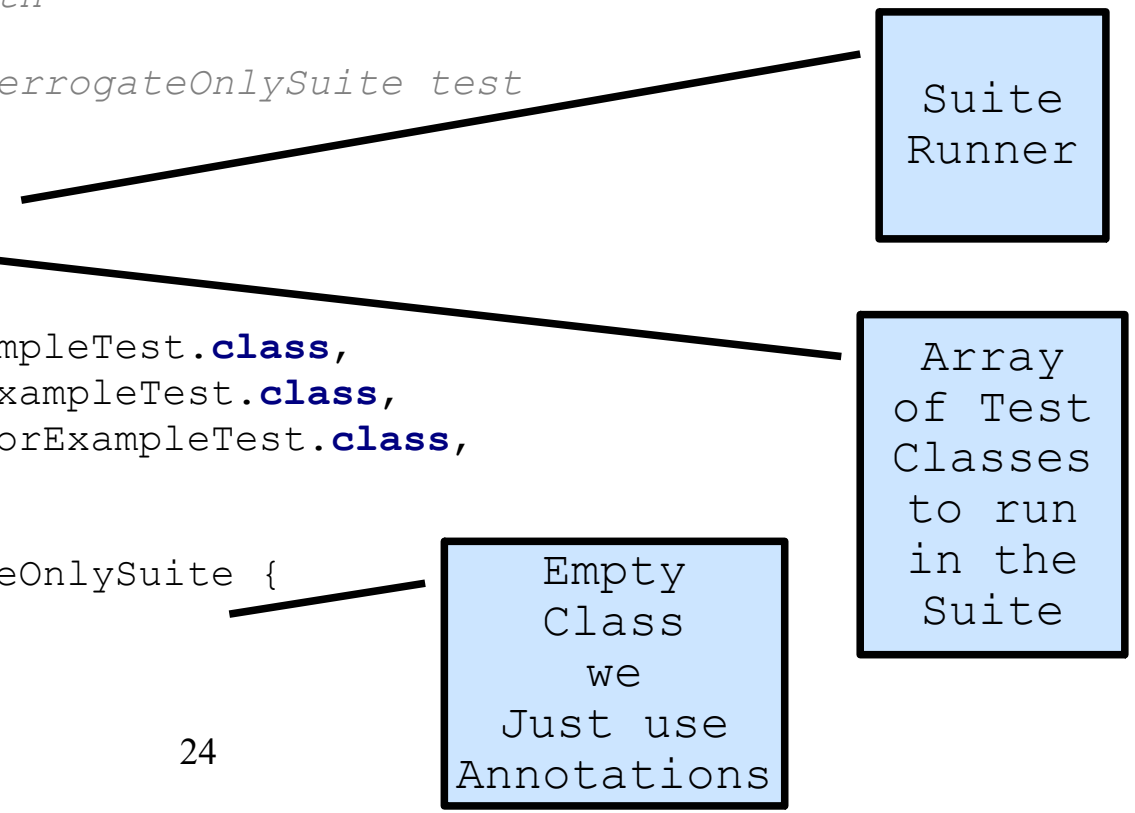
Junit Suites

- github.com/junit-team/junit4/wiki/aggregating-tests-in-suites

```
/**
 * A simple suite that contains a few Interrogation tests
 * as an example of how to collate specific Test Classes into
 * a Suite
 *
 * Can run the suite with
 *
 * mvn clean -Dtest=InterrogateOnlySuite test
 *
 */
@RunWith(Suite.class)
@Suite.SuiteClasses({

        AFirstFindByExampleTest.class,
        ChainingFindByExampleTest.class,
        FindByCssSelectorExampleTest.class,

})
public class InterrogateOnlySuite {
}
```



How to use

- Can run from IDE
- Can run from command line

```
mvn clean -Dtest=InterrogateOnlySuite test
```

- Can run from CI

Why?

- Organise Test classes into groups
  - e.g.
    - Android Only Tests
    - Tests that fail on firefox
    - Tests that only run with a proxy
    - Intermittent And Require Investigation
    - etc.

Maven Profiles

- In the pom.xml we can add a <profiles> section

maven.apache.org/guides/introduction/introduction-to-profiles.html

- We can control which test methods are run using include and exclude in the surefire plugin

maven.apache.org/surefire/maven-surefire-plugin/

A simple profile to run Suites

```
<profiles>
    <profile>
        <id>IMtests</id>
        <build>
            <plugins>
                <plugin>
                    <groupId>org.apache.maven.plugins</groupId>
                    <artifactId>maven-surefire-plugin</artifactId>
                    <version>2.14.1</version>
                    <configuration>
                        <includes>
                            <include>**/InterrogateOnlySuite.class</include>
                            <include>**/ManipulateOnlySuite.class</include>
                        </includes>
                    </configuration>
                </plugin>
            </plugins>
        </build>
    </profile>
</profiles>
```

How to run

mvn test -PIMtests

- Run the suites defined in profile IMtests

mvn clean -PIMtests -Dselenium2basics.webdriver=HTMLUNIT test

- Run the suites defined in profile Imtests using the HTMLUNIT browser

Exercise: Creating Suites & Profiles

- Try to do this on your @Test code
- Run each test class against each driver (by changing the driver)
- Identify any cross browser failures
  - can they be fixed easily or
  - are driver limitations e.g. HtmlUnit getText
- Add the @Test classes into appropriate suite e.g.
  - AllBrowsers, IEOnly, FailOnFirefox etc.
- Create profiles which allow you to run reliably on the different browsers
- Run profiles and suites from Jenkins

Continuous Integration Cross Browser Testing

- Older set of videos
- Discuss changes made to course source for cross browser and CI testing

Continuous Integration in Practice

- Older set of videos
- Show more detailed configuration of Jenkins jobs in maven for scheduled times
- Examples are using SVN
- Also discussion of test intermittency