**Presentation Notes**

**==================**

**Title:**

Automated testing update

**Slide 1:**

“If you don’t like testing your product, most likely your customers won’t like to test it either.” (Anonymous)

**Slide 2:**

Our testing challenge

1. Regression testing

2. Unit testing

3. Monitoring

4. Continuous Integration

- these are all separate things, but they are part of the same thing really

- the result of this is a way to rapidly assurance of code, web pages, functions, digital artefacts

**Slide 3:**

Willy Wonka

"By all means, tell me more about this 'Regression Testing'"

**Slide 4:**

Squiz Matrix Regression Test

- 163 tests in 35 categories

- takes about 3 hours to execute manually

- a live test plan -> more tests added constantly

- doesn't scale well

- requires familiarity with Squiz Matrix, to keep the test plan documentation simple

**Slide 5:**

Boromir

"One does not simply do regression testing in one day"

- the Squiz Matrix regression test takes less than a day to complete, but needs to

- resource the testing session

- compile test documentation to record results

- imagine trying to execute Squiz Matrix, Wordpress, Confluence and other regression tests in one day?

**Slide 6:**

Selenium IDE

- Selenium IDE tool quick demo -> a front end testing tool

- Selenium IDE isn't the only one (there is GhostInspector service and others)

SHOW THE MATRIX TESTS IN THE WINDOW

- show the failed test -> b/c review dates are currently disabled

- show the conventions we have imposed to help with robustness

- file naming conventions

- Test IDs,

- correlating test IDs with the test plan -> test plan is source of truth as some tests can't be automated

- show the logging

- show the ability to record tests without knowing code

**Slide 7:**

Chuck Norris

"Chuck Norris can unit test entire applications with a single assert"

- Sadly, we don't have Chuck Norris to help us.

- Any test suite is going to have numerous test cases.

**Slide 8:**

Selenium IDE test practices

- Idempotent test cases

- Watch out for AJAX / asynchronous functions

- parameterise where you can (makes test maintenance easier)

- for robustness, replace ids used by the recorder tool with more unique ones

- needs ability to navigate page markup

**Slide 9:**

Future steps: Higher power testing with Selenium Server (or any test framework)

- One application of this tool is to create test cases and export them to a Selenium server

- A Selenium server can test multiple cross-browsers

- Selenium tests are stronger and use stronger programmatic logic

- But you need to be able to code, so only good for technically minded devices

- Many language options, so you can code Selenium tests in php, javascript, java, ruby etc.

**Slide 10:**

Continuous Integration Roadmap

- I guess the holy grail is to have a continuous integration server (what form that takes will need investigation)

- I've played with Jenkins with Selenium Server, written in Java, but could be others.

SHOW THE JENKINS WINDOW

- Hooks the tests into the CI server

- the CI server can pull code down from a repository, run a grunt task or tasks, then execute the tests

- keeps a score on what tests have passed or failed

**Slide 11:**

Beer guy

"I don't always test my code, but when I do, I do it in production"

- Don't do what the beer guy does

- Final note: Imagine these scenarios:

- releasing a new version of Squiz Matrix the day after it is released by the vendor

- minimal time between iterations and release b/c testing is automatic upon commit