Behaviour Driven Development

Delivering Zero Known Defect Releases

v1.0 June 2015
BDD: Delivering Zero Known Defect Release

Capita IT Professional Services

Colin Deady
Technical Test Manager

colin.deady@capita.co.uk
07500 917434
@ethicalwebsites
Software should just work
We should have fun while making software work
Feature: Confirm Zumo George is alive (ExplorerPro is attached and working)
In order to confirm that Zumo George is alive and well
As George's zoo keeper
I want to run some initial tests

@zumotest
Scenario Outline: Verify lights are responding
  Given the lights are all off
  When the <colour> light is switched <state>
  Then the <colour> light should be <state>

Examples:
<table>
<thead>
<tr>
<th>colour</th>
<th>state</th>
<th>expected result</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>on</td>
<td>on</td>
</tr>
<tr>
<td>red</td>
<td>off</td>
<td>off</td>
</tr>
<tr>
<td>green</td>
<td>on</td>
<td>on</td>
</tr>
<tr>
<td>green</td>
<td>off</td>
<td>off</td>
</tr>
<tr>
<td>blue</td>
<td>on</td>
<td>on</td>
</tr>
<tr>
<td>blue</td>
<td>off</td>
<td>off</td>
</tr>
<tr>
<td>yellow</td>
<td>on</td>
<td>on</td>
</tr>
<tr>
<td>yellow</td>
<td>off</td>
<td>off</td>
</tr>
</tbody>
</table>

@ir @zumotest
Scenario: Verify infrared range finder is responding
  Given the distance to an object is greater than 10 cm
  When I read the infrared range finder
  Then I should see a distance greater than 10 cm

3 features (3 passed)
9 scenarios (9 passed)
27 steps (27 passed)
What is Behaviour Driven Development?
What is Behaviour Driven Development?

A leap of faith:
Analysis, Development and Testing are the same, not separate
And now….

Why did we choose BDD?
Why did we choose BDD?

3 major releases: “OK” software

4th release: Potential for penalties

Must: rock-solid, and extendable

Included a complex state machine

Must: 100% regression tests run

Should: reduce AT duration
Why did we choose BDD?

1500 scenarios  PASS
All under Continuous Integration
<5 minutes to execute
Zero Known Defects at point of release
And now....

Zero Known Defects
BDD with Zero Known Defects

We thought:

- Why stop at passing all scenarios?
- Wouldn’t it be great if we could also fix every defect?
  (e.g.: found during exploratory testing)

We realised:

- As Testers, Developers and BAs we can choose to do so
  (and it is a choice)
BDD with Zero Known Defects

IMPORTANT:

- **ZKD** is not a rehash “Zero Defects” from the 1960s.

**ZD:** “…**fastest ways to kill motivation** is what is called in the US Army a zero defects mentality… an atmosphere that tolerates absolutely no mistakes; perfection is required…”

Lean Software Development by Mary and Tom Poppendieck
(http://www.targetprocess.com/blog/2009/03/zero-defects-are-you-kidding-me.html)
BDD with Zero Known Defects

Why do we choose to do so?

- Expectations: “it should just work”
  (compare s/w to buying a car)
- We want to minimise re-work due to failure
- As engineers we want to be proud of our work
- As customers we want to look forward to what is delivered
BDD with Zero Known Defects

About 1.5 years ago my team decided:

we will **attempt to fix** every outstanding known defect by the end of each day

we will **guarantee to fix** every outstanding known defect by the end of the week and always before delivery

we will **never allow the open defect count to go above three** at any one time

*We chose these conditions as something we wanted to achieve*
BDD with ZKD – What did we get (and still have)

Business benefits:
- Clean builds: 100% BDD scenario pass
- ZKD prior to release to production
- No defect triage meetings needed
- Automatic ongoing regression testing for no extra cost
  (this IS important)

Plus:
- Happy Developers, Business Analysts & Testers
- Happy Clients
BDD with ZKD: what happened

How did we get to a position of zero known defects?

- Inherited defects from previous
- Drew line in sand: fixed all outstanding
- Team voluntarily committed to 3 guidelines
- Project Manager ensured time to achieve
- First release over 1.5 years ago, ZKD added
- Eight subsequent releases all with ZKD
BDD with ZKD – what have we learnt?

**BDD and ZKD can be modular** - some projects will start with just requirements analysis, others will automate regression testing

**ZKD is driven by the team**, with management support

**BDD with ZKD means no more compressed delivery schedules** requiring high contingency levels and expensive overtime

**Senior managers gain confidence** in BDD with ZKD as they start to see success in its use.

**ZKD becomes addictive** - 1.5 years later and we are still following it.

This is the behaviour we want to adhere to, and be known for
BDD with ZKD – what have we learnt?

BDD + ZKD:
This is the behaviour we want to adhere to and be known for