

Behaviour Driven Development

Outline

Beyond Test Driven Development

Single specification used by all disciplines = increased team effectiveness

Greater focus is placed on the specification of behaviour rather than "test"

Motivation: If I can express the requirement so that it can be understood by all, it can be delivered.

Articulates customer requirements

Behavioural Specifications

User Requirements authored to provide functional specification of "behaviour", in a semi-formal style (e.g. user story style).

Story:Wake me up in the morning

In order to get me to work on time
As a dedicated employee
I want to be alerted when it is time to get up

Scenario 1: Wake me up
Given the alarm is set
When the alarm time is reached
Then Sound audible alarm

Acceptance Criteria

Ubiquitous domain language

*Then sound audible alarm
When the alarm time is reached
Given the alarm is set*

Traceability

Implementation binds to Behavioural Specification

```
@Given("the alarm is set")
public bool theAlarmIsSet() {
    // Check if alarm time is now
}

@When("the alarm time is reached")
public void theAlarmTimeIsReached() {
    // Raise alarm
}

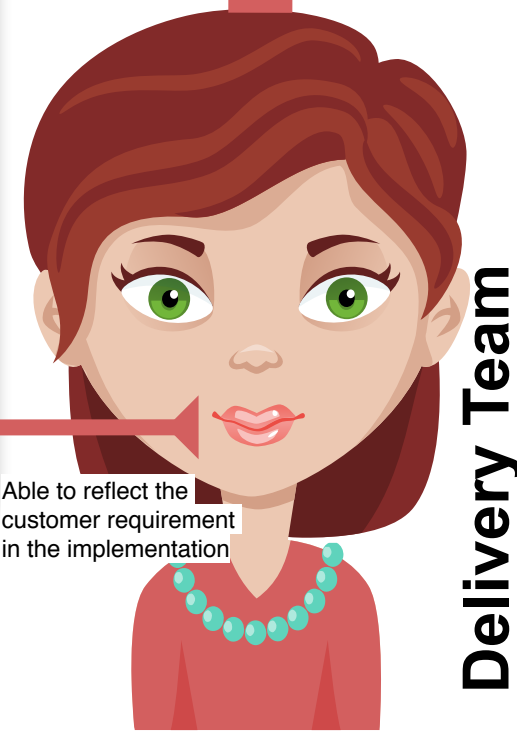
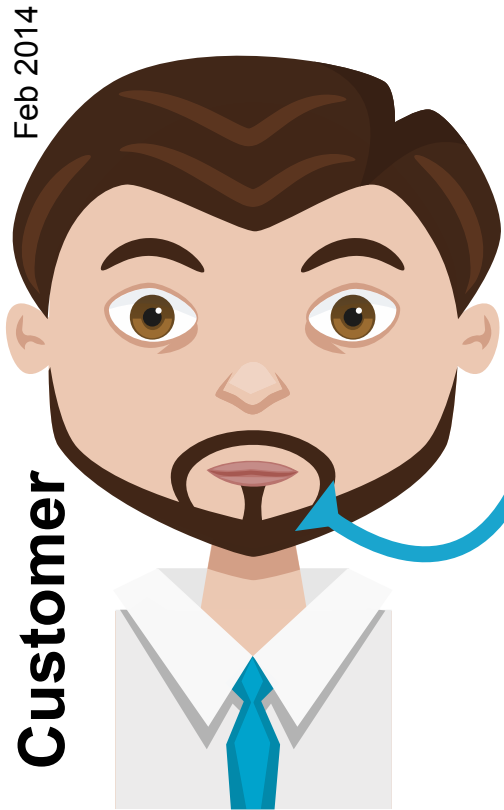
@Then("sound audible alarm")
public void soundAudibleAlarm() {
    // sound alarm
}

}
```

Motivation: If all the members in my team can truly understand what is required to be delivered, we can deliver it.

Able to understand and use the customer requirements and language

Able to reflect the customer requirement in the implementation



"About the author"

With over 20 years in the IT industry, **Stefan Brittain** is currently a Senior Manager and lean development proponent at a leading UK defence company. Original available at www.nonose.co.uk/whitepapers
Clipart courtesy of www.openclipart.org