Behavior Driven Development From the Trenches

Some anti-patterns

Public Domain, National Library of Scotland

DevNexus 2023 ❖ M. Jeff Wilson ❖ geek.macbeer@gmail.com

What is BDD?

"...Using examples at multiple levels to create a shared understanding and surface uncertainty to deliver software that matters."

—Dan North

What does it mean to do BDD?

BDD is a way of working that closes the gap between the technical team and the business by:

* Establishing collaboration across roles
* Building shared understanding on the desired behavior of the system
* Producing system documentation that guides development and is automatically checked against the system's behavior
* Working in rapid, small iterations to increase feedback and the flow of value.

Test Automation !== BDD

"Even on this list, the majority of people seem to think that Cucumber == Automated Tests == BDD, which is WRONG.

What people need to understand is:

* Cucumber is a tool for Specification By Example
* Specification By Example is just a better name for BDD
* Specification By Example means examples are written *before* implementation
* Specification By Example should happen iteratively, in collaboration with non-technical stakeholders
* Automated Tests are a by-product of Specification By Example
* Writing Automated Tests does *not* imply you're doing Specification By Example
* Using Cucumber for Automated Tests without doing Specification By Example is stupid
* Cucumber is not a tool for Automated Testing, it's a tool for Collaborative, Executable Specifications."

Aslak Hellesøy – December 2013, Cukes Google Group

The Anti-patterns

- Eyes on the Screen
- Gherkin Exhaustion
- Hammer, Meet Screw
- Chain Gang
- Important and Irrelevant
- Bad Company
- Know Your Audience
- What's My Motivation?

Eyes on the Screen

- **Description**: Business logic or rules are tested via the UI.
- **Symptoms**: Scenarios describe UI screens, URLs, buttons, forms, menus, etc.
- **Effects**: Slows down feedback, tests are fragile and slow.
- **Solution**: Refocus/rewrite scenarios to only cover the "what" and not the "how". Think in terms of user functionality, not clicks.

Self amusement, © Daniel Turner, CC-BY-SA-4.0 License

```
Feature: Transfer between accounts

 Scenario: Transfer from checking to savings
  Given I have $100 in my checking account
  * I have $72 in my savings account
  * I click on my checking account
  * I click “Transfer between accounts” button
  * I select my savings account as target
  * I enter $50.00 in the amount field
  When I click “Transfer Now”
  Then checking balance field should show $50
  * savings balance field should show $122
```

Ask yourself:

- What does the user intend to do?
- Are there ways to test that without (or before) the UI?

```
Feature: Transfer between accounts

 Scenario: Transfer from checking to savings
  Given I have $100 in my checking account
  * I have $72 in my savings account
  * I click on my checking account
  * I click “Transfer between accounts” button
  * I select my savings account as target
  * I enter $50.00 in the amount field
  When I click “Transfer Now”
  Then checking balance field should show $50
  * savings balance field should show $122
```

```
Feature: Transfer between accounts

  Scenario: Transfer from checking to savings
    Given I have $100 in my checking account
    * I have $72 in my savings account
    When I transfer $50 from checking to savings
    Then I should have $50 in my checking account
    * I should have $122 in my savings account
```

Gherkin Exhaustion

- Description: Trying to immediately capture scenarios in gherkin.
- Symptoms: Three Amigos collaborations are derailed by "trying to get the gherkin right".
- Effects: Initial enthusiasm followed by disillusionment when scenarios are missed or incomplete.
- Solution: Use the Three Amigos to the examples right. Convert them to gherkin later.

Step One

PO drafts the user story and provides some bullet point acceptance criteria.

User Story: ATM Withdrawals

As a bank customer, I need to obtain money from an ATM machine, so that I can deal with unexpected cash needs.

Acceptance Criteria:

- Withdrawal < balance = success
- Withdrawal > balance = failure
- Amount exceeds daily withdrawal limits

Step One

PO drafts the user story and provides some bullet point acceptance criteria.

Step Two

Three amigos identify examples needed to fully clarify the requirements.

Acceptance Criteria scenarios:

- Account balance is sufficient
- Insufficient funds
- Insufficient funds but account has overdraft protection
- Amount exceeds daily withdrawal limits
- ATM has insufficient cash
- ATM has incorrect denominations

Step One

PO drafts the user story and provides some bullet point acceptance criteria.

Step Two

Three amigos identify examples needed to fully clarify the requirements.

Step Three

Three amigos determine behavior of each scenario.

Acceptance Criteria scenarios:

- Account balance is sufficient
  - account debited, cash dispensed
- Insufficient funds
  - account not debited, no cash dispensed
- Insufficient funds but account has overdraft protection
  - difference transferred from line of credit to checking, account debited, cash dispensed
- Amount exceeds daily withdrawal limits
  - customer asked to try a lower amount or cancel transaction
- ATM has insufficient cash
  - customer asked to try a lower amount or cancel transaction

Step One

PO drafts the user story and provides some bullet point acceptance criteria.

Step Two

Three amigos identify examples needed to fully clarify the requirements.

Step Three

Three amigos determine behavior of each scenario.

Step Four

Developer and/or QA convert each example into gherkin as separate session.

```
Scenario: Insufficient funds but account has
overdraft protection
   Given my checking account has $57
   * I have overdraft protection
   * my available credit is $500
   When I withdraw $100 from the ATM
   Then my checking account should have $0
   * my available credit should be $457

Scenario: Amount exceeds daily limits
   Given the daily withdrawal limit is $350
   * I previously withdrew $320 today
   When I withdraw $40 from the ATM
   Then I should be prompted to withdraw an amount
   of $30 or less
```

Hammer, Meet Screw

- Description: The gherkin and the test automation are different representations of the requirements.
- Symptoms: To be automated, the gherkin must be changed or technical details about the test automation added.
- Effects: The source of truth about the requirements is lost.
- Solution: Adopt a tool such as Cucumber that can execute unaltered gherkin.

Revolver "casse-tete", ©Latente Flickr, CC-BY-SA-2.0 License

```
*** Settings ***
Resource          ../resources.robot
Resource          ../pages/login_pages.robot
Resource          ../pages/home_pages.robot
Resource          ../steps/login_steps.robot
Test Setup        Open Login Page Using Chrome Browser
Test Teardown     Close Browser

*** Variables ***
${valid_username}       UserID_Testing
${valid_password}       YOUR_VALID_PASSWORD
${invalid_data}         qwerty

*** Test Cases ***
Valid Login
  GIVEN Login Page Opened
  WHEN Input Username And Password    ${valid_username}    ${valid_password}
  THEN Home Page Opened

Invalid Login
  [Template]    Login with Invalid Credentials
  # username   password
  ${valid_username}   ${invalid_data}
  ${invalid_data}   ${valid_password}
  ${invalid_data}   ${invalid_data}

*** Keywords ***
```

```
Feature: Authenticate User
  As a bank user, I need the ability to
  prove who I am, so that no one else can
  access my accounts.

  Scenario: Valid Login
    Given Login Page Opened
    When Input Username 'Valid User' and
    Password 'Valid Password'
    Then Home Page Opened

  Scenario: Login with Invalid Credentials
    Given Login Page Opened
    When Input Username 'Valid User' and
    Password 'Invalid Data'
    Then Error Message Displayed
```

```
*** Settings ***
Resource          ../resources.robot
Resource          ../pages/login_pages.robot
Resource          ../pages/home_pages.robot
Resource          ../steps/login_steps.robot
Test Setup        Open Login Page Using Chrome Browser
Test Teardown     Close Browser

*** Variables ***
${valid_username}       UserID_Testing
${valid_password}       YOUR_VALID_PASSWORD
${invalid_data}         qwerty

*** Test Cases ***
Valid Login
  GIVEN Login Page Opened
  WHEN Input Username And Password    ${valid_username}    ${valid_password}
  THEN Home Page Opened

Invalid Login
  [Template]    Login with Invalid Credentials
  # username   password
  ${valid_username}   ${invalid_data}
  ${invalid_data}   ${valid_password}
  ${invalid_data}   ${invalid_data}

*** Keywords ***
```

```
Feature: Authenticate User
  As a bank user, I need the ability to prove
  who I am, so that no one else can access my
  accounts.

  Scenario: Valid Login
    When I login with valid credentials
    Then I should have access to my accounts

  Scenario: Login with Invalid Credentials
    When I login with invalid credentials
    Then I should have no access to any accounts
```

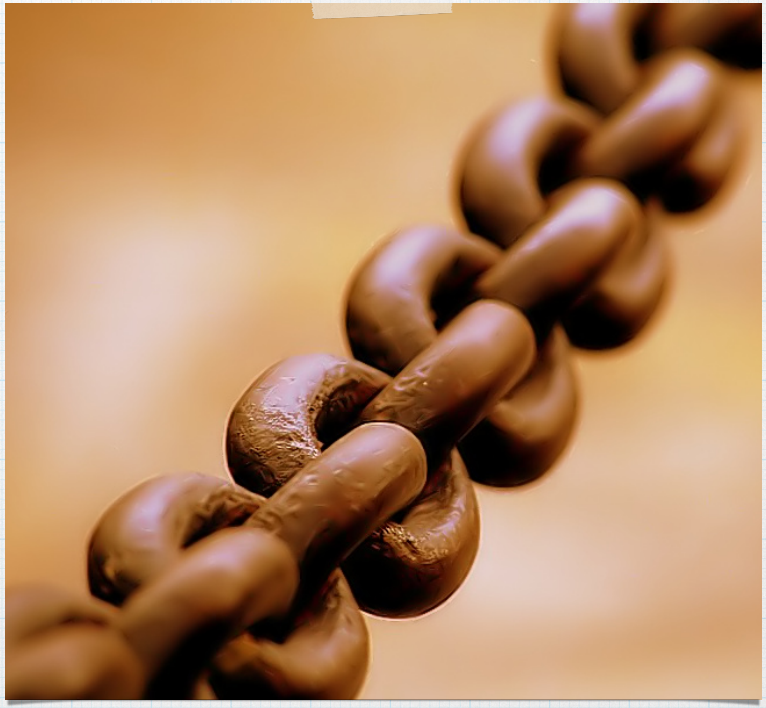
```
*** Settings ***
Resource          ../resources.robot
Resource          ../pages/login_pages.robot
Resource          ../pages/home_pages.robot
Resource          ../steps/login_steps.robot
Test Setup        Open Login Page Using Chrome Browser
Test Teardown     Close Browser

*** Variables ***
${valid_username}       UserID_Testing
${valid_password}       YOUR_VALID_PASSWORD
${invalid_data}         qwerty

*** Test Cases ***
Valid Login
  GIVEN Login Page Opened
  WHEN Input Username And Password    ${valid_username}    ${valid_password}
  THEN Home Page Opened

Invalid Login
  [Template]    Login with Invalid Credentials
  # username   password
  ${valid_username}   ${invalid_data}
  ${invalid_data}   ${valid_password}
  ${invalid_data}   ${invalid_data}

*** Keywords ***
```

```
Feature: Authenticate User
  As a bank user, I need the ability to prove
  who I am, so that no one else can access my
  accounts.

  Scenario: Valid Login
    When I login with valid credentials
    Then I should have access to my accounts

  Scenario: Login with Invalid Credentials
    When I login with invalid credentials
    Then I should have no access to any accounts
```

Which of these look like requirements?

Chain Gang

- Description: Workflow steps cannot be separated and tested in isolation.
- Symptoms: Multiple "When" and "Then" statements in a scenario.
- Effects: Slows down feedback as the entire workflow has to be complete before verification.
- Solution: Design or refactor workflows for testability and/or make the output of one scenario be the givens in the next.



Broad iron chain, ©Toni Lozano, CC-BY-2.0 License

"The Cardinal Rule of BDD: One Scenario, One Behavior!"

—Andrew Knight, Automation Panda

```
Scenario: Booking a flight
 When I searched for flights from LA to NYC
 Then only flights with available seats that match
 my fare should be displayed
 When I select a flight
 Then I should be prompted to select a seat
 And a seating map for the flight is displayed
 And only unassigned seats that match my fare will
 be selectable
 When I skip seat selection
 And the flight and fare selected indicates a meal
 Then I will be asked to select a meal option
 And only the meal options for that flight are
 displayed
 When I select a meal option
 Then I should be asked to confirm flight details
 And a warning message should appear “Seat will be
 assigned randomly, proceed?”.
```

Ask yourself:

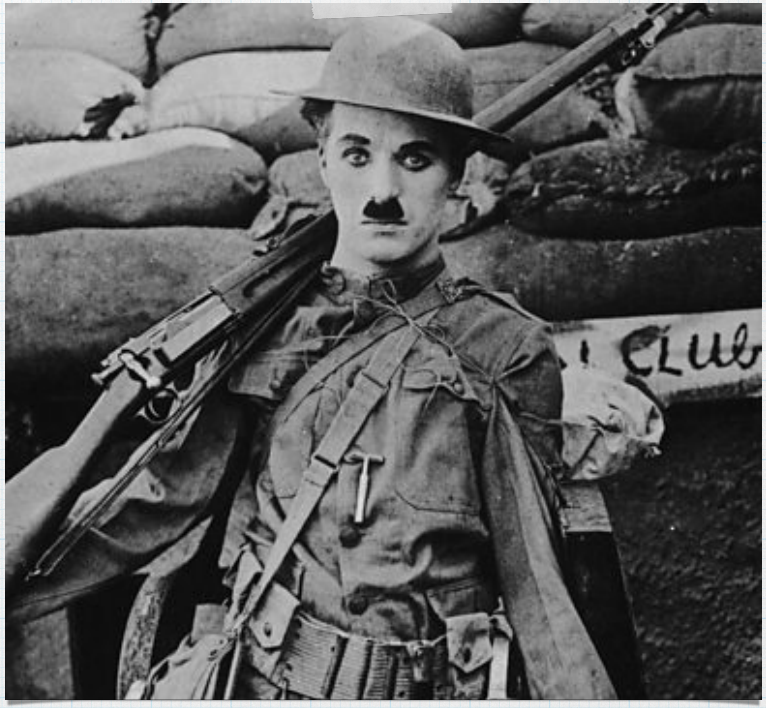
* How many business rules are being validated?
* Could I design the system to allow the steps to be broken up and individually tested?

```
Scenario: Booking a flight
 When I searched for flights from LA to NYC
 Then only flights with available seats that match
 my fare should be displayed
 When I select a flight
 Then I should be prompted to select a seat
 And a seating map for the flight is displayed
 And only unassigned seats that match my fare will
 be selectable
 When I skip seat selection
 And the flight and fare selected indicates a meal
 Then I will be asked to select a meal option
 And only the meal options for that flight are
 displayed
 When I select a meal option
 Then I should be asked to confirm flight details
 And a warning message should appear "Seat will be
 assigned randomly, proceed?".
```

```
Scenario: Flight availability
 When I searched for flights from LA to NYC
 Then only flights with available seats that
 match my fare should be displayed

Scenario: Seat Selection
 When I select a flight
 Then I should be prompted to select a seat
 And a seating map for the flight is displayed
 And only unassigned seats that match my fare
 will be selectable

Scenario: Skipping Seat Selection
 Given I have selected a flight
 And I skipped seat selection
 When I confirm the flight details
 Then a warning message should appear "Seat will
 be assigned randomly, proceed?".
```

Important And Irrelevant

* Description: Details that are important to the tests but not necessary to understanding the business rule are included in the gherkin.
* Effects: The goal of having the gherkin serve as specification is diminished because the extra detail confuses the reader.
* Solution: Move the detail from the gherkin to the glue code that implements the test.



```
Scenario: Update Password
  Whenever the password is updated, the user gets
  an email about the update as a security
  precaution

  Given I am editing my user profile
  And I enter a new password
  And I confirm the new password
  When I save the change
  Then I should get an email that tells me my
  password was updated
```

Ask yourself:

- Are the steps necessary to the business rule?
- Do they help clarify the requirement or do they tell the test what must be done?

```
Scenario: Update Password
Whenever the password is updated, the user gets an
email about the update as a security precaution

Given I am editing my user profile
And I enter a new password
And I confirm the new password
When I save the change
Then I should get an email that tells me my
password was updated
```

```
Scenario: Update Password
  Whenever the password is updated, the user gets
  an email about the update as a security
  precaution

  Given I am editing my user profile
  When I change my password
  Then I should get an email that tells me my
  password was updated
```

```
Scenario: Update Password
Whenever the password is updated, the user gets an
email about the update as a security precaution

Given I am editing my user profile
And I enter a new password
And I confirm the new password
When I save the change
Then I should get an email that tells me my
password was updated
```

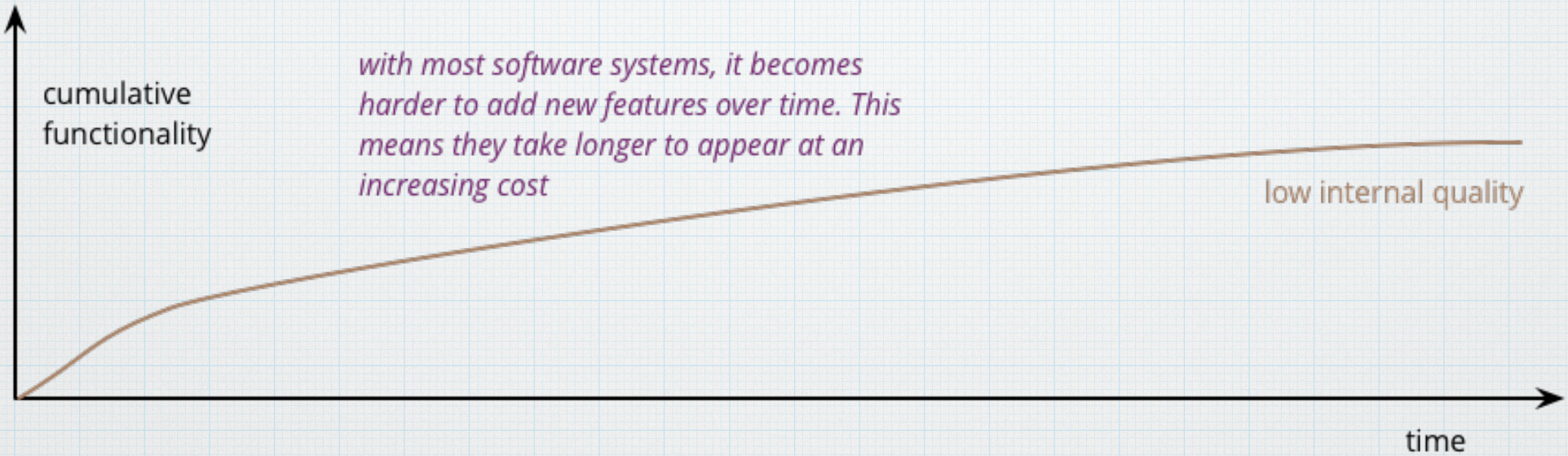
```
Scenario: Update Password
  Whenever the password is updated, the user gets
  an email about the update as a security
  precaution

  Given I am editing my user profile
  When I change my password
  Then I should get an email that tells me my
  password was updated

Scenario: Update Password
  Whenever entering a new password the user is
  required to enter it twice to confirm.

  Given I am editing my user profile
  When I change my password
  Then I should be prompted to re-enter the new
  password to confirm
```

Bad Company

- Description: A separate developer/team is brought in to write the test automation.
- Effects: Handoffs lead to delayed feedback and lower quality. The automation team may have different goals than the scrum team.
- Solution: Make test automation a development activity and part of your definition of done. Train the team, give them support and expect them to deliver less for a couple of sprints.

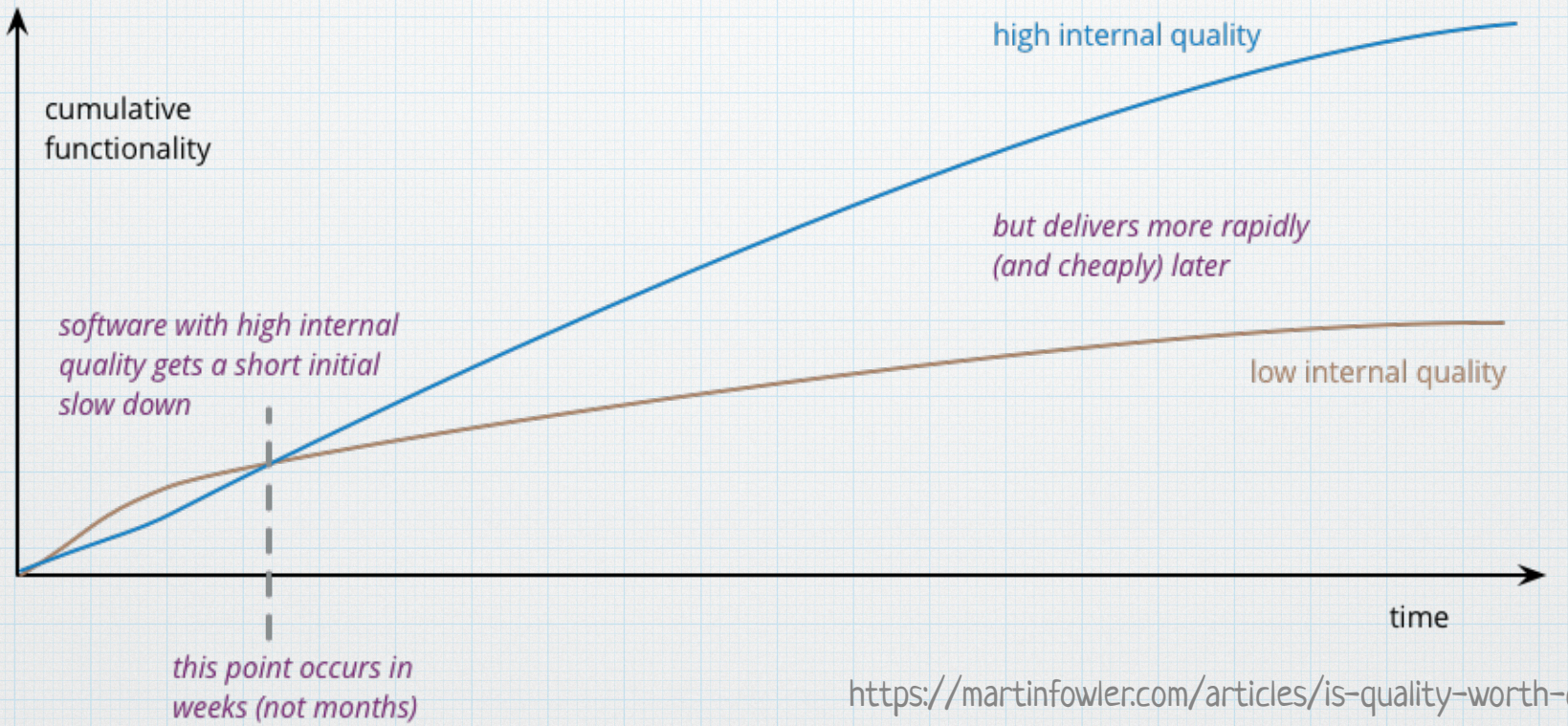
Dark Entity, ©George Fatin, CC-BY-SA-4.0

Visualizing Impact of Internal quality



https://martinfowler.com/articles/is-quality-worth-cost.html

Visualizing Impact of Internal quality



https://martinfowler.com/articles/is-quality-worth-cost.html

Know Your Audience

* Description: BDD specifications (business-facing) tests are mixed into test automation that checks internal quality (technology-facing).
* Effects: More difficult to create, find and maintain BDD specifications.
* Solution: Design separate repositories for businesss-facing versus technology-facing tests.

Captive Audience, Andrew Smith CC-BY-SA-2.0 License

Agile Testing Quadrants

Business-Facing (Building the Right Code)

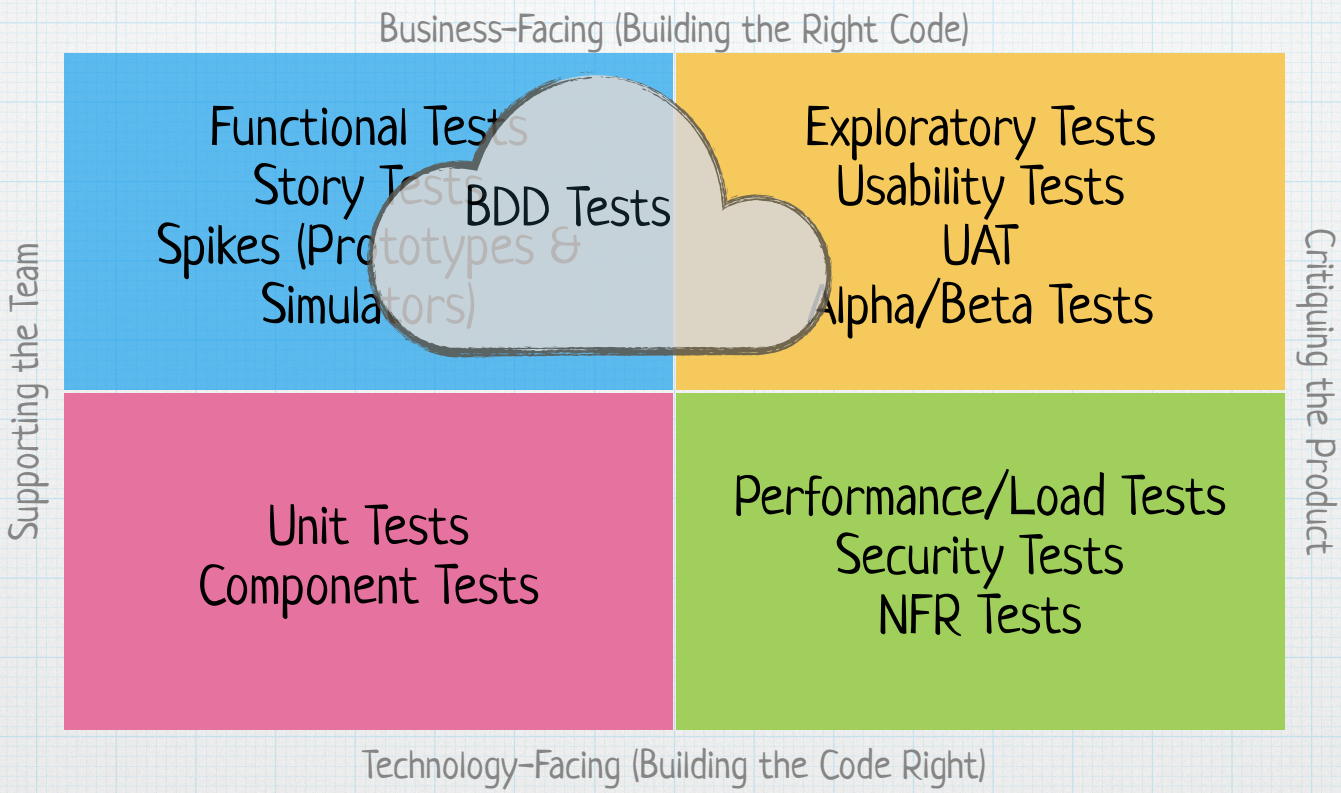
Supporting the Team

Functional Tests Story Tests Spikes (Prototypes & Simulators)	Exploratory Tests Usability Tests UAT Alpha/Beta Tests
Unit Tests Component Tests	Performance/Load Tests Security Tests NFR Tests

Critiquing the Product

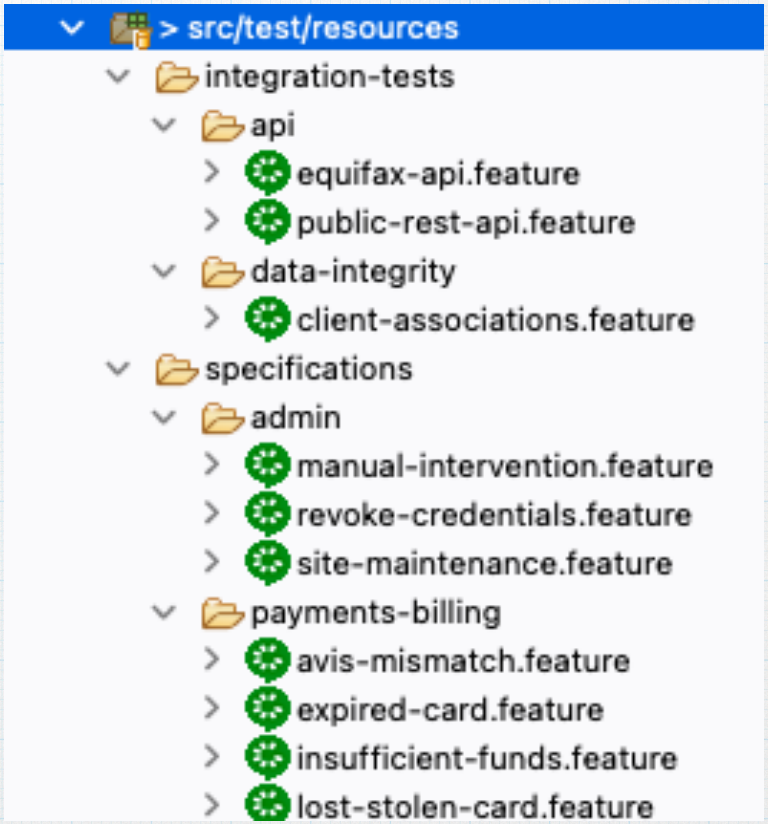
Technology-Facing (Building the Code Right)

Agile Testing Quadrants



- Cucumber will run all feature files found by recursing through a directory structure
- Organize BDD Specifications in its own directory to reflect logical groupings of features.
- Organize other tests in a different directory

https://cucumber.io/blog/bdd/solving-how-to-organise-feature-files/



What's My Motivation?

* Description: Scenarios are missing information that make them useful as documentation.
* Symptoms: Meaningless or missing names, no clarifying text.
* Effects: The value of using BDD as a collaboration tool is diminished.
* Solution: Treat every feature file as requirements documentation first.

```
# Comment any where in the file
@tag1
Feature: Title of your feature
  Free-form text description of your feature.
  Typically written in user story narrative
  format (As a ...  I need ... so that ...)

  @tag2
  Scenario: Title of your scenario
   Description of the scenario.

   Given I want to write a step with
   precondition
   And some other precondition
   When I complete action
   Then I validate the outcomes
   And check more outcomes
```

Treat the Feature file as if it were a page in your product's user manual.

Use the description fields to explain the feature and its scenarios.

Avoid use of "user" in both feature description and scenario steps — pick a user role or persona.

```
@slow
Feature: Avis mismatch
  As an app credit user, I need to update my
  address in the app when it doesn't match with
  the credit card record, so that my card is not
  declined.

  # We are using the standard persona for these
  scenarios: Amy, an 18-30 year old app user who
  wants to do everything through the app.

  Scenario: Update address when card is declined
    If the purchase is declined due to a mismatch
    in address, give the app user a chance to
    update the address and retry the transaction.

    Given Amy updated her credit card address
    But has not updated her address in the app
    When a purchase is declined because the
    addresses do not match
    Then Amy should be asked if she wants to update
    her address and resubmit the transaction

  Scenario: Save updated address
    If the app user updates the address, give the
    user a chance to update the address and retry
    the transaction.

    Given Amy has updated her address in response
    to a declined transaction
    And she has retried the transaction
    When the transaction is successful
    Then Amy should be asked if she would like to
    save the new address associated with the credit
    card
```

```
Scenario: Bonus saver account

 Given a bonus saver account with an end-of-month balance of <balance>
 And a previous end-of-month balance of <previous>
 When the interest is calculated
 Then the applicable interest rate should be <rate>
 And interest <earned> should be deposited into the account

 Examples:
  balance | previous | rate | earned
  10000   | 10000    | 4    | 40.00
  9000    | 9000     | 2    | 18.00
  10000   | 10300    | 4    | 40.00
  9000    | 9301     | 1    | 9.00
  10000   | 10301    | 2    | 20.00
```

How is interest calculated?
What is the business rule?

```
Scenario: Bonus saver account

 Given a bonus saver account with an end-of-month balance of <balance>
 And a previous end-of-month balance of <previous>
 When the interest is calculated
 Then the applicable interest rate should be <rate>
 And interest <earned> should be deposited into the account

 Examples:
  balance | previous | rate | earned | rule
  10000   | 10000    | 4    | 40.00  | Over 10000 earns 4%
  9000    | 9000     | 2    | 18.00  | Under 10000 earns 2%
  10000   | 10300    | 4    | 40.00  |
  9000    | 9301     | 1    | 9.00   | Under 10000 with over 300 withdrawn earns 1%
  10000   | 10301    | 2    | 20.00  | Over 10000 with over 300 withdrawn earns 2%
```

- There are more anti-patterns we didn't cover
  - Scenarios written by a single person
  - BDD as test scripts
  - Boring scenarios (No Sh!t, Sherlock)
  - Written too early
  - Etc.

Questions?

- Agile Coach
- Certified SAFe® Practice Consultant (SPC6)
- Senior/Lead Developer/Architect
- DevOps + Test Automation SME

geek.macbeer@gmail.com
@mjeffw
https://linkedin.com/in/mjeffwilson

Coaching at the intersection of code and agile.