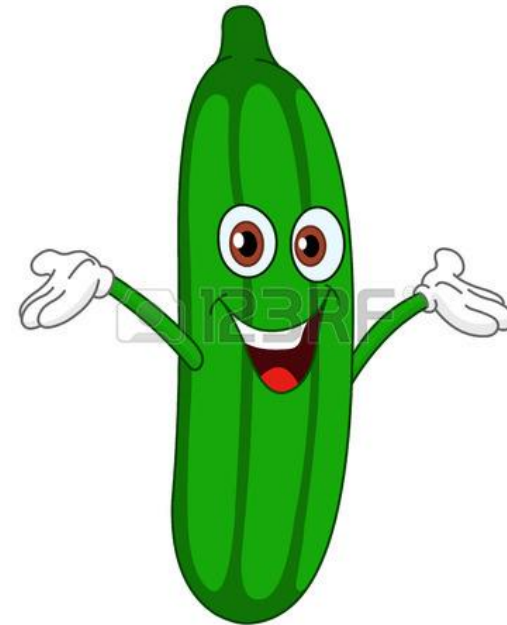
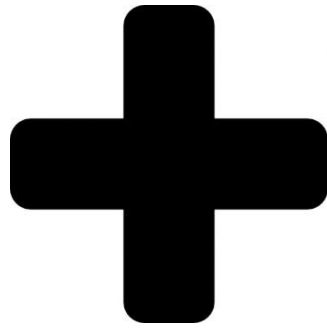




Improve Quality Outcome by Adopting BDD with Cucumber Automation

Barbecued Cucumbers



Writing Gherkin – Practice on Non-IT Things



Scenario: Preparing a talk for SIGIST

Given I am in the kitchen

And I am writing a talk on 'BDD' with 'Cucumber' -

And the love of my life is looking over my shoulder

When she reads what I have written

And makes an amusing comment about 'Cucumber BBQ'

Then I will include that story in my talk as an ice breaker

And my audience will be relaxed and amused

Good Gherkin Vs Bad Gherkin



Bad Gherkin – Everything Done by the Developer!



```
@test_text
Scenario: Test1
  Given I browse to 'https://pypi.python.org/pypi'
  #Given I browse to [google_home.url]
  When I enter 'selenium' to [pypi.home.txt_search]
  And I click ['#submit']
  Then I see browser url contains 'action=search&term=selenium'
  Then I see ['#content > div.section > h1'] text contains 'Index of Packages Matching'
  #Then I see [pypi.home.txt_search] @value contains 'sele'
  Then I see [pypi.home.txt_search] @value not contains 'sele'
  And I see ['#term'] @value is empty
  And I see ['//*[@id="term"]'] @value is empty
```

Bad Gherkin – Everything Done by a Non-Technical Tester



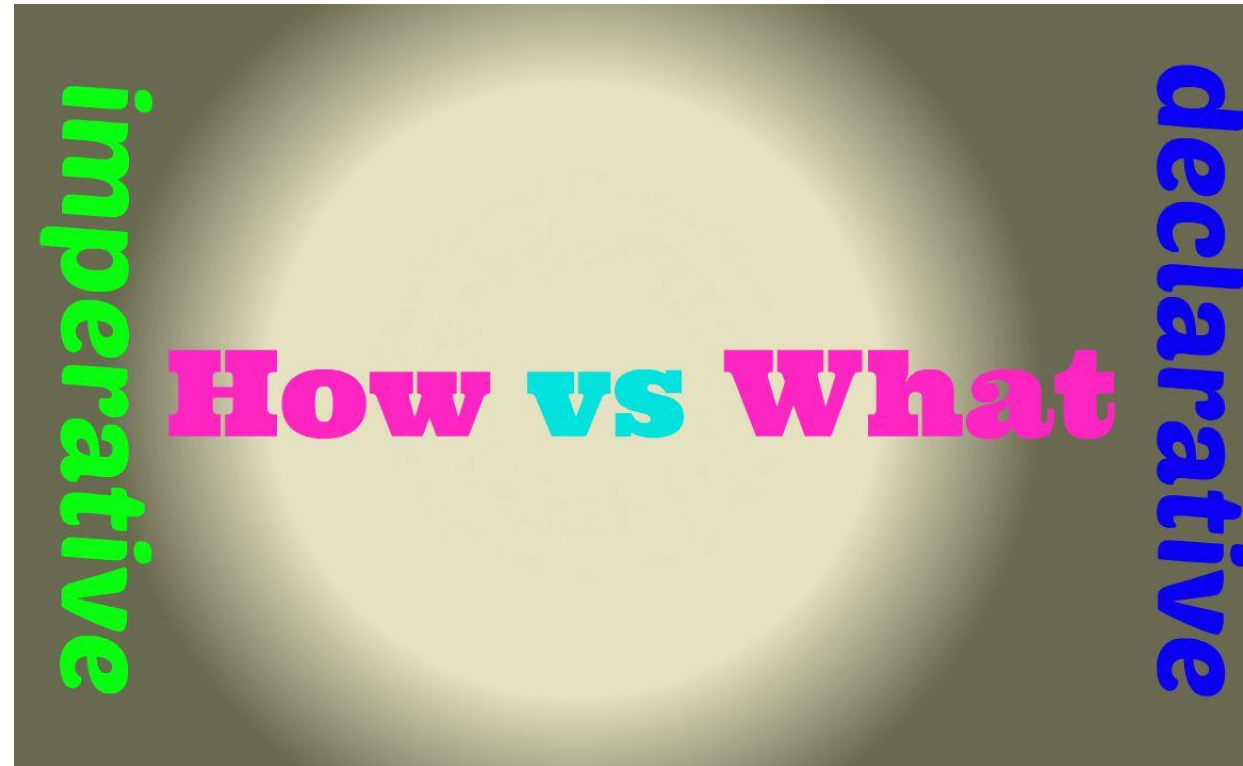
```
Scenario: As a user I want to make sure that ABOUT is available as an option
  Given I am on the "login_layout" screen
  Then I wait for 2 seconds
  When I tap on the "membership_no" field
  Then I wait for 2 seconds
  Then I enter my "test_android12" details into "membership_no"
  Then I wait for 2 seconds
  Then I enter my "test1234" details into "password"
  When I tap on done from the keyboard
  Then I tap on the "btn_login" button
  Then I wait for 5 seconds
  Then I touch the settings button
  Then I wait for 2 seconds
  When I press the menu item for "About"
  Then I wait for 3 seconds
```

Still Bad!



```
Scenario: As a user I want to make sure that ABOUT is available as an option
  Given I am on the "login_layout" screen
  When I tap on the "membership_no" field
  Then I enter my "test_android12" details into "membership_no"
  Then I enter my "test1234" details into "password"
  When I tap on done from the keyboard
  Then I tap on the "btn_login" button
  Then I touch the settings button
  When I press the menu item for "About"
```

Good Gherkin - Imperative vs Declarative



Imperative - How

Given I am on the 'Login' page

When I enter 'Tim' in the 'Username' field

And I enter 'Tim123' in the 'Password' field

And I click the 'Login' button

Then I see the 'Home' page

And I see that the 'Username' label contains 'Tim'

Imperative – Pros and Cons

Pros

- Easy to build a re-usable testing framework
 - Easy to Maintain
 - Easy to add tests without touching the technical stuff
- Step definitions are more resistant to becoming anti patterns
- Can still be Understood by non-technical users

Cons

- Can make Scenarios verrrrrrryyyyyy looooooonnnnnng
- Can make the Scenarios dry and boring to read
- Is probably not how a business user would choose to write the Scenarios
- The two examples of Bad Gherkin I gave previously are when Imperative takes complete precedence over Declarative.

Declarative - What

Given I am on the Login page

When I enter my username and password

Then I am logged into the system

Declarative – Pros and Cons

Pros

- Business Facing Stakeholders and easily read, write and understand the narrative – Narrative is better
- Leaves more room for creativity with the development team

Cons

- Will create more step definitions to maintain
- Will create Anti-Patterns in the Step Definitions

Conjunction Steps

When I enter my username and password

Should Be:

When I enter my username

And I enter my password

Data Tables

Given I am on the 'Login' page

When I log in as a user:

UserName	Password
Tim	Tim123

Then I am logged into the system

Examples

Scenario Outline: A Template

Given I am on the 'Login' page

When I enter <username> in the username field

And I enter <password> in the password field

And I click on 'Login'

Then I am logged into the system

Examples:

username	password
Tim	Tim123
Jim	Jim345

The Step Definition – A Parameter

```
[Given(@"I am on the (.*) page")]
public void GivenIamOnaPage(String page)
{
    //some test automation that takes you to any page based upon the
    //parameter
}
```


The Step Definition – A Data Table

```
[When(@"I Log in:")]  
public void GivenIamOnaPage(Table table)  
{  
    //some test automation that reads the table collection  
    //some test automation that logs you in  
}
```

Cucumber is About Communication

Cucumber is a communication tool and not a programming language:

- Learn how to write really good feature files – practice with non-IT related Gherkin first
- Badly written Gherkin will cost your organisation money – as the BDD champion of your organisation, make your Gherkin awesome
- Identify a style of narrative that will best suit your organisation
- Use Regex in the step definitions effectively

Getting Buy In



Organisational Readiness



- Do other people on my team understand BDD and do they want to do it?
- What is my role on my team and who do I need help from to do this properly?
- Will there be resistance to what I would like to do and how do I tackle this?
- Do we have supportive infrastructure?

Selling Points



Improved Accountability

- Which involves the whole team and not just the testers in the quality of the **PRODUCT!**
- Improved Teamwork

Improved Communication

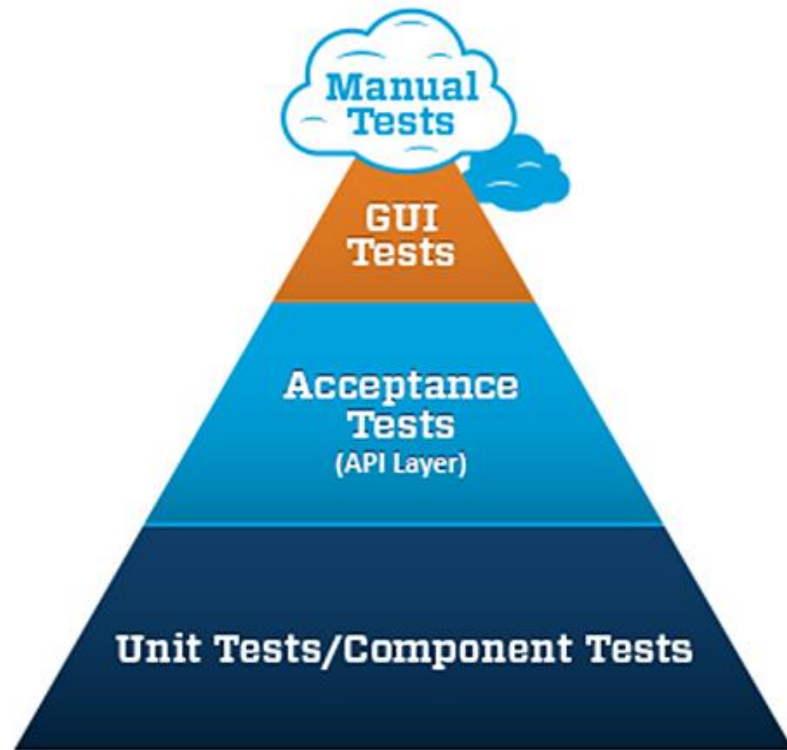
- Testable requirements
- All stakeholders engaged early in the process

The Right People



- Test Automation Specialists – DevOps Testers
- Developers who are keen to work in this way

Testing Frameworks



- GUI Tests – Have we built the right thing
- Acceptance Tests – Have we built the right thing in the right way
- Unit Tests – Have we built the thing right

Summary – Get the Most out of BDD

- Get Buy-in from all levels in your organisation
- Ask yourself questions about what you need in place to get your BDD right
- Identify selling points in the process first – get people interested in the process behind BDD before Cucumber
- If you identify that you have a shortfall in skills, don't just battle on, see what you can do to get the right skills
- Use Automation tools effectively, in the right way to maximise stability and reduce brittleness

Roles

Software Developers
Business Analysts
Software Testers

¡THREE AMIGOS!



Software Developers

- Developers have to get things right first time
- Added Visibility and Accountability can increase pressure
- Developer time can become more compressed



Mitigating Actions

- Manage expectations from the start
- Have good managerial support
- Try to find the root cause of the problems
 - Time management - make sure that the Developers can manage their time effectively; that they are not taking on too much
 - Are your iterations too short
 - Are your stories too big
- Keep your BDD tests close to the Source Code and write the steps in the same programming language so they cannot be forgotten
- Don't put added pressure on the developers by forcing them to write the tests themselves

Business Analysts

- Lean on the regex to allow slight differences in sentence structure and grammar
- Create a dictionary of Gherkin Steps that they can refer to or document sets of rules around them
- In the setup phase get the BA's to learn the style of Gherkin you will use by practicing with day-to-day events – Get them to BBQ some Cucumbers
- Get the Manual Testers to tidy up any Gherkin the BA's write

Software Testers



Where Testers May Help the Team

- Testers can get involved in requirements planning with the BA's to help in writing good, accurate Gherkin
 - Testers will be freed up from doing the majority of Acceptance Testing and testing that we have “built the right thing”
- Testers may become more technical
- Testers are used to writing test cases so should be able to write really good Gherkin
- A Test Automation Engineer with Dev Ops experience is required

Food for Thought - Takeaway

- Learn how to use your tools effectively
- Plan your setup well and implement supportive processes
- Have good Buy In
- Be well prepared to tackle any issues effectively
- Fix issues early and keep going
- If the process works well, you will see an improvement in quality and ROI
 - Because you will have fewer bugs in your backlog
 - You will have more time to tackle non-functional testing earlier in the SDLC e.g. Performance and Security