Contrasting test automation and BDD
- an “interactions over tools” perspective

Seb Rose
Twitter:  @sebrose
Blog:   https://cucumber.io/blog/
E-mail:   seb.rose@smartbear.com
WARNING

This presentation includes interactive segments.

There will be polls, breakout rooms, and online whiteboards.
Agenda

Interactions over tools
Test automation
Behaviour-driven development
Compare and contrast
Shift left
Confusion
Wrap-up
How would your team respond, if I asked them: “Do you use BDD?”
Agenda

Interactions over tools
Test automation
Behaviour-driven development
Compare and contrast
Shift left
Confusion
Wrap-up
Names have power
Naming is a process

“Names are the place we communicate our insights and intentions to other humans.”

Arlo Belslee

https://www.digdeeproots.com/articles/naming-as-a-process/
What does “agile” mean?

https://newmatilda.com/2016/10/10/jobseekers-for-every-job-the-unemployed-elephant-in-the-room/
Perspective is important

“... we have come to value individuals and **interactions** over processes and **tools**...”

http://agilemanifesto.org/
Agenda

Interactions over tools

Test automation

Behaviour-driven development

Compare and contrast

Shift left

Confusion

Wrap-up
Define test automation

“... test automation is the use of software separate from the software being tested to control the execution of tests and the comparison of actual outcomes with predicted outcomes.”

Wikipedia
What are the benefits?
What are the benefits?

Faster feedback
What are the benefits?

- Faster feedback
- More frequent feedback
What are the benefits?

- Faster feedback
- Higher quality
- More frequent feedback
What are the benefits?

- Faster feedback
- Higher quality
- More frequent feedback

@sebrose seb.rose@smartbear.com
What are the benefits?

- Faster feedback
- Higher quality
- More frequent feedback
- Repeatability
What are the benefits?

- Faster feedback
- Higher quality
- Repeatability
- More frequent feedback
What are the benefits?

- Higher quality
- Faster feedback
- More frequent feedback
- Repeatability
- Lower costs
What are the benefits?

- Faster feedback
- Higher quality
- More frequent feedback
- Repeatability
- Lower costs
Discuss benefits and risks

Higher quality
Repeatability
Lower costs

Once you arrive in your breakout room, note the room number (which is displayed in the title bar of the Zoom window).

What are the challenges?
What are the challenges?

Skills
What are the challenges?

Skills

Testability
What are the challenges?

Skills

Testability

Late feedback
What are the challenges?

- Skills
- Testability
- Late feedback
- Overconfidence
Interactions
Interactions

“The business”
Interactions

“The business”

Here are our requirements

Architecture, Design
Interactions

“The business”

Here are our requirements

Architecture, Design

Here’s the specification

Test Automation

Development & Testing
Here are our requirements

Here's the specification

It works on my machine
Interactions

“The business”

Here are our requirements

Architecture, Design

Here’s the specification

Test Automation

It works on my machine

Development & Testing

Something’s not right!

Here’s the specification

It works on my machine
Interactions

"The business"

Here are our requirements

Architecture, Design

There are some outstanding issues

Test Automation

Here's the specification

Development & Testing

It works on my machine

Something's not right!
Agenda

Interactions over tools
Test automation
**Behaviour-driven development**
Compare and contrast
Shift left
Confusion
Wrap-up
Define BDD

Create a shared understanding of the requirements through collaboration, typically achieved through a structured conversation centered on rules and examples.

Examples of system behaviour are documented using business terminology.

The documentation is automated, creating living documentation that verifies the system’s behaviour.

https://cucumber.io/docs/bdd/
Define BDD

Create a shared understanding of the requirements through collaboration, typically achieved through a structured conversation centered on rules and examples.

Examples of system behaviour are documented using business terminology.

The documentation is automated, creating living documentation that verifies the system's behaviour.

https://cucumber.io/docs/bdd/
Define BDD

Create a shared understanding of the requirements through collaboration, typically achieved through a structured conversation centered on rules and examples.

Examples of system behaviour are documented using business terminology.

The documentation is automated, creating living documentation that verifies the system's behaviour.

https://cucumber.io/docs/bdd/
Define BDD

Create a shared understanding of the requirements through collaboration, typically achieved through a structured conversation centered on rules and examples.

Examples of system behaviour are documented using business terminology.

The documentation is automated, creating living documentation that verifies the system's behaviour.

https://cucumber.io/docs/bdd/
What are the benefits?
What are the benefits?

Early learning
What are the benefits?

- Early learning
- Shared understanding

Diagram:
- User Story
- Discovery
- Formulation
- Automation
- Working Software
What are the benefits?

- Early learning
- Shared understanding
- Faster feedback
What are the benefits?

Early learning

Shared understanding

Faster feedback

Living documentation
What are the benefits?

- Early learning
- Shared understanding
- Faster feedback
- Living documentation
- Higher quality
What are the benefits?

- Early learning
- Shared understanding
- Faster feedback
- Living documentation
- Higher quality
- Lower costs
What are the challenges?


By Gary Todd - https://www.flickr.com/photos/101561334@N08/28169055190/

https://www.eso.org/public/images/potw1938a/
What are the challenges?

Change in working practices


By Gary Todd - https://www.flickr.com/photos/101561334@N08/28169055190/

@sebrose

seb.rose@smartbear.com
What are the challenges?

- Change in working practices
- Immature tooling

By Gary Todd - https://www.flickr.com/photos/101561334@N08/28169055190/

@sebrose

seb.rose@smartbear.com
What are the challenges?

- Change in working practices
- Immature tooling
- Other automated testing needed
Interactions

#1 Pick a user story
Elicit and prioritize functionality

#2 Requirement workshop
Explore and discover details through examples

#3 Formulate
Convert example to scenario

#4 Review
Get feedback on scenarios from business

#5 Automate
Write test automation code

#6 Red
Write a programmer test

#6 Green
Write enough code to make the test pass

#6 Red
Tidy up your code

#8 Release
Produce a potentially shippable increment

Development & Testing:
- Red
- Green
- Refactor
- Implement
- Develop

Supplementary tests
e.g. exploratory, penetration, load

Business:
- Architecture, Design
- Test Automation

http://bddbooks.com

@sebrose
seb.rose@smartbear.com

https://www.linkedin.com/pulse/keep-your-scenarios-brief-seb-rose/
Interactions

Business

Architecture, Design

Development & Testing

Test Automation

@sebrose

seb.rose@smartbear.com

Interactions

#1 Pick a user story
Elicit and prioritize functionality

#2 Requirement workshop
Explore and discover details through examples

#3 Formulate
Convert example to scenario

#4 Review
Get feedback on scenarios from business

#5 Automate
Write test automation code

#6 Red
Write a programmer test

Green
Write enough code to make the test pass

#7 Supplementary tests
E.g. exploratory, penetration, load

#8 Release
Produce a potentially shippable increment

https://www.linkedin.com/pulse/keep-your-scenarios-brief-seb-rose/

http://bddbooks.com

@sebrose

seb.rose@smartbear.com
Interactions

1. Pick a user story
   * Elicit and prioritize functionality

2. Requirement workshop
   * Explore and discover details through examples

3. Formulate
   * Convert example to scenario

4. Review
   * Get feedback on scenarios from business

5. Automate
   * Write test automation code

6. Red
   * Write a programmer test

7. Supplementary tests
   * e.g. exploratory, penetration, load

8. Release
   * Produce a potentially shippable increment

http://bddbooks.com

@sebrose
seb.rose@smartbear.com
Agenda

Interactions over tools
Test automation
Behaviour-driven development
**Compare and contrast**
Shift left
Confusion
Wrap-up
Comparison

BDD

Test-automation
Comparison

BDD

Discovery
- collaborative

Test-automation
Comparison

BDD

Discovery
- collaborative

Formulation
- collaborative
- shared understanding

Test-automation
Comparison

**BDD**
- Discovery
  - collaborative
- Formulation
  - collaborative
  - shared understanding
- Automation
  - by/with developers
  - guides coding
  - living documentation
  - other testing needed

**Test-automation**
Comparison

BDD
- Discovery: collaborative
- Formulation: collaborative, shared understanding
- Automation: by/with developers, guides coding, living documentation, other testing needed

Test-automation
- Automation: after/in parallel to coding, often siloed, performance/penetration etc., not necessarily business-readable

@sebrose
seb.rose@smartbear.com
**Comparison**

**BDD**
- Discovery
  - collaborative
- Formulation
  - collaborative
  - shared understanding
- Automation
  - by/with developers
  - guides coding
  - living documentation
  - other testing needed

**Test-automation**
- Automation
  - after/in parallel to coding
  - often siloed
  - performance/penetration etc.
  - not necessarily business-readable

Automated, business-readable acceptance tests
Prevalence

BDD

Test-automation
Executive summary

Breadth of impact

Applicable test domains

BDD

TA
Executive summary

Breadth of impact

Applicable test domains

BDD

TA

Size of circle has no meaning

@sebrose

seb.rose@smartbear.com
Agenda

Interactions over tools
Test automation
Behaviour-driven development
Compare and contrast
**Shift left**
Confusion
Wrap-up
Cost of change curve

https://www.amazon.co.uk/Engineering-Economics-Prentice-Hall-Computing-Technology/dp/0138221227
Cost of change curve

https://www.amazon.co.uk/Engineering-Economics-Prentice-Hall-Computing-Technology/dp/0138221227
Cost of change curve

Time

Cost of change

Requirements | Design | Code | Test | Acceptance | Operations

https://www.amazon.co.uk/Engineering-Economics-Prentice-Hall-Computing-Technology/dp/0138221227
The principle of Shift Left is to take a task that's traditionally done at a later stage of the process and perform that task at earlier stages.
Shift Left Testing?

Can we test ...
Shift Left Testing?

Can we test ...

Requirements  Performance  Contracts
Design          Code          Tests
Shift Left Testing?

Can we test ...

- Requirements
- Performance
- Contracts
- Design
- Code
- Tests

?
Shift Left Testing?

Can we test ...

- Requirements
- Performance
- Contracts
- Design
- Code
- Tests

We can, if we define testing as:

“a process of gathering information ... with the intent that the information could be used for some purpose”

Gerry Weinberg, Perfect Software and Other Illusions About Testing
What are the benefits?
What are the benefits?

Early learning

Shift Left

https://devopedia.org/shift-left
What are the benefits?

- Early learning
- Faster feedback

https://devopedia.org/shift-left
What are the benefits?

- Early learning
- Faster feedback
- Lower costs

Shift Left

https://devopedia.org/shift-left

@sebrose
seb.rose@smartbear.com
What are the benefits?

- Early learning
- Faster feedback
- Lower costs
- Higher quality

Shift Left

https://devopedia.org/shift-left
What are the benefits?

- Early learning
- Faster feedback
- Lower costs
- Higher quality
- Reduced risk

Shift Left

https://devopedia.org/shift-left
In breakout rooms, compare these benefits to those promised by BDD and Test Automation, with a short exercise in Miro:

https://miro.com/app/board/o9J_IR3qH40=/?moveToWidget=3074457355254936142&cot=10
Shift Left compared

Delivered by Test Automation

Delivered by BDD

Reduced risks
Lower costs
Higher quality
Faster feedback
Early learning
Shift Left compared

Delivered by Test Automation

Delivered by BDD

- Faster feedback
- Reduced risks
- Lower costs
- Higher quality
- Early learning
Shift Left compared

Delivered by Test Automation

Delivered by BDD

Faster feedback

Lower costs

Reduced risks

Early learning

Higher quality
Shift Left compared

Delivered by Test Automation

Delivered by BDD

Faster feedback

Higher quality

Lower costs

Reduced risks

Early learning
Shift Left compared

Delivered by Test Automation

Faster feedback

Higher quality
Lower costs
Reduced risks

Early learning

Delivered by BDD

@sebrose
seb.rose@smartbear.com
Shift Left compared

Delivered by
Test
Automation

- Faster feedback
- Higher quality
- Lower costs
- Reduced risks
- Early learning

Delivered by BDD

@sebrose  seb.rose@smartbear.com
Agenda

Interactions over tools
Test automation
Behaviour-driven development
Compare and contrast
Shift left

Confusion

Wrap-up
Confusion

https://cucumber.io/blog/bdd/bdd-is-not-test-automation/
Given/When/Then are just words

Following a BDD approach

Using Given/When/Then for test automation
Working software is the goal

“... we have come to value working software over comprehensive documentation...”
Tester skills still valuable

1. Pick a user story
   - Elicit and prioritize functionality

2. Requirement workshop
   - Explore and discover details through examples

3. Formulate
   - Convert example to scenario

4. Review
   - Get feedback on scenarios from business

5. Automate
   - Write test automation code

6. Red
   - Write a programmer test
   - Write enough code to make the test pass

7. Supplementary tests
   - e.g. exploratory, penetration, load

8. Release
   - Produce a potentially shippable increment
Tester skills still valuable

1. Pick a user story
   - Elicit and prioritize functionality

2. Requirement workshop
   - Explore and discover details through examples

3. Formulate
   - Convert example to scenario

4. Review
   - Get feedback on scenarios from business

5. Automate
   - Write test automation code

6. Red
   - Write a programmer test
   - Write enough code to make the test pass

7. Supplementary tests
   - e.g. exploratory, penetration, load

8. Release
   - Produce a potentially shippable increment

@sebrose
seb.rose@smartbear.com
Tester skills still valuable

#1 Pick a user story
Elicit and prioritize functionality

#2 Requirement workshop
Explore and discover details through examples

#3 Formulate
Convert example to scenario

#4 Review
Get feedback on scenarios from business

#5 Automate
Write test automation code

#6 Red
Write a programmer test

#7 Supplementary tests
e.g. exploratory, penetration, load

#8 Release
Produce a potentially shippable increment

Develop

Implement

Refactor
Tidy up your code

Green
Write enough code to make the test pass
Agenda

Interactions over tools
Test automation
Behaviour-driven development
Compare and contrast
Shift left
Confusion
Wrap-up
POLL

Does your team use BDD?
Takeaways
Takeaways

Test automation has more applicable domains
Takeaways

- BDD has a broader impact
- Test automation has more applicable domains
Takeaways

- Given/When/Then is not a discriminator
- BDD has a broader impact
- Test automation has more applicable domains
Takeaways

Given/When/Then is not a discriminator

BDD has a broader impact

Test automation has more applicable domains

Test automation on its own does not help "shift left"
Takeaways

- Given/When/Then is not a discriminator
- Test automation has more applicable domains
- BDD has a broader impact
- It’s not either/or. The best teams practice BDD and test automation
- Test automation on its own, does not help “shift left”
Seb Rose

Twitter: @sebrose
Blog: https://cucumber.io/blog/
E-mail: seb.rose@smartbear.com