Shift Testing Left to Build In Quality

Ken Pugh
ken@kenpugh.com
What Are the Tests for This Session?

- Also known as objectives
- Describe how Shift Left Testing helps build in quality to foster continuous integration and continuous deployment
- Identify how loopbacks and delays are reduced by Shift Left Testing
- Identify the different aspects of Shift Left Testing
- Describe the collaboration in Shift Left Testing
Ken Pugh
Co-creator of SAFe Agile Software Engineering
Ken Pugh, Inc.
Ken Pugh

- ATDD/BDD, TDD, BVDD, Lean, Scrum, SAFe, Design Patterns
- Over 2/5 century of software development experience
- Author of seven books including:

  Lean Agile Acceptance Test-Driven Development: Better Software Through Collaboration
  Prefactoring: Extreme Abstraction, Extreme Separation, Extreme Readability
  Interface Oriented Design

atdd@kenpugh.com
https://www.linkedin.com/in/kenpugh/
http://acceptancetestdrivendevelopment.com
Overall Rule

There are exceptions to every statement, except this one
Context is everything

Everything exists in a context

Everything is always true in some context
Perspective
Introduction
CALMR

- Culture of shared responsibility
- Automation of continuous delivery pipeline
- Lean flow accelerates delivery
- Measurement of everything
- Recovery enables low risk releases
Built In Quality

- Key Value in SAFe ®
- Building the right thing
- Building that thing right
Why invest in shifting left?

Relative cost of a defect based on when it was discovered

This is relative cost. Do you know what your relative costs are?
Traditional testing (V-Model)

- Write Feature
- Write Story
- Write Code
- Test Feature
- Test Story
- Test Code
Delays and loopbacks

1. Decide
2. Analyze
3. Design
4. Implement
5. Test
6. Deploy

Delay
Delay
Delay
DELAY
Delay
Which is it?

- Shift Testing Left
- Shift Left Testing

- You can’t test something until you have that something
- So it’s really about creating the test on the left, not executing it
Shift Left Testing

Think testing early

Decide → Analyze → Design → Implement → Test → Deploy

Feature

Customer
Shift testing left

Write Feature → Feature Test
Write Story → Story Test
Write Code → Code Test

... always thinking testing...
... always thinking testing...
... always thinking testing...

Shift Testing Left
Write testable features, stories, and code

- Define Hypothesis
- Validate Hypothesis
- Write Feature
- Test Feature
- Write Story
- Test Story
- Write Code
- Test Code

- CODE TESTS (TDD)
- STORY TESTS (BDD)
- FEATURE TESTS (BDD)
- BENEFIT TEST (HDD)
Test and requirements are related

Every requirement should have a test
Every test is a requirement
   If no deploy with failing test
Need requirements before implementation
Flow

- HDD – Hypothesis Driven Development – Product Management / Owner
- BDD – Behavior Driven Development -Triad
- TDD – Test Driven Development – Developer (or Pair)
Hypothesis Driven Development (HDD)
Hypothesis Driven Development (HDD)

- Idea for a functional change
- Hypothesis → customer’s use of that functionality
- Hypothesis → test as to whether you are building the right thing

- From Lean Startup / Lean UX / others
Minimum Marketable Feature (MMF):

**DEFINITION**
Smallest piece of functionality with intrinsic business/market value

**PURPOSE**
Release functionality sooner, learn from users faster
Use benefit hypothesis for MMF to validate business value

- Learn what users truly desire based on their behavior
Cruise control with speed automatically by speed limit
- Speed limit determined by
  - GPS with Map
  - or
    - Sign reader
    - or both

Autonomous vehicles will require this
Minimum Marketable Feature – More Detail

Given speed limit sign

When car views it

Then speed is set to 35 mph
MMFs need a benefit hypotheses with metrics

Template:

- We think <capability> will produce <outcome> as measured by <metric>

Example:

- We think that Cruise Control with Speed Limit Finder will be used by drivers 90% of the time as measured by telemetry
So what do you think will be the results?

Add ability to set speed X amount over speed limit

- 5
- 10
- 15
- 20

Just display the speed limit

Display a warning if over speed limit
Behavior Driven Development (BDD)
Behavior Driven Development (BDD)

- Tests are created for the functionality prior to its implementation.
- To ensure you are building the thing right.
THE TRIAD

Customer

Developer

Tester
No code goes in until the test goes on
It is easier to automate tests written before an implementation.
Behavior for Scenario (Flow)

Setup (Given)

Current State

Action or Event

Trigger (When)

Expected Results

Assert (Then)
Test for Scenario (Flow)

- **Setup (Given)**
- **Current State**
- **Trigger (When)**
- **Action or Event**
- **Assert (Then)**
- **Actual Results**
- **Expected Results**
**Scenario**

Given car speed is 30 mph  
When speed limit changes to 20 mph  
Then car speed becomes 20 mph

**Test**

Given car speed is 30 mph  
When speed limit changes to 20 mph  
Then CHECK car speed becomes 20 mph

BDD focuses on behavior which becomes test

ATDD focuses on test of desired behavior

Any other behavior to specify?
## Domain Terms with Tests

- **Domain term behavior**
  - Minimum /maximum values
  - Format (e.g., ###-##-####)

<table>
<thead>
<tr>
<th>Speed</th>
<th>Valid?</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>No</td>
</tr>
<tr>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>149</td>
<td>Yes</td>
</tr>
<tr>
<td>150</td>
<td>No</td>
</tr>
</tbody>
</table>
Business Rule

Speed limit varies in school zones
### Business Rule Test

#### Business rule data

<table>
<thead>
<tr>
<th>Time</th>
<th>Speed Limit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:44 AM</td>
<td>45 mph</td>
<td>Speed limit for road</td>
</tr>
<tr>
<td>7:45 AM</td>
<td>35 mph</td>
<td>School zone applies</td>
</tr>
</tbody>
</table>

#### Some tests for business rule

<table>
<thead>
<tr>
<th>Start Time</th>
<th>Stop Time</th>
<th>Speed Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:45 AM</td>
<td>8:45 AM</td>
<td>35 mph</td>
</tr>
<tr>
<td>2:45 PM</td>
<td>3:45 PM</td>
<td>35 mph</td>
</tr>
</tbody>
</table>
Test Driven Development (TDD)
Don’t test code, code to the test
Test Driven Development (TDD)

- Develop in small steps with tests
- Check at each step that implementation passes tests of previous steps
Design allocates responsibilities for passing external tests to internal entities (components, classes, etc.)
Test-Driven Development (TDD) cycle

1. Code the test
2. Check that test fails
3. Write code to pass the test
4. Check other tests pass
5. Refactor code if necessary
6. Check all tests pass
## Example of TDD

### Business Rule – School Zone

<table>
<thead>
<tr>
<th>Time</th>
<th>Speed Limit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:44 AM</td>
<td>45 mph</td>
<td>Speed limit for road</td>
</tr>
<tr>
<td>7:45 AM</td>
<td>35 mph</td>
<td>School zone applies</td>
</tr>
<tr>
<td>8:45 AM</td>
<td>35 mph</td>
<td>School zone applies</td>
</tr>
<tr>
<td>8:46 AM</td>
<td>45 mph</td>
<td>Speed limit for road</td>
</tr>
</tbody>
</table>
Another Example of TDD

- Speed Sign Processing
  - Sign head on with clean text
  - Sign side on with clean text
  - Sign head on with fuzzy text
  - Sign head on with missing letters
Not an Ending, But a Beginning
Layers

AS-IS

VALIDATE HYPOTHESIS (HDD)

STORY TESTS (BDD)

CODE TESTS (TDD)

APPLICATION

COMPONENT

ENABLER TESTS

INFRASTRUCTURE/ARCHITECTURE

TO-BE

APPLICATION

COMPONENT

CODE TESTS (TDD)

NEW CODE TESTS

NEW STORY TESTS

STORY TESTS (BDD)

ENABLER TESTS

INFRASTRUCTURE/ARCHITECTURE
Recap

- Describe how Shift Left Testing helps build in quality to foster continuous integration and continuous deployment
- Identify how loopbacks and delays are reduced by Shift Left Testing
- Identify the different aspects of Shift Left Testing
- Describe the collaboration in Shift Left Testing

Did this session pass your tests?
Go Forth and Shift Left Your Testing

ken@kenpugh.com
LinkedIn : kenpugh

Coming soon – presentation downloads at global.safesummit.com/presentations
Question and Answer

- Which CALMR was not covered?
- What challenges do you think you would face in adopting BDD/ATDD?
- What challenges do you think you would face in adopting TDD?
- What challenges do you think you would face in adopting HDD?
Please rate sessions

1. Click the Schedule icon in mobile app and locate the session

2. ‘Check in’ by clicking the plus sign next to the session title

3. Tap star rating at top of screen