Test-Driven Development in the Larger Context

### TDD & ATDD





20200304SB

### Scott Bain

**Senior Technical Trainer** 



#### Scott.Bain@PMI.org

Agile Development Emergent Design, Design Patterns, OO Analysis and Modeling, Test-Driven Development, Acceptance Test-Driven Development









PM Project

Management.com

Teal-driven development is a very powerful technique for analyzing, designing, and testing quality software. However, if done incorrectly, TDD can incur massive maintenance contrast the test software force. This is such a common Los 
 DUE
 A question that we are often asked its: "What is the difference between Acceptance Test Driven
 Development (TLD)?" These two activities are related by name
 but otherwise seem to have little to do with each other.

ATDD is a whole-seam practice where the team members discuss a requirement and come to an agreement about the acceptance criterio for that requirement. Through the process of accurately spechring the acceptance criteria — the acceptance test — the team Betters out the requirement, discovering and corrobarding the various assumptions made by the team members and lentifying an awarent phe various questions that, unanswered, would prevent the team fittees and lentifying the system. Circlettice is a subscription of the spectra members and lentifying the system. Circlettice is a subscription and the system circlettice is a subscription of the spectra members of the system.

The word acceptance is used in a wide sense here:

The customer agrees that if the system, which the team is about to implement, fulfills the

#### https://www.projectmanagement.com/blogs/654443/Sustainable-Test-Driven-Development

© Copyright PMI All Rights Reserved

Citizen I

### A Pertinent Joke

- A woman (schoolteacher) notices her husband (software developer) is heading out the door
- "Pick up a gallon of milk, and if they have eggs then get six"
- The husband comes back with 6 gallons of milk
- Asked why he replied "because they had eggs"



© Copyright PMI All Rights Reserved

### A Frequently Asked Question

# TDD? ATDD?



#### Test-Driven Development

"Developers writing unit tests before the production code"

True, but incomplete



#### Acceptance Test-Driven Development

DEFINITION -

"Acceptance tests being written before the development cycle begins"

Usually seen as a separate thing from what the developers do which is typically thought of as "TDD"

### They Are Both TDD

**DEFINITION** 

- Done differently
- By (somewhat) different people
- For different reasons
- But they synergize powerfully

TDD is a Software Development Paradigm

### Agenda ATDD vs. UTDD Differences:

- Audience
- Cadence
- Automation
   Synergy





#### Stakeholder

Someone with expectations of the system to whom we must listen

Anyone who's qualified to have an opinion of the system's correctness (or lack thereof)

There are many more than we often think



### **Stakeholders**

- End Users
- Sponsors
- Regulators
- EULA?

Etc...

- Legal Department
- Other teams
- Developers themselves





### AUDIENCE



### Audience ATDD

All Stakeholders:



PM

Business Analysts, Product Owners, Project Managers, Legal Experts, Developers, Marketing People, Testers, End-Users, Regulators, SME's, etc...



Spec

Test

PN

evelopment

### Audience ATDD

All Stakeholders:

Business Analysts, Product Owners, Project Managers, Legal Experts, Developers, Marketing People, Testers, End-Users, Regulators, SME's, etc...



Range: A2E6  Variable Names Row: 1 SELECTION patients.xls % A B			Output Type: Table Column vectors Numeric Matrix string Array Cell Array		UNIMPORTABLE CEL	LS Impor Selection MPOR	t m 👻		
	patients								
	LastName Text	Gender Categorical	Age Number -	Location Categorical	Height Number +	Weight Number -	Smol Text		
1	LastName	Gender	Age	Location	Height	Weight	Smoker ^		
2	Smith	Male	38	County Gen	. 71	176			
3	Johnson	Male	43	VA Hospital	69	163			
4	Williams	Female	38	St. Mary's	64	131			
5	Jones	Female	40	VA Hospital	67	133			
6	Brown	Female	49	County Gen	. 64	119			
7	Davis	Female	46	St. Mary's	68	142			
8	Miller	Female	33	VA Hospital	64	142	~		
/	< Sheet1						>		

© Copyright PMI All Rights Reserved



### Audience ATDD



PN

These different representations would imply different approaches to automate them

ortho meta public void foo(i) { while(i>0) Given: api(1/2); i--; When: Then:

	SELEC	пом	Column v	Column vectors		MPOR	Selection -	
	A B		String Ari	String Array		F	G	
				patients				
	LastName Text	Gender Categorical	Age •Number	Location Categorical	Height Number	Weight Number •	Smo Text	
1	LastName	Gender	Age	Location	Height	Weight	Smoker	
2	Smith	Male	38	County Gen	71	176		
3	Johnson	Male	43	VA Hospital	69	163		
4	Williams	Female	38	St. Mary's	64	131		
5	Jones	Female	40	VA Hospital	67	133		
6	Brown	Female	49	County Gen	64	119		
7	Davis	Female	46	St. Mary's	68	142		
8	Miller	Female	33	VA Hospital	64	142		
	<					l)	>	
	Sheet1							

### ATDD

### The Big Goal is: Collaboration Leading to a Shared Understanding





### Audience UTDD



PN

}

#### **Technical People**

```
@Test public void testAssetFirstYearStraightLineCalculation() {
      double anyValue = 70000.00;
      int anyTerm = 5;
      int yearToWriteOff = 1;
```

```
double straightLineResult = anyValue / anyTerm;
Asset testAsset = new Asset(anyValue, anyTerm);
assertEquals(straightLineResult,
       testAsset.getWriteOff(yearToWriteOff), .02);
```



PN

"... one of the things we see in a lot of [U]TDD projects is that projects go south on about their 3rd sprint, and they crash and burn because they cannot go any further."

Is this because UTDD is not sustainable?

JavaOne Conference, James O. Coplien:

No. It has to do with what you think TDD is, and how you've been trained to do it

### CADENCE



### Cadence ATDD

#### Part of Iteration Planning



© Copyright PMI All Rights Reserved

### ATDD Feeds Development



#### A Behavior is the "Unit" in a Unit Test

© Copyright PMI All Rights Reserved



### Cadence ATDD -> UTDD

Unfold behaviors onto the backlog





### Cadence ATDD -> UTDD

Draw one from the backlog







#### One at a time: Red, Green, Refactor





### Cadence UTDD & ATDD

#### **Behavior Comes From Acceptance**





### Cadence UTDD & ATDD

#### **Behavior Comes From Acceptance**





#### This is a false dichotomy

© Copyright PMI All Rights Reserved

### AUTOMATION

#### AND TEST ARCHITECTURE



### Automation ATDD

**Given:** A Sale of \$10000 **When:** The Discount is Calculated **Then:** The Result is 10 Percent







### Automation ATDD

**Given:** A Sale of \$10000 **When:** The Discount is Calculated **Then:** The Result is 10 Percent







### Automation ATDD

**Given:** A Sale of \$10000 **When:** The Discount is Calculated **Then:** The Result is 10 Percent





### Automation UTDD

Execution: Unit Test Specification and Binding API DB Sproc Etc...



PN

### Automation UTDD: Better Architecture





PM

### Automation UTDD: Feeding ATDD



### TDD: Two Adoption Paths

#### ATDD First

PM

- 1. Teach the organization to collaborate around acceptance test creation
- 2. Pick an automation framework for tests once competent
- 3. Developers write binding code for automation needs
- 4. Developers learn UTDD, reusing the the binding they created for ATDD: the test architecture is already in place

### <u>UTDD First</u>

### **TDD: Two Adoption Paths**

#### ATDD First

PM

- 1. Teach the organization to collaborate around acceptance test creation
- 2. Pick an automation framework for tests once competent
- 3. Developers write binding code for automation needs
- 4. Developers learn UTDD, reusing the the binding they created for ATDD: the test architecture is already in place

#### **UTDD First**

- 1. Teach the developers good design (patterns, etc...) and how to conduct sustainable UTDD
- 2. They also learn to architect their tests to separate Spec and Bind
- 3. The rest of the organization learns to collaborate using ATDD
- 4. Once the automation framework is selected, developers tune their existing binding layer to be reusable in ATDD automation

### SYNERGY



### Synergy: TDD

• Shared understanding of valuable behavior





PN

### Synergy: TDD

- Shared understanding of valuable behavior
- Knowledge captured persistently/verifiably, and allowed to be changed safely





PN

### Synergy: TDD

- Shared understanding of valuable behavior
- Knowledge captured persistently/verifiably, and allowed to be changed safely
- Promote high quality design and appropriate architecture





### Synergy: TDD

- Shared understanding of valuable behavior
- Knowledge captured persistently/verifiably, and allowed to be changed safely
- Promote high quality design and appropriate architecture
- Neither replaces traditional testing (QA)
- TDD promotes a healthy culture

PN



© Copyright PMI All Rights Reserved



© Copyright PMI All Rights Reserved



# Dev/QA

### Conclusions

ATDD and UTDD are part of the same approach: **TDD** 

They are done somewhat differently, by somewhat different people

They synergize powerfully







#### THANK YOU Scott.Bain@pmi.org

PM

© Copyright PMI All Rights Reserved