

# CS1632: Lecture 14

**Behavior Driven  
Development**

**Bill Laboon/Dustin Iser**

# What is Behavior Driven Development?

An evolution of TDD which focuses on the general expected behavior of the application as opposed to focusing on blindly meeting the exact specifications of a system based on requirements.

# BDD != TDD

While BDD and TDD are both test-first development methodologies, they are not the same thing.

TDD focuses on writing good code.

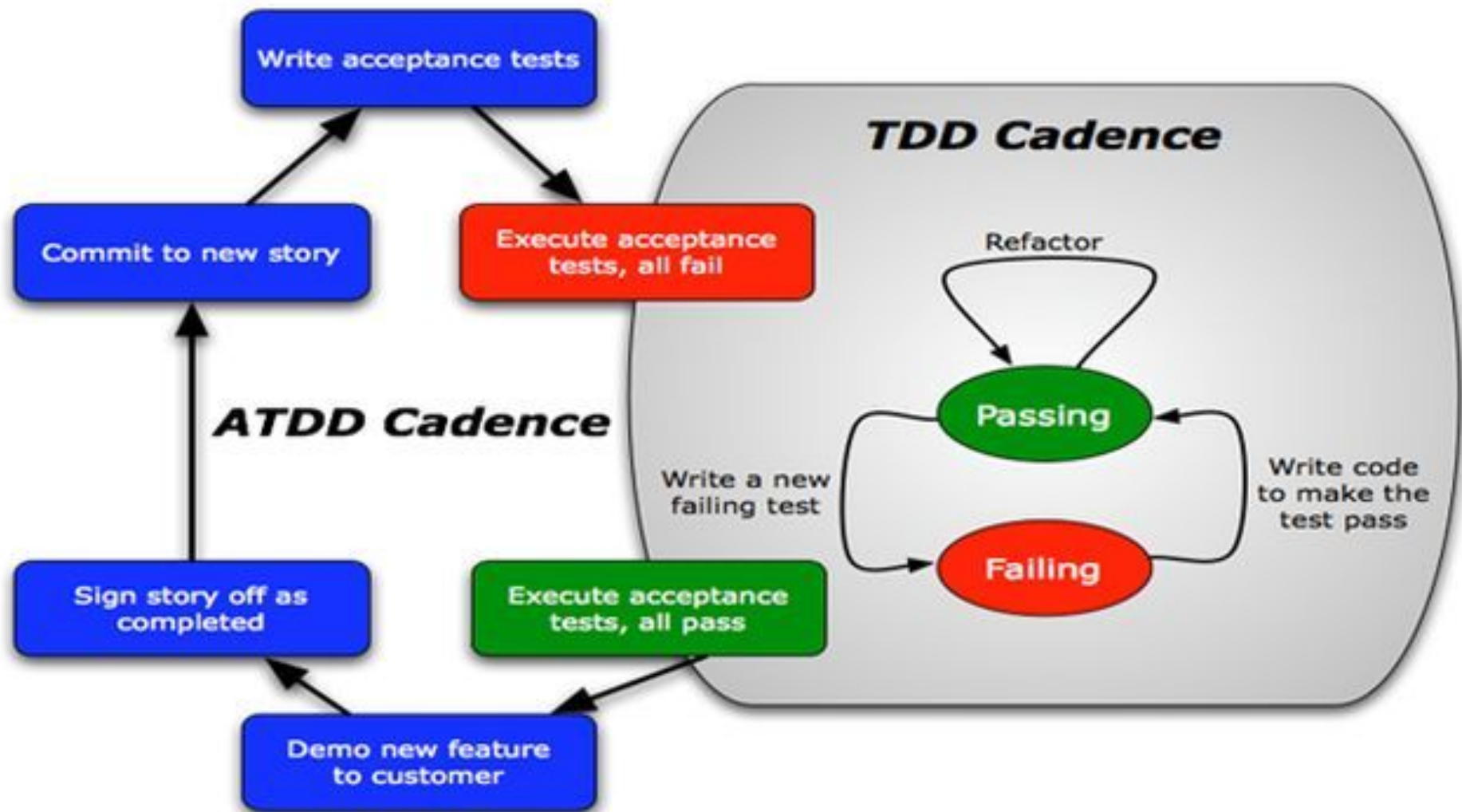
BDD focuses on building a good product.

# ATDD - The missing link between TDD and BDD

ATDD is “acceptance test-driven development”.

ATDD helps developers focus on a product as a whole by guiding development with a red-green-refactor loop.

The difference is in TDD, we wrote unit tests. With ATDD we write system level tests.



# BDD

BDD is the same as ATDD, but instead of using system level tests as our gatekeeper, we use tests written from the user's perspective.

# Requirement not written from the user's perspective

“The system shall enable the LOWTEMP warning light when two out of three internal thermometers agree that the ambient temperature is below -10 degrees Celsius ( 14 degrees F, 263 degrees Kelvin), as indicated by the INTHERM<sub>1</sub>, INTHERM<sub>2</sub>, and INTHERM<sub>3</sub> registers.”

# The user story

As a <persona>

I want to <perform an action>

So that <reason / benefit>



# Examples

As a systems administrator

I want to create users

So that users in my domain can log in

As a user

I want to see my bank account balance

So that I know how much money I have

# Testing User Stories

User stories are purposely ambiguous.

The focus of the development team should be on building a product that thrills its customers. Not on blindly following requirements.

The key to building tests for user stories is communication.

The tester's ability to empathize with the customer becomes paramount.

Common in an agile software development environment.

# Scenarios

Scenarios are particular instances of a user story.

As a user

I want to log in

So that I can access my banking account

Scenario 1: I log in with correct username and password

Scenario 2: I log in with correct username, but incorrect password

Scenario 3: I log in with incorrect username

# Given / When / Then

We can use the “Given / When / Then” template for scenarios.

Just like our test cases and our unit tests: preconditions, execution steps, and postconditions.

**Given** a correct username

And an incorrect password

**When** I try to log in with those credentials

**Then** I should receive an error message with “incorrect password entered” on it