Behavior Driven Development

Location: Bloomington,

Date: 26th Feb, 2018,

Author: Venkata Raghavendra Talluri

What is BDD..?

Behavior Driven Development(BDD) is a Software development process that emerged from Test Driven Development(TDD) and Acceptance Test Driven Development(ATDD).

What Is TDD:

- Relies on the repetition of a very short development cycle.
- First the developer writes an (initially failing) automated test case that defines a desired improvement or new function.
- Then produces the minimum amount of code to pass that test, and finally refractors the new code to acceptable standards.

What is ATDD:

- Development methodology based on communication between the business customers, the developers, and the testers.
- Acceptance tests are created when the requirements are analyzed and prior to coding.
- Tests and requirements are interrelated.

Continue…

What is BDD:

BDD was developed by Dan North as a response to the issues encountered in TDD.

- Where to start in the process
- What to test and what not to test
- How much to test in one go
- What to call the tests
- How to understand why a test fails

At the heart of BDD is a rethinking approach to the Unit Testing and Acceptance Testing that North came up with while dealing with these issues.

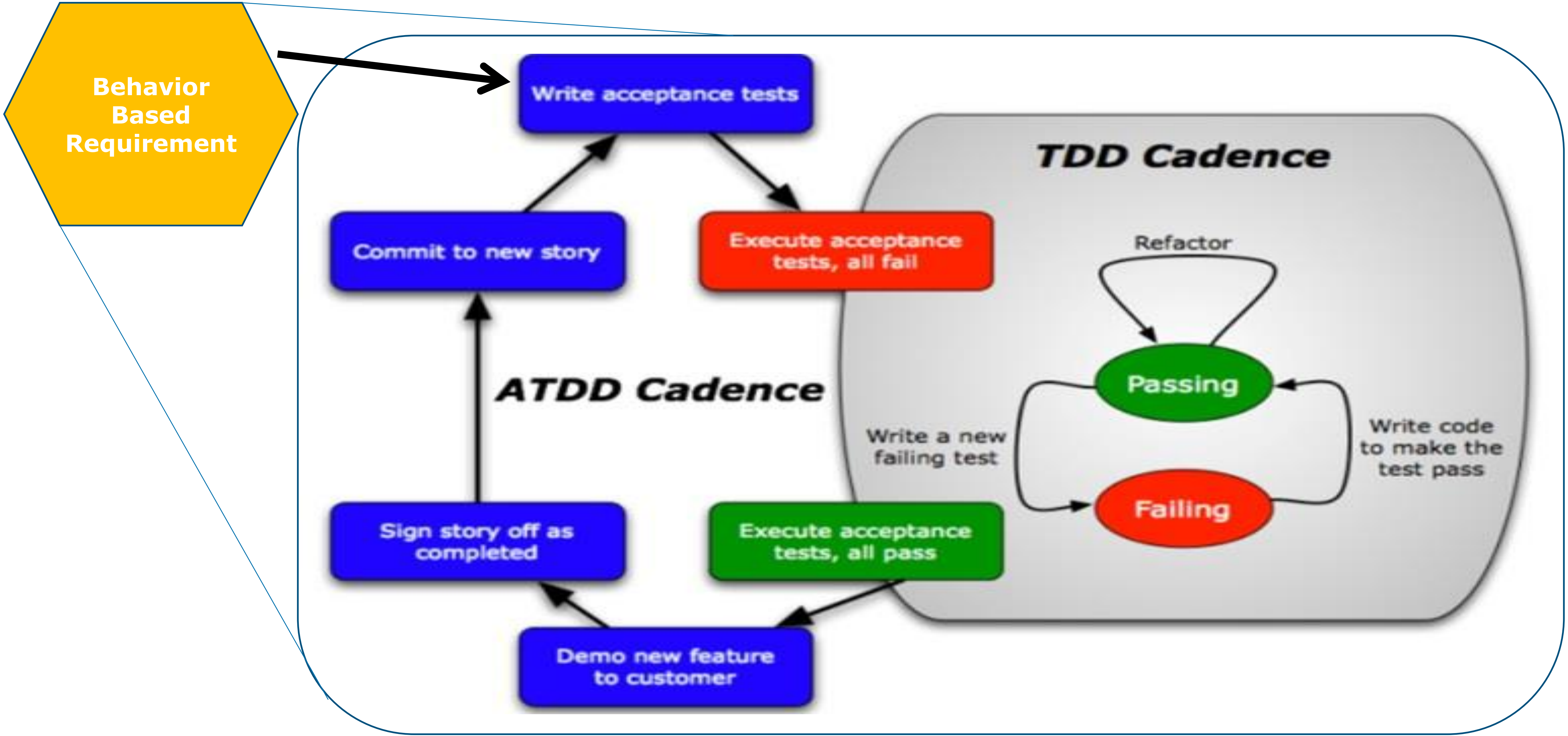
Why BDD:

- Ability to write test cases(scenarios) and acceptance criteria before performing any development.
- Communication between business and development is extremely focused as a result of common language.
- Business needs tie directly to the code that is written.
- Stories are easy to “groom” – breakdown, task and plan.

Continue…

- Provides a common domain language, which everybody across the board(developer, tester, BA and Business) understands.
- Notations originating in the BDD approach, in particular the given-when-then canvas, are closer to everyday language.

TDD-ATDD-BDD



Behavior Based Requirement

Write acceptance tests
Commit to new story
Execute acceptance tests, all fail
TDD Cadence
Refactor
Passing
ATDD Cadence
Write a new failing test
Write code to make the test pass
Failing
Sign story off as completed
Execute acceptance tests, all pass
Demo new feature to customer

BDD Process

- Identify a Feature that the system needs to have.
- Describe the Feature in terms of **Who** needs **What** is needed and **Why** they need it.
- Describe specific scenarios of how the system is to behave while providing this feature.
- Automate scenario steps by writing test methods and map them to scenario steps.
- Make the system behave as described by the requirements by writing software.
- Run the stories to validate the system.

BDD Behavioral Specifications Process

Identify a Feature that the system needs to have.

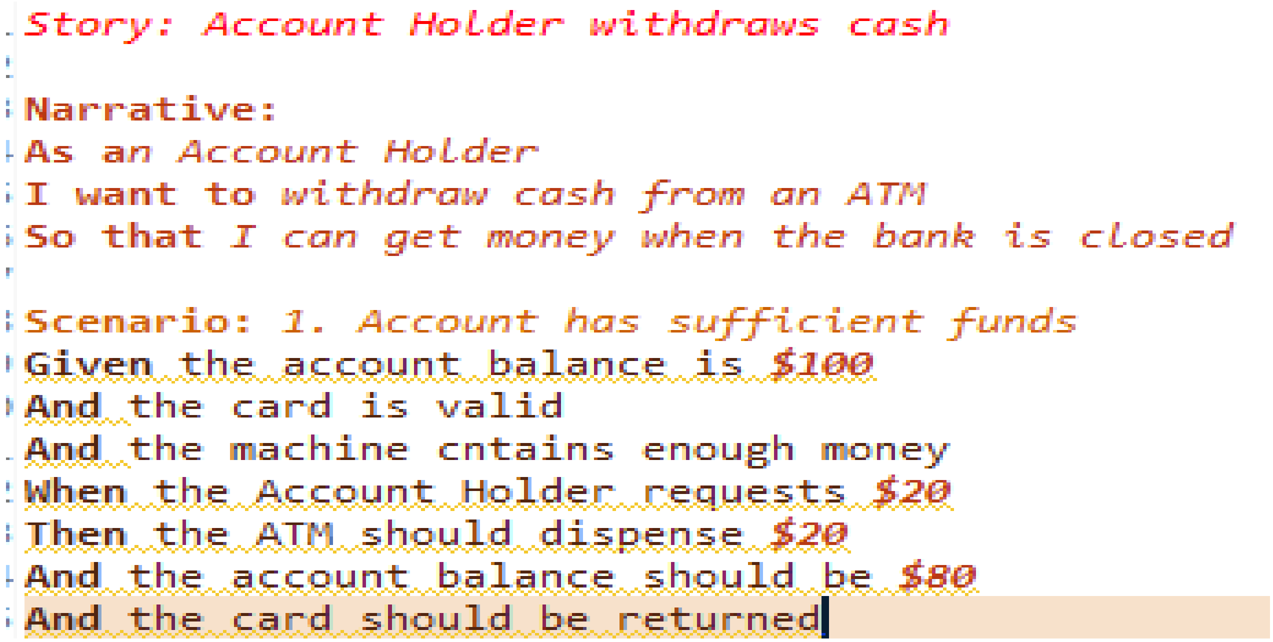
- BDD Specifies that business analysts and developers should collaborate in this area and should specify behavior in terms of user stories, which are each explicitly written down in a dedicated document.

Each user Story should, in some way, follow the following structure:

- **Title:** The Story should have a clear, explicit title
- **Narrative:** A short, introductory section that specifies
  - **Who** (Which business or project role) is the diver or primary stakeholder of the story (the actor who derives business benefit from the story)
  - **What** effect the stakeholder wants the story to have
  - **What** business value the stakeholder will derive from this effect
- **Acceptance criteria of scenarios:** A description of each specific case of the narrative. Such a scenario has the following structure:
  - It starts by specifying the initial condition that is assumed to be true at the beginning of the scenario. This may consist of a single clause, or several.
  - It then states which event triggers the start of the scenario.
  - Finally, it states the expected out come, in one or more clauses.

Continue…

- In 2007 Dan Suggested a template for textual format which has found wide following in different BDD software tools.



```
Story: Account Holder withdraws cash

Narrative:
As an Account Holder
I want to withdraw cash from an ATM
So that I can get money when the bank is closed

Scenario: 1. Account has sufficient funds
Given the account balance is $100
And the card is valid
And the machine cntains enough money
When the Account Holder requests $20
Then the ATM should dispense $20
And the account balance should be $80
And the card should be returned
```

- This format is referred to as the Gherkin language, which has a syntax similar to the above example.
- The term Gherkin, however, is specific to the Cucumber, JBehave, and Behat software tools.

how
why
when
which
how
which
how
why
how
when
where
what
where
which

People matter, results count.

This message contains information that may be privileged or confidential and is the property of the Capgemini Group.

About Capgemini

A global leader in consulting, technology services and digital transformation, Capgemini is at the forefront of innovation to address the entire breadth of clients' opportunities in the evolving world of cloud, digital and platforms. Building on its strong 50-year heritage and deep industry-specific expertise, Capgemini enables organizations to realize their business ambitions through an array of services from strategy to operations. Capgemini is driven by the conviction that the business value of technology comes from and through people. It is a multicultural company of 200,000 team members in over 40 countries. The Group reported 2016 global revenues of EUR 12.5 billion.

Learn more about us at

www.capgemini.com