Importance of ATDD in Agile Development

Sudhish Koloth

Jan 2021

Abstract: Culture of Acceptance Test Driven Development as part of Agile Software development methodology helps to improve code quality

Keywords: Agile, TDD, ATDD, Test Driven Development, Code Quality, Defect Reduction, Coding to Requirement, Coding Standards, Defect Cycle, Solving Over Engineering, Testing, Junit.

Author

Sudhish Koloth is a Lead developer working with a Banking and Financial services company. He has spent over 13 years working in information technology. He worked in various technologies including full stack, big data, automation, and android development. He also played a significant role in delivering important impactful projects during COVID-19 pandemic. Mr. Sudhish Koloth uses his expertise to solve common problems faced by humanity and a volunteer for non-profit applications and providing help. Mr. Sudhish Koloth is also a mentor who helps fellow professionals and colleagues, technically, with his expertise. Mr. Sudhish is also an active preacher and motivator of Stem education's importance to School Kids and Young college graduates. Mr. Sudhish Koloth was recognized for his work inside and outside of his career network.

Introduction

ATDD - Acceptance Test-Driven Development is a practice where developers, testers, and product owners collaborate and write test conditions to satisfies the given acceptance criteria of an Agile story.

Use Cases

- A practice used in software development cycle.
- A standard to enhance best practices.
- A technique to make Test Driven Development more efficient.
- A plan to meet the product requirement and implementation more closer

Examples of ATDD

Now, let us consider an example, suppose you are building a simple Weather Application, where the requirement is to enter the city and time then it should display current weather. Now as part of Agile story intake, you have a simple story created for this feature. The story description will like - 'Implement logic to fetch weather based on city and time'. Now as part of the story writing process, we have to define acceptance criteria. The simple acceptance criteria would be:

1. Should display current weather when city and time is entered.

2. Should display some error message when city and time are entered incorrectly.

Now, as part of the next step, we can now introduce ATDD to this story, here the developers, testers, and product/business owner have a collaboration session and list down all ATDD criteria for this story which can then be leveraged to do Test Driven Development.

For the above example, there can N number of ATDD that can be listed depending upon agreement between the product owner and the team. We can list a few in gherkin format. Gherkin format is basically writing test cases in the Given When Then condition.

```
Feature: Fetching weather details

# The first example can be positive scenario
Scenario: Retrieving weather details when city and time details are available
Given the city is "Houston" and time is "8.05 PM EST"
When  the weather service is called
Then output should return current weather

# The second scenario can be negative one if not city details are provided
Scenario : Return error message when no city is provided
Given the time is "9.05 PM EST"
When the weather service is called
Then output should return error message

#The third scenario can be negative one if time is not provided
Scenario: Return error messages when no time is provided
Given the city is "Philadelphia"
When the weather service is called
Then  output should return error message

#The fourth scenario can be negative when both time and city are not provided
Scenario: Return error message when no city and no time are provide
Given city and time are null
when the weather service is called
Then output should return error message
```

As you see, we developed at-least four scenarios that needed to be tested to fully satisfy the acceptance criteria of the story. Note, here we have four scenarios, but there can be many scenarios that can be listed. So once we finalize all the scenarios, we can now attach this Gherkin file to the story card

What happens next?

After sprint planning is done, and once the story is ready to implement stage, the developer who is picking this story to work, now knows exactly what this is story is about and what are all test cases to be fulfilled.

As part of his development, he now starts adopting Test Driven Development to adopt all those gherkin test use cases. The developer can start with either cucumber or JUnit or any testing framework and develop the code.

```
@Test
public void testGivenNoCityAndNoTimeWhenWeatherServiceIsCalledReturnError() {
  //given
  String city = null;
  String time = null;
  //when
  WeatherService service = new WeatherService();
  //Then
  String output = service.fetchWeather(city, time);
  assertEquals("Failed to fetch", output)
}
```

For instance, a developer can start with a Junit class, which can be called WeatherServiceTest.java, and start building the code. Here is one sample test case to start with, if you observe we started with a negative test case from Gherkin, it is one good way to start developing code.

The developer now then writes the minimum code to make this test pass. Then developer moves to the next gherkin test case, and finally builds the entire code.

Benefits

Now we have seen what ATDD and how to embed that in Agile practice, the main advantage of following this process is the ability to identify all acceptance test cases that can make this story feature complete. It is a very great practice to introduce in the team and combining this with Test Driven Development will surely improve code quality, minimum defects, and easier rollouts of features.

References

Ulf, U. E. (2016, October 31). 7 Things About Acceptance Test Driven Development Everyone Should Know. Https://Reqtest.Com/. https://reqtest.com/testing-blog/acceptance-test-driven-development-2/

agilealliance. (n.d.). Acceptance Test Driven Development (ATDD). Https://Www.Agilealliance.Org/. https://www.agilealliance.org/glossary/atdd