Adaequare Software Engineering

Software Test Engineering

Test Driven Development (TDD)/Behavior Driven Development (BDD)Using Adaequare's Action Driven Testing Framework Tool (ADTF)

ᄂ|ADAEQUARE

Table of Contents

Executive Summary

Humans are social animals and this is one of the major reasons why we evolved into one of the most intelligent species on this planet. Social interactions bring forth learning and collective wisdom. This is one of the hallmarks of Agile/Scrum in software development and has revolutionized the way we perceive, develop and test software. There are many variants to Agile/Scrum and the ones which are gaining prominence are Test Driven Development (TDD) and Behavior Driven Development (BDD).

What is TDD?

Test-driven development is related to the test-first programming concepts of <u>extreme programming</u>, begun in 1999, but more recently has created more general interest in its own right.

Test-driven development (**TDD**) is a <u>software development process</u> that relies on the repetition of a very short development cycle: first the developer writes an (initially failing) automated <u>test case</u> that defines a desired improvement or new function, then produces the minimum amount of code to pass that test, and finally <u>refactors</u> the new code to acceptable standards. <u>Kent Beck</u>, who is credited with having developed or 'rediscovered' the technique, stated in 2003 that TDD encourages simple designs and inspires confidence.

What is BDD?

Behavior-driven development (abbreviated **BDD**) is a <u>software development process</u> based on <u>test-driven development (TDD)</u>. Behavior-driven development combines the general techniques and principles of TDD with ideas from <u>domain-driven design</u> and <u>object-oriented analysis and design</u> to provide <u>software developers</u> and <u>business analysts</u> with shared tools and a shared process to collaborate on software development. Behavior-driven development was developed by Dan North as a response to the issues encountered teaching <u>test-driven development</u>:

The Key Principles of TDD/BDD

At its core, both TDD/BDD focuses on behavioral specification of software units.

- Define a test set for the unit *first*;
- Then implement the unit;
- Finally verify that the implementation of the unit makes the tests succeed.

This definition is rather non-specific in that it allows tests in terms of high-level software requirements, low-level technical details or anything in between. The original developer of BDD (Dan North) came up with the notion of BDD because he was dissatisfied with the lack of any specification within TDD of what should be tested and how. One way of looking at BDD therefore, is that it is a continued development of TDD which makes more specific choices than TDD.

Behavior-driven development specifies that tests of any unit of software should be specified in terms of the desired behavior of the unit. Borrowing from agile software development the "desired behavior" in this case consists of the requirements set by the business — that is, the desired behavior that has business value for whatever entity commissioned the software unit under construction. Within BDD practice, this is referred to as BDD being an "outside-in" activity

What is ADTF?

Action Driven Test Automation Framework (ADTF) is a portable software testing automation framework for web applications. It is developed using Selenium2, TestNG, ReportNG and J2EE to fulfill all the automation needs of TDD/BDD based testing.

ADTF combines the key elements of TDD/BDD into a single framework. The desired behavior of a software unit as described in the key principles of TDD/BDD is captured as actions in ADTF from requirements written in English language.

- ADTF tooling reads a specification document.
- ADTF directly understands completely formal parts of the ubiquitous language   Based on this, the tool breaks each scenario up into meaningful actions.
- Each individual clause in a scenario is transformed into some sort of parameter for a test for the user story. This part requires project-specific work by the software developers.
  - The ADTF framework then executes the test for each scenario, with the parameters from that scenario.

Build and Run Test Automation   Manage Test Automation   Reporting and Analysis

Features and Benefits of ADTF

Why	Features	Benefits
Next generation generic automation framework	The best of Selenium and TestNG are incorporated in ADTF	Web Interface for test case management and execution
Recording	XML driven framework	Web based reporting
Easy to manage objects and data	Extendable and easy to maintain	Faster execution
No need to customize the script	Customizable Test execution reports	Extendable
Easy to create and maintain test suites	Supports automating internationalization testing	Maintainable
Easy to manage test cases		
Change management and Version maintenance made easy		
Multi browser, Multi-OS testing		

About Adaequare

Established in 2001, Adaequare is a CMMi Level 3 Certified software services company specializing in product development, data and test engineering. We assist IT teams in delivering high performing solutions to the business with faster time to market and high returns on investment.

With a global presence across 3 different locations, we focus on medium to large businesses and encapsulate consulting, results driven engagement and delivery models which are important to our target customers.

We work with several companies in USA such as CoreLogic, Tria Beauty and have been either a dedicated IT partner or a preferred IT partner. We have offshore delivery centers in India that support our customers in product development, Tier 1 and 2 support and testing with over 300+ resources.