**Test Driven Development (TDD)**

TDD is a software development methodology in which test cases are created to define and validate what the code will accomplish. Simply said, test cases for each capability are generated and tested first, and if the test fails, new code is produced to pass the test while keeping the code simple and bug-free. Test-Driven Designing and building tests for each little capability of an application is the first step in development. TDD requires developers to build new code only if an automated test fails. This prevents code duplication. TDD stands for test-driven development. The basic idea behind TDD is to build and correct failing tests before developing new code (before development). This helps to reduce code duplication since we develop a tiny piece of code at a time to pass tests. (Tests are nothing more than requirement conditions that must be tested in order to be fulfilled.) Test-Driven development is the technique of creating and executing automated tests prior to the actual development of an application. As a result, TDD is also known as Test First Development. TDD (Test Driven Development) is a software development approach that focuses on establishing unit test cases prior to developing actual code. It is an iterative process that involves development, unit testing, and refactoring. The TDD technique is based on the ideals of the Agile manifesto and Extreme programming. The test process, as the name implies, drives software development. Furthermore, it is a structural method that allows developers and testers to create efficient code that is long-lasting. TDD begins with developers building tiny test cases for each feature based on their first knowledge. The primary goal of this technique is to only modify or write new code if the tests fail. This avoids the duplication of test scripts. Agile development necessitates regular feedback in order to produce the desired product. Agile development can also be referred to as Feedback Driven Development. The project requirements are likely to vary during the development sprint cycle. To cope with this and produce solutions that are aligned with the changing needs of the customer, teams require frequent input to prevent delivering unsuitable software. TDD is designed to provide such feedback early on. TDD's test-first strategy also aids in the reduction of major bottlenecks that impede software quality and delivery. The system develops to guarantee that everything functions as planned based on continual input, issue patches, and the inclusion of new features. TDD improves communication among members of the development and QA teams, as well as with the client. Furthermore, because the tests are pre-created, teams do not need to spend time developing lengthy test scripts. Test-Driven Designing and building tests for each little capability of an application is the first step in development. TDD requires developers to build new code only if an automated test fails. This prevents code duplication. TDD stands for test-driven development.

الاسم: شهد فالح الدوسري عدد الكلمات: 477 كلمة

الرقم الجامعي: 441051140