Tel: 086 100 0248 / int: +27 31 764 5233
Fax to Email: 086 513 4755
Email: info@chillisoft.co.za
www.chillisoft.co.za

Unit 4 | St Helier Office Park | 1 Valdean Road
Gillitts | KwaZulu-Natal | South Africa | 3610

Company Reg No.: 2005/000255/07 | VAT No: 4770225821

TDD – Evolving Development

Since the dawn of the computer era, developers have been writing and testing code in a never-ending cycle until the objective of the software has been achieved or delivered. This, understandably, can be a very time consuming and frustrating journey, often interfering with deadlines and the like.

In most instances, developers write code and successfully move it to production without any glaring problems. However, sometimes, after code has been moved to production, an error occurs and the developer is left fixing holes that seemingly keep creating more errors.

Test-driven development, or TDD, is a programming technique that requires a developer to write actual code and automated test code simultaneously in a bid to avoid any future issues. This also guarantees the testing of code can be done quickly and easily, without having to go back to the drawing board.

Ultimately, TDD eliminates excuses and encourages confidence.

Here's how it works -

When a program has been developed using TDD, it allows for the changing and testing of code more efficiently by means of automated tests.

If the coding passes all automated tests, the code is good to implement; if not, the developer can easily backtrack and pinpoint the affected code.
What this means is, the developer can quickly find which exact parts of the test failed, pinpointing which part of the changes it broke, making fixing the bugs and niggles easier.

TDD revolves around a short iterative development cycle that starts before any implemented code is written.

- First, automated test code is created, taking into account any and all possible inputs, errors and outputs.
- Since there's now already an automated test, as long as the code fails the testing process, it means that it's still not ready and the code will need to be fixed until it passes all assertions.
- Once the code passes the test, it can be cleaned up.
  As long as the code still passes the test, you no longer have to worry about changes that introduce new bugs.
- And should your program require new functionality, the writing of new tests is made dramatically easier. The developers need only rinse and repeat the procedures and start the cycle all over again further building confidence by reducing the build-test cycle time.

TDD ensures that the testing is done and the end result is thoroughly de-bugged. In truth, TDD is the future of quality development, which shows that the time spent in the initial phases, results in timesaving down the line.