Chillisoft
Hot Software | Innovative by Design
Powered by Passion

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 – Making Small Work of Large Projects

A lot has been said about the effectiveness of TDD and its inherent need in the development space. And yes, like anything, test-driven development has its positives and negative aspects. However, one of those positives is its ability to reduce time frames for big development builds but reducing troubleshooting with automated testing. And that, for us at least, is a major plus point.

That's not to say TDD won't work for smaller projects too. TDD practices can be effective whether for one person creating a small website or for a million-line application with a team of 70 programmers, because it solves problems before they actually occur, by forcing the developer to apply specific focus on the expected end result and then in developing solutions to achieve it.

This way of thinking ensures the design is simpler and ultimately more practical, reducing the quantity of bugs and improving the product's quality – critical for larger builds that require extensive testing.

When compared to more traditional approaches, where code is only tested after its deployment, the time needed for maintenance resulting from unexpected errors, is notably less.

TDD provides code deployment in a more flexible and modular way, also promoting development through layers – which for larger projects aids in "downscaling" in the workload.

It also forces the developer to think about his or her code in small chunks, which might be tested in isolation or independently of other strings. As a result, it becomes easier to deploy.

The fact that more work done upfront saves us time later on - and results in an end product with less bugs and shorter maintenance periods – means we can focus on delivering more quality, modular, flexible and easy to maintain solutions that meet the client's exact requirements.

Test-driven development is most certainly not the solution to all problems. But it does help developers improve the quality of their programming and save them time. And if there is one thing that we can all do with more of – it's time.