Test Driven Development
GUI Tests with GUI\textit{dancer}®

Product Demo at

eclipse Summit

October 10th - 11th
Ludwigsburg, Germany
Goals

Test development parallel to SW development
Test creation, reuse and adaptation must be quick and easy

No programming
Takes too much time, generates more code, strain on resources

Support creativity of testers

Usable for regression and integration tests

Dramatically reduce the cost of testing
GUIdancer Features

- Support for Java Swing, RCP, SWT and Web
- Multi user support
- Multilingual tests
- “Building block” approach
  Test Steps specified with Component, Action, Parameter
  Test Steps combined to form Test Cases of any depth
GUIdancer Features

Reusability
Test Cases form reusable libraries
Ease of modifying data or components in reused Test Cases
Abstract components, i.e. "Component with text" vs. Label

Robustness
Intelligent object-based recognition

Easily maintainable
Building block approach results in well-structured tests
Central change updates whole test
Demo

The demo is run using GUI\textit{dancer}

The AUT is a simple calculator as a SWING implementation

Later we’ll see that we can reuse the same test for SWT and WEB
GUI Tests

Demo

GUI dancer to create and execute a test on the simple adder

AUT

- Runs demo as test on GUIDancer
- to create and execute a test on the simple adder
Summary

Tests are easy to specify, reuse and maintain

Supports agile processes, test development parallel to SW development

Complex tests are just as easily created as small tests

Building blocks can be combined and reused as often as necessary

“One test fits all“

Test different implementations, languages, data, components, toolkits – all with the same test.

www.guidancer.com