Product Description

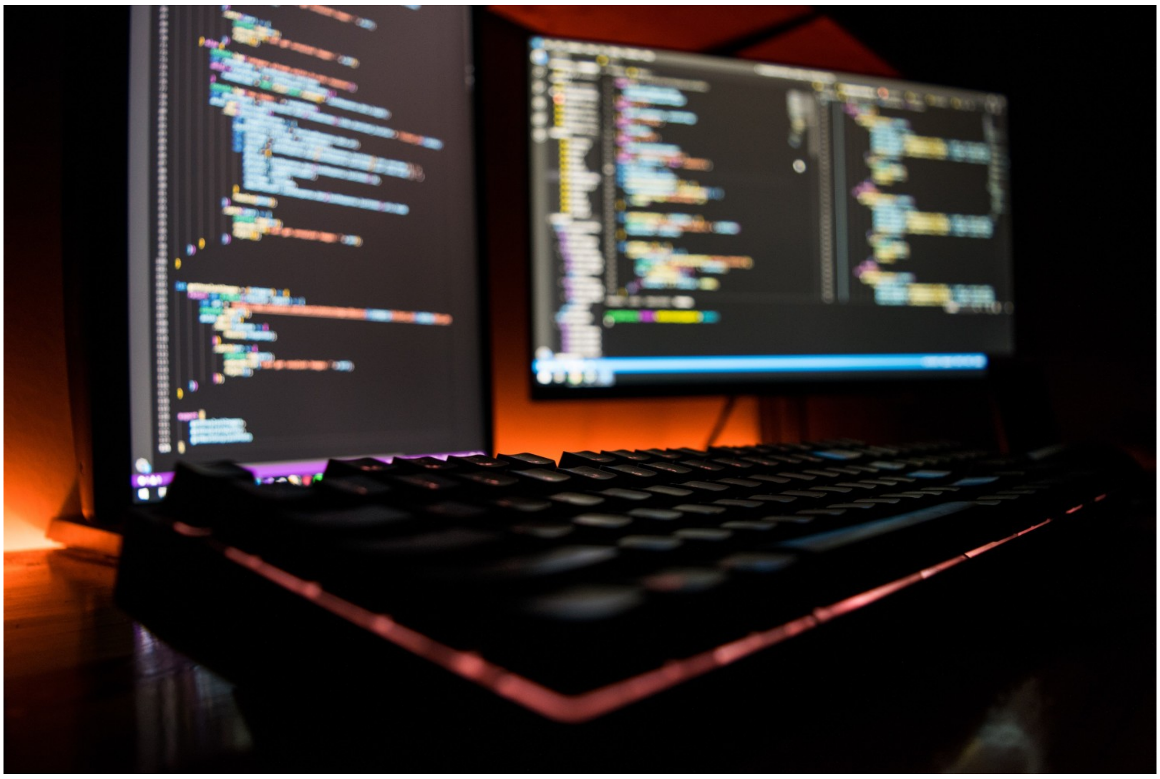
Software development with TDD and Unit Test Cases

8-lessons on how to develop Reliable Embedded Systems Software

Test Driven Development (TDD) and Unit Testing (UT) are the basis of a software development path which leads to reliability and stability of software products. Developing software for embedded systems is believed to be among the most unpredictables engineering practices. Is it possible to:

- produce predictable results?
- deliver consistently?
- avoid endless maintenance of products?

Actually, error-free software is a realistic target for any team. In this 2-module training, Bluewind explains the rules of TDD and how to build UT-Unit Test Cases for reliable and maintainable code base.



Purpose

Software Development with TDD and Unit Test Cases training offers a set of knowledge and best practices that sum up on tens man/years experience. It enables engineers to develop software in a highly comfortable and dependable way. Quality and development pace are no longer opposing needs, and maintenance after delivery is a viable option because each line of code has been developed with a Test Case in mind and a precise knowledge of how to stress each feature before delivering.

Outcome

After this training the attendee will have a deep knowledge of all practices and tools in use today for performing a TDD based software development on embedded devices. Moreover a set of solutions to special situations where standard Unit Test frameworks are not applicable will be touched. Starting from the software development process, the training will also give extensive knowledge about how to measure the quality of the development process and of the software deployed. Overall at the end of this course a good software developer will become a conscious engineer with capabilities that go far beyond solving problems, being able to guarantee that all solutions are robust and future proof.

Module 1 – TDD Test Driven Development

This module provides a practical introduction to TDD as a development process, and a number of examples, best practices and useful tools and tricks for the embedded world. Writing test first will become the natural way after this module.

Lesson 1: TDD Introduction

- Test Driven Development as a Workflow
- Unit Testing
- Test Doubles: Stub and Fake
- Framework: Ceedling
- Exercise

Lesson 2: TDD in practice

- Doubles: Mock
- Black-box testing
- Dual Target benefits
- Code Coverage report tool
- Test framework integration with Visual Studio Code
- Techniques for embedded development
- Exercise

Lesson 3: TDD Techniques

- Testable code drawbacks
- Doubles: linker vs runtime
- Techniques for embedded development
- Pure functions VS Side effects
- Exercise

Lesson 4: Other test techniques

- Static testing
- White-box testing
- Debugger with Unit Test
- Exercise

Module 2 – Unit Test Cases

This module shows how to apply TDD and Unit Testing for enhancing the quality and stability of existing and new software on embedded devices. Not only a new way for writing error free software, but also a toolbox for the software maintainer (Module 1 is prerequisite).

Lesson 1: Testing the surface

- API Testing
- Integration testing
- Exercise

Lesson 2: Testing the untestable

- Legacy: testing
- Snapshot testing
- Exercise

Lesson 3: Finding bugs

- Human code review
- Pair programming
- Memory sanitizers
- From bug to test
- Exercise

Lesson 4: Hardening

- In-target testing
- Fault injections
- Fuzz test
- exercise

Bluewind Srl

Via della Borsa, 16/A – 31033
Castelfranco Veneto (TV) Italy
+39 0423 723431 - info@bluewind.it