

Agile for Embedded C++

Course category	Agile for Embedded
Training area	Agile for Embedded
Course code	AEC++-401
Duration	4 days
Additional information	This is a new course in development and subject to minor changes. Please contact us for course availability and prices

Agile for Embedded Training Course

Since the publication of the Agile Manifesto back in 2001, Agile practices have become mainstream in software development with processes such as Scrum and techniques such as Test-Driven Development (TDD) becoming common practice.

However, almost all current books and training on Agile ignore the challenges of applying it in an embedded C++ environment. Aspects such as aligning Agile with hardware and mechanical teams, testing on the target, hardware dependency and managing both host and cross-compiler toolchains are generally ignored or oversimplified.

The course attempts to provide a pragmatic introduction to both Agile practices, such as Scrum, and techniques, such as TDD, within the context of Embedded Systems. Attendees perform practical exercises using both host (Linux / g++) and cross-development (Cortex-M / arm-none-eabi-g++) toolchains supported by a modern build system (SCons) and configuration management (Git) tools.

The course also looks at modern Continuous Integration (CI) with Docker containers.

Course objectives:

- To understand the principles of Agile Software Development
- To appreciate Agile for Embedded is different from Agile for non-Embedded
- To identify the required tools required to support Agile Software Development
- The gain practical experience of applying Agile principles to an Embedded system
- The challenges and pitfalls of applying TDD to an embedded system

Delegates will learn:

- How to cherry-pick appropriate Agile practices
- The principles of Scrum and Lean

- How to practice Test-Driven Development with version control (Git)
- How far Continuous Integration (CI) can be applied to embedded software development
- Why moving to 'Agile' can be difficult

Pre-requisites:

- Attendees should have a good working knowledge of the C++ language
- Have experience of developing software for embedded systems
- Understand the build process (preprocessor-compiler-linker-executable)

Who should attend:

The course is designed for software engineers working on an embedded C++ project. The target audience is engineers working in, or looking to move to, an Agile project environment (e.g. Scrum) and who want a pragmatic introduction to Agile process and techniques in the context of writing software for embedded systems.

Duration:

Four days

Course materials:

- Delegate manual
- Delegate workbook
- Delegate bootable Linux datakey with all tools installed

Course workshop:

Attendees perform a mixture of hands-on programming for both host and embedded targets. Approximately 50% of the course is given over to practical work.

The host code is developed on Linux using g++ and Scons supported by Git versioning. The embedded code is targeted at an ARM Cortex-M based MCU using arm-none-eabi-g++.

As part of the course, students are introduced to Docker for build isolation and cloud-based build systems, such as Travis-CI.

This combination gives attendees a real sense of the challenges of applying Agile development to an embedded application.

Introduction to Agile

- Manifesto
- Principles
- The Agile Onion

Agile Assumptions and Embedded Systems

- Embedded System continuum
- New Product Development (NPD)
- Hardware/Software/Mechanical Teams

Agile Process

- Foundations
- Scrum
- Lean
- Water-scrum-fall
- Scrum-ban
- Push vs Pull workflow

Agile Techniques

- Core
- Middle layer
- Outer layer

The Embedded Agile Toolset

- Dump the IDE
- Toolchains
- Test framework
- Version control
- Make tools
- Build servers
- Build environment
- Target Download
- Debug tool
- Metrics
- Markup
- Team workflow

Pair-Programming

- Principles

- Benefits
- Problems / Issues
- Pairing variations

Design Principles

- Abstraction and encapsulation
- SOLID

TDD for C++

- TDD foundations
- Test construction
- Unit isolation
- Testing APIs

Refactoring

- Principles
- Techniques
- Editor / IDE support

Versioning

- From centralised to distributed
- Workflows
- Self-hosted vs. cloud-based repositories

Continuous Integration

- Goals
- Architecture
- Isolation of application from platform
- Build systems
- Workflow and builds

Why is 'Agile' so difficult?

- Common Agile [Scrum] pitfalls
- Dark Scrum
- Flaccid Scrum

Feabhas Ltd - UK Office 15-17 Lotmead Business Park, Wanborough, Swindon, SN4 0UY, UK ·

+44 (0) 1793 792909 · info@feabhas.com · www.feabhas.com