Technical Practices: TDD-2

Continue your journey toward mastery of Agile Technical Practices at an advanced level, building on what you learned in the fundamental course, learning about outside-in software design, practising more-advanced styles of Pair Programming, applying a mix of TDD techniques,...

Please note: This is an advanced course, therefore, our course 'Technical Practices: TDD-1' or a similar course is a pre-requisite!

The course is for software developers interested in learning or improving their knowledge of effective technical practices. The classes can be offered in various programming languages for the demos and the exercises: contact us for details. Please note: These are primarily hands-on courses, and you will need your computer with your development environment of choice.

The training material is in English.

Your Take-Aways

- understanding in more detail modern concepts of agile technical practices
- master the practices of Test-Driven Development (TDD), Pair Programming, Iterative and Incremental Development and Emergent Design
- know how to produce flexible, easily modifiable and maintainable code
- having experienced live and practical exercises for the application and integration of the skills acquired in the class

Course Organisation

The course with a total contact time of 16 hours is delivered in presence or in interactive online mode. The course is split in various modules, none of which exceeds two hours, with short breaks as needed and sufficiently long breaks between the sessions. The actual times for breaks are agreed upon in the group at the beginning of the course.

Pre-course and post-course activities are part of the training and are presented via our interactive online learning platform in various formats (video, text, quizzes, worksheets, further reading, ...).

Course Agenda

- Outside-in and inside-out design
- Test doubles – mocks, stubs, fakes
- Test case design guidelines for outside-in design
- Balancing inside-out and outside-in design
- Guiding outside-in design through microtests
- Review of software design principles
- Common code smells and remedies
- Establishing a test baseline: Golden Master approach
- Establishing a test baseline: Read by Refactoring approach
- Identifying seams in code
- Writing characterisation tests
- Breaking dependencies to enable microtesting
- Practicum: