

## Technical Practices: TDD-1



With this course, you are introduced to the fundamental concepts of agile technical practices, with particular attention to the techniques of eXtreme Programming. The course provides the necessary basis to allow the team to improve the quality of work in a sustainable manner in the long run by producing flexible, easily modifiable and maintainable code. In the course, theoretical presentations are alternated with practical exercises that apply and integrate the concepts presented.

Start learning Agile Technical Practices in this intensive, hands-on class. Guided by experienced practitioners, you will learn the basics of Test-Driven Development (TDD), Pair Programming, Iterative and Incremental Development, Emergent Design, applying robust software design principles, and working with frameworks and libraries.

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 the fundamental concepts of agile technical practices
- know the basics 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

- TDD as a software design technique & testing tool
- Where TDD fits in the overall picture of software testing
- History of TDD and when to use it
- Concepts of unit testing and microtesting
- Code isolation
- Detroit school or “classic” style TDD practicum
- Inside-out design
- Test coverage metrics – meaning and use
- Characteristics of effective test cases
- Organisation and naming considerations for test suites
- Test-driving code that interacts with frameworks and libraries
- IDE and VCS integration
- Value of collaborative work
- developing with artificial intelligence as a pairing partner
- Pair programming basics, anti-patterns, different styles
- Test suites as design documentation
- Up-front and emergent design and how to balance them
- Software design principles
- Behavior-focused vs. implementation-aware tests
- Guiding design through microtests – practicum
- Branching strategies