

CLEAN CRAFTER

A JOURNEY TO SOFTWARE CRAFTSMANSHIP

by Yoan THIRION



About Me

I have been crafting software for over 15 years.

My various experiences in fields such as banking, insurance and industrial environments have allowed me to understand what is fundamental to succeed in this area: agility and technical excellence.

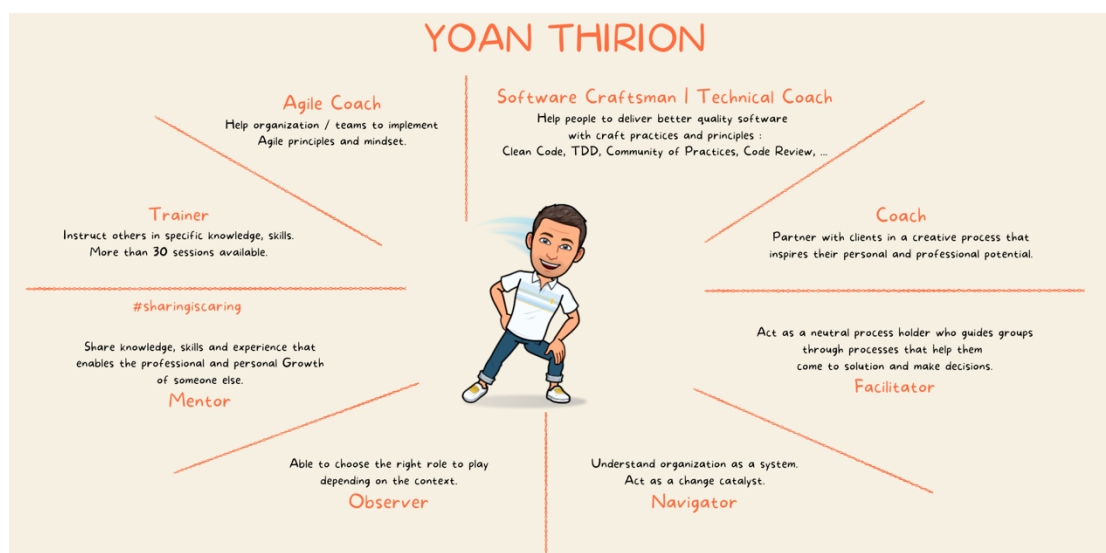
That's why, for the past 5 years, I've been focused on helping teams deliver well-designed software by coaching them in the implementation of agile practices.

I help teams to improve in the following areas: XP practices, integration and continuous development, Domain Driven Design, Clean Code, and many other technical practices (mob programming, code review, refactoring practices, ...)

My daily goal: *"Help individuals grow/acquire new technical, agile and human skills to deliver high quality software in the most efficient way possible.*

This involves making them autonomous in their practices and learning (developing the ability to adapt)."

More about it: https://www.yoan-thirion.com/#what_i_can_do



Yoan THIRION

LuxPayroll S.à r.l – 15 Rue de l'industrie, L-8069 Luxembourg

Numéro de TVA : LU25 058 449 – Numéro d'immatriculation : 2011 2442 953



Why This Training?

Because there is no agility without technical excellence and continuous learning. In this training attendees will discover what is software craftsmanship, the associated mindset, and the key techniques to be able to work in an iterative and incremental way. This training aims to help programmers become better at writing and maintaining code.

"Continuous attention to technical excellence and good design enhances agility."

Agile Manifesto – P9

Learning Objectives

- Understand what is meant by Software Craftsmanship and related practices (mainly from Extreme Programming)
- Understand the strong relationship between software craftsmanship and the agile world
- Be able to apply "Clean code" and "SOLID" principles daily
- Refactoring at heart: refactor, refactor, and refactor again
- Learn Test Driven Development and understand that it is not a Test approach but a Design approach
- Discover how to pair program and mob program efficiently, as well as the associated benefits

How Does it Work?

This training is built as a mix of theory and practice based on the "Training from the back of the room" / inversed pedagogical approach.

Attendees will experiment "learning by doing" through different activities:

- Code katas using Pair and Mob Programming
- Serious Games and collective intelligence activities

Here is a list of information about the training:

- Duration: 2 days
- Number of participants: from 6 to maximum 12
- Facility & material required:
 - White board(s) / Paper board(s)
 - Video projector
 - Attendees' computers (at least 1 for 2) with their favorite IDE
 - Internet access: With open access to github / miro / nuget.org

Yoan THIRION

LuxPayroll S.à r.l – 15 Rue de l'industrie, L-8069 Luxembourg

Numéro de TVA : LU25 058 449 – Numéro d'immatriculation : 2011 2442 953



Day 1: Clean Coder

- What is a software craftsman?
 - Software Craftsmanship manifesto
 - Associated practices: mostly Extreme Programming practices
 - Craft challenges game
- Clean Code and SOLID principles
 - Good code vs Bad Code
 - Crappy-Driven Development kata (using mob programming)
 - Refactoring using S.O.L.I.D principles
- Hands-on Test-Driven Development (TDD)
 - Chicago school
 - 3 rules of TDD
 - Roman Numeral Kata
 - Why TDD is not a Testing approach

Day 2: Testing at Heart

- Specification by Example
 - Agile Testing principles
 - Example Mapping
- Clean Tests principles
 - Unit Test Anti-patterns and best practices
 - Clean Tests surgery
- How to refactor Legacy Code (Trip Service kata)
 - Using Seams, Test Data Builders, Properties of Good Unit Tests
- Measure quality of our tests
 - Code coverage vs Code Quality
 - Hands-on Mutation Testing

Concepts Covered

- Software Craftsmanship and its manifesto
- Extreme Programming
- Clean Code
- S.O.L.I.D principles
- Specification by Example
- Example mapping
- Test Driven Development (Chicago and London school)
- Automated refactoring techniques
- Mikado method
- Unit Testing anti-patterns and best practices
- Mutation Testing
- Clean Architecture: Hexagonal / Onion Architecture
- Pair and Mob Programming

Optional Modules

According to your specific needs the discussions we would have before the session, you might decide to add some modules to this training:

- A Journey to Property-Based Testing: from example based-testing to Property-Based Testing
- Discover how you can ensure team architecture decision with ArchUnit
- How to refactor Legacy code - Theatre Kata
- Domain Driven Design re-Distilled: From anemic to Rich Domain Model
- Functional Programming Demystified in C# / java
- Programmer's Brain: understand how programmer brain's work and become better at reading complex code
- Approval Testing: Discover Use Cases for Approval Testing and how it can help on new codes and refactor legacy code
- Make Code Reviews Great Again
- Create a Learning culture (Lightning Talks, Community of Practices, ...)
- An introduction to sustainable IT

Attendees Background

- Code katas are in C# on .NET 6 or java (≥ 11)
- Developers should at least feel comfortable with a covered language
- Attendees can get assistance during exercises, but we will not have time to cover basics of installation, compilation, syntax, and such

Yoan THIRION

LuxPayroll S.à r.l – 15 Rue de l'industrie, L-8069 Luxembourg

Numéro de TVA : LU25 058 449 – Numéro d'immatriculation : 2011 2442 953



Contact Me

Yoan Thirion

Email: yoan.thirion@outlook.com

Phone: (+33) 6 46 42 36 71

Website: <https://yoan-thirion.com/>

Knowledge Base: <https://yoan-thirion.gitbook.io/knowledge-base/>



Yoan THIRION

LuxPayroll S.à r.l – 15 Rue de l'industrie, L-8069 Luxembourg

Numéro de TVA : LU25 058 449 – Numéro d'immatriculation : 2011 2442 953

