

Experiencing Domain-Driven Design

Three day workshop with Mathias Verraes



Software design can have a very **slow feedback loop**: by the time you find out if the architectural and design decisions you made were the right ones, it may be too late for drastic changes. As developers, we tend to phrase all problems as technical problems, which can be sufficiently challenging in their own right. We tend to **lose sight of the underlying business concerns**.

Domain-Driven Design is **total approach to building software for complex domains**. It's a way of thinking that helps you decide how to tackle complexity, and where to focus your efforts. By putting the business at the heart of our software, and keeping our domain model closely tied to our understanding of the business, we can create value at a much deeper level.

In this workshop, we'll explore this central premise of Domain-Driven Design. We'll take a practical approach: participants will be working on real-world problems. There will be **no easy answers**. Instead, you'll learn how to ask the right questions, and how to evaluate different models.

Contents

- Discovering the domain and the Ubiquitous Language
- Event Storming, conversations with Domain Experts
- Creating a model to share insights between both technical and non-technical shareholders
- Applying Systems Thinking and problem-solving techniques
- Visualizing strategic design using Bounded Contexts and Context Mapping
- Integrating with brownfield systems
- CQRS, Event Sourcing, temporal models
- Tactical Design Patterns
- Modelling exercises
- ...

By keeping the group small, everyone's questions can properly be addressed.

About the instructor

Mathias Verraes is a programmer, consultant, blogger, speaker, and trainer. He advises companies on building enterprise applications for complex business domains, and on dealing with legacy systems. He has advised for clients in Automotive, Logistics, Finance, E-Commerce, Web, and more. He's the organizer of the Domain-Driven Design Belgium community. When he's not working, he's at home in Kortrijk, Belgium, helping his two sons build crazy Lego train tracks.

<http://verraes.net>

Testimonials

"I spent three days immersed in the Domain-Driven Design world with Mathias. It was a great experience where I learned a lot about DDD. Mathias created an environment where we could interact all together experimenting with DDD. Furthermore Mathias adapted the workshop content dynamically to our needs and our concerns. It is a recommended experience!"

– Jordi Llonch, Technical Lead, Akamon Entertainment (Spain)

"Mathias made Domain-Driven Design very real and practical. He managed to find a good balance between practical workshop, theory, and sharing experiences. This workshop really gave me the keys to use DDD in our projects and showed me new perspectives like Event Sourcing. It is definitely a must have complement to the usual books by Eric Evans and Vaughn Vernon. I would definitely recommend this workshop to any developer or architect who is keen to learn a different way to model software."

– Sébastien Abras, Senior Software Architect, EVS Broadcast (Belgium)

"Event Storming is truly valuable, also as an exercise. I found the patterns to separate domains and their relations very interesting. (...) I've discovered that DDD has many patterns and strategies that we seem to have lost. It is essential knowledge for every developer, from junior to architect. For more experienced developers (and business architects), the Strategic Domain-Driven Design workshop gives theory and experience to better deal with (business) domains, relations, Bounded Contexts, and how to map systems. Must attend!" (translated from Dutch)

– Marc Van Andel, Integration Specialist / Java Architect, Kadaster (Netherlands)

Participant Requirements

- You are a Software Developer with some years of experience in working with complex domains. You're familiar with object-oriented programming.
- It's recommended to read "[Domain-Driven Design](#)" (Evans, 2004), but not essential.
- Laptops or other devices are not required, but useful for taking notes.
- The course is programming language agnostic.