

Here's a list of some other resources to get you further in your DDD journey:

Architectural Guidance

- [CQRS Journey](#) - nice experience report (.NET-centric) on applying CQRS and EventSourcing on a sample project
- [Why Use Event Sourcing?](#) - Greg Young's original blog post (Greg is prolific, so Google for his material as videos, etc).
- [Hexagonal Architecture](#) - Alistair Cockburn's original article (there's also some good info in the IDDD book by Vaughn Vernon)
- Microservice Patterns - Chris Richardson - Due for publication in late 2017.
- [The Dark Side of Event Sourcing: Managing Data Conversion](#) - by Overeem, Spoor, & Jansen.
- [Reactive Messaging Patterns with the Actor Model](#) - Vaughn Vernon.

Functional Approaches

- [An idiomatic F# implementation of Domain-Driven Design](#) - by Leo Gorodinski
- [Domain Driven Design with the F# Type System](#) - by Scott Wlaschin

Succeeding with Legacy Systems

- [Getting Started with DDD When Surrounded by Legacy Systems](#) by Eric Evans - this is great material!!! (white paper - pdf), associated talks are online as [Four Strategies for Dealing with Legacy Systems](#) videos
- [Refactoring to Deeper Insight - Lessons Learned Applying DDD to Large Scale Ruby and Rails](#) (talk - video) - my recent experience report on applying DDD techniques in a legacy system
- [Legacy Application Strangulation : Case Studies](#) - Paul Hammant

Talks

- [Design and Agile](#) - My talk from DDD Europe 2016 (there's a [nice writeup on InfoQ too](#)) on traps and tricks in making design work be the sweet, sweet center of your agile process. There are many other great talks from DDD Europe 2016 and 2017.
- [EventStorming](#) - My talk from YOW! Australia Melbourne, Dec 2016.
- [DDD Exchange](#) - Videos of talks are available from previous years

Sample Applications

- [Ruby Sample App](#) - my sample app contribution. Has a comprehensive list of resources on techniques and shows how DDD patterns might be applied in a dynamic language, using a document store, with a hexagonal-style approach
- [Updated Java Sample App](#) - still a work in progress. An updated version of the original [2009 Java version](#).
- [.NET Sample App](#) - a little outdated code-wise now, but still relevant for the techniques.
- [DDD Sample App in Go](#) - interesting port, since each application service is exposed as an API. It is accompanied by an Angular JS application to demonstrate the various use cases.