



Patterns Principles and Practices of Domain Driven Design

Scott Millett

[Download now](#)

[Read Online](#) ➔

Patterns Principles and Practices of Domain Driven Design

Scott Millett

Patterns Principles and Practices of Domain Driven Design Scott Millett

This book distils the ideas and theories of the Domain-Driven Design (DDD) philosophy into a practical play-book that you can leverage to simplify application development for complex problem domains. A focus is placed on the principles and practices of decomposing a complex problem space as well as the implementation patterns and best practices for shaping a maintainable solution space.

Patterns Principles and Practices of Domain Driven Design Details

Date : Published 2015 by Wrox (first published May 19th 2014)

ISBN :

Author : Scott Millett

Format : Paperback 736 pages

Genre : Computer Science, Programming, Science, Technology, Architecture, Software, Technical, Nonfiction



[Download Patterns Principles and Practices of Domain Driven Desi ...pdf](#)



[Read Online Patterns Principles and Practices of Domain Driven De ...pdf](#)

Download and Read Free Online Patterns Principles and Practices of Domain Driven Design Scott Millett

From Reader Review Patterns Principles and Practices of Domain Driven Design for online ebook

Michal Paszkiewicz says

<http://www.michalpaszkiewicz.co.uk/bl...>

Chandan Khatwani says

Amazing read

I feel great after reading this book on ddd. Why was this book not available before. I have read this book twice and enjoyed thoroughly.

Jan Van Ryswyck says

The very first book I've read on domain-driven design was Jimmy Nilsson's "Applying Domain-Driven Design and Patterns". I read that book in 2006, after which at the beginning of 2007 I read the blue book by Eric Evans for the first time. I had a lot of questions back in those days on the practicalities of the topics that Eric Evans explains in his book. The book that Jimmy Nilsson wrote was his view on domain-driven design and the things he learned from wrestling through the writings of Eric Evans. The next and last noteworthy effort in book form on this topic dated from 2012 when Vaughn Vernon released his take titled "Implementing Domain-Driven Design". This book, which I still have to get through, takes another practical stance on domain-driven design just as Jimmy Nilsson did for his book.

After I learned that Scott Millett had written a new book on domain-driven design, he was so kind to sent me a copy. I was very eager to pick it up and read it. And I'm very glad I did. This book can be regarded as the current state of affairs when it comes to domain-driven design. Scott took the right approach by first discussing the strategic patterns before the tactical patterns. What I personally like is how Scott kindly builds up knowledge with the reader, bringing back and repeating topics from previous chapters and also pointing to next chapters, pointing out how things relate and influence each other. The book read as a story, slowly building up knowledge.

As already mentioned, the first part of the book discusses the strategic patterns of domain-driven design. The chapters that explain the tactical patterns are richly illustrated with lots of code. What I particularly like is how Scott informs the reader by showing different implementations and clearly discussing their advantages and disadvantages. This unopinionated approach is quite refreshing. I particularly liked the chapter about domain events, where several approaches are touched and explained. However, I did not care a lot about the chapter on repositories where I found it to have too much code listings that were too long for my taste. The last part of the book also touches on topics like CQRS and event sourcing.

I'm a slow reader, so it took me quite a while to get through it. But it was definitely worth turning over all 736 pages. I learned a lot of new things and also relearned some known things. This is a perfect book for developers who are new to domain-driven design and want to learn what it's all about. But also developers

who are more experienced when it comes to domain-driven design can benefit a great deal by reading this book. Highly recommended!

Moqbil says

I read this book for only event sourcing chapters, and I end up reading the entire book, Best book ever about domain driven design and specially event sourcing.

Robson Castilho says

PPPoFDDD is similar to "Implementing Domain-Driven Design" book by Vaughn Vernon, although it presents the examples with .Net technologies. Both books give focus on bounded contexts and strategic design and they're more technical than Evans' DDD book, showing a lot of implementation details regarding bounded contexts integration with REST and messaging systems.

This book may be considered a "software book" or "software architecture book" because of its wide range of technical topics. The author wastes too much pages with small tutorials teaching how to implement some concept with different strategies and technologies. If you're only worried about DDD main topics, you may think this book is too much big and somewhat boring.

Rezaul Karim Sajib says

I first got myself introduced with the ddd concept by reading Martin Fowler's blog. I then read the book of Eric Evans, read lot of articles in the web regarding the implementation of CQRS, DDD. In my opinion, Eric Evans book may be the bible of the concepts of DDD but anyone who wants implement DDD in real life should read this book. This book describes the concepts so clearly i finished the large book within two weeks. This book also helps me understand the concept of event store, what things we need to consider implementing my own event store, temporal queries which I didn't found completely in any other source. Apart from that, it compares several of the design patterns with pros and cons in implementing DDD are well described with examples. I highly recommend any person who want to move on to the traditional application development to DDD, CQRS to read this book.

Daniel says

It was ok. The content was highly repetitive. In the plus side, it makes the reading go by faster as you can skip entire paragraphs later in the book, but it does end up making the book very thick. The one upshot to this layout, though, is that it's probably better as a reference to various topics since they are bit more self-contained.

Cheng Xu says

Covers many useful topics on DDD. Overall a bit dry and repetitive when explaining some concepts

Ahmad hosseini says

What is DDD?

Domain Driven Design or DDD is an approach to software development that enables teams to effectively manage the constructions and maintenance of software for **complex problem domains**.

DDD has two kind of patterns:

- 1- Strategic patters that shapes the solution and it can be useful for any application.
- 2- Tactical pattern that are used to implement a rich domain model. They are only useful if your model is sufficiently rich in domain logic.

About the book

This is a perfect book for developers who are new to domain-driven design and want to learn what it's all about. But also developers who are more experienced when it comes to domain-driven design can benefit a great deal by reading this book. Not only does it have plenty of theory, it has plenty of practical examples in both diagrams and code.

A big plus is that the author reflected on many of the modern tendencies in application architecture such as CQRS, Event Sourcing, REST, Microservices, Messaging system etc.

Cons

There are plenty of references to .NET, it may not be a Pro for everyone.

Rory Armstrong says

Serves as an introduction to Architecting systems, Microservices, Domain Driven Design, and requirements gathering with customers.

Explores the strategic and tactical patterns that domain driven design covers such as Rest/RPC/SOAP, value objects, CQRS etc.

A great book if you're moving from an intermediate role to senior / architecting roles.

My only criticisms would be the sheer length, fairly dry writing, and it's choice of C# for coding samples (although that's just a personal nitpick).

Steven Ramirez says

It's done. The nightmare is over. This is the driest book I have read in a while. There is a ton of useful information but you have to get through a lot of pain to get it. I found it immensely useful for architecture design but it is not for the faint of heart.

Damian Zydek says

I recommend this to anyone who is looking for practical knowledge of applying Domain-Driven Design in .NET applications.

Sebastian Gebski says

Domain Modeling is hard. Teaching & presenting about Domain Modeling is even harder (I've tried it, honest word). Writing a *good* book about Domain Modeling is ninja-level challenge. And fortunately PPPofDDD is a *good* book about Domain Modeling. Maybe it's more 4.5 stars than 5, but I still believe it's more 5 than 4. Why?

- 1.) Author does not try to re-tell The Blue Book. I really, really had the honest feeling that his greatest intention was to show how do you use DDD principles while building actual systems. Not just on line-of-code level. And what's important, I think he did it well, at least clearly better than VV in Red Book.
- 2.) It's quite easy to write about desing in a boring way: either because you being too abstract (only conceptual notions) or to specific (by diving deep into the examples you readers do not give an f-word). But to be honest this book is not boring -> maybe it's author's writing style. Anyway, this book is in general more pleasant to read (& more engaging) than both Blue and Red Book.
- 3.) There are plenty of references (incl. all examples) to .NET! ;) Ok, it may not be a Pro for everyone, but I appreciate it.
- 4.) Part IV (which I initially expected a total shit & 'space-filler') appeared to be the best part of the book - very practical & pragmatic. Even if I don't necessarily agree with all the statements (or rather - with some I agree with 'yes, but ...' suffix :>), I really like all the thoughts gathered in such a clear & approachable way.
- 5.) Any specific points made I've found useful? Yes, some about composability (bounded context & dependencies between them), state pattern & lazy loading.

Cons? Yes, some:

- 1.) They were at least few examples that didn't really add much value - either too obvious or too bloated (what makes me feel they miss the point or don't have any).
- 2.) Chapters 9 & 10 are surprisingly shallow. I've kinda expected more. Fortunately book compensates it later.

To summarize: it's a very good book about DDD. It doesn't change the fact that you're supposed to read Blue Book first anyway. But if you're about to choose whether read this one or Red Book ("Implementing DDD"), I believe that PPPofDDD is a better choice.

Sandeep says

Rating 3.5/5

As a technical book, I was expecting it to be more precise and hitting the target. But here and there it turned out to be too much of story telling. (at-least in the first part of the book).

It may act as an advantage to read this and follow all of the steps if and only if

- the stakeholder gives sufficient money to invest time into this.
- if the project appears to be from scratch and you need all of the best practices to be a part of the recipe.

I really liked Part 3 and 4, those were what impressed me, we often don't get enough time and money to involve in a project from scratch (esp considering I am from India).

I read this book only looking at specific chapters, so there are chances this review is biased!

Jaskula says

The most up to date book about Domain Driven Design patterns I read so far. Every pattern is discussed from different angles and based on business scenarios. It covers different architectural patterns going from classic layered architecture to CQRS and Event Sourced systems. Many interesting patterns are discussed on how integrate different bounded context, etc. Really a god read.
