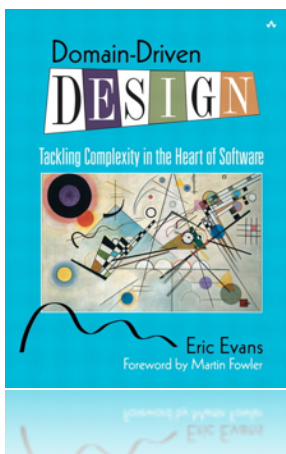


References for DDD

Core Domain



Domain-Driven Design Tackling Complexity at the heart of software - Eric Evans

Addison Wesley

This is where everything started, and it is still the most authoritative source of information about DDD. However many things have happened since 2004, so it is better to keep an eye open for some fresher content.

How to read the book: A good recommendation is to read the introduction and skip the initial part. Go straight to the strategic domain-driven design section, and the start reading the initial part.

Domain-Driven Design community site

Have a look to <http://domaindrivendesign.org> the official website of the DDD community. This is where official informations are published.

The video section contains some interesting content too.

<http://dddcommunity.org/library/>

Domain-Driven Design mailing list

The official mailing list for Domain-Driven Design is hosted on yahoo groups.

<http://tech.groups.yahoo.com/group/domaindrivendesign/>

If you speak Italian, I might also suggest the Italian mailing list about DDD.

<http://it.groups.yahoo.com/group/DDD-IT/>

Eric Evans interviews and talks on InfoQ

Interview on Domain-Driven Design

<http://www.infoq.com/interviews/domain-driven-design-eric-evans>

Putting the model to work <http://www.infoq.com/presentations/model-to-work-evans>

Why DDD matters today <http://www.infoq.com/articles/eric-evans-ddd-matters-today>

Strategic Domain-Driven Design <http://www.infoq.com/presentations/strategic-design-evans>

DDD and Domain Specific languages <http://www.infoq.com/presentations/ddd-dsl-evans>

DDD sample Application

A working implementation of DDD principles in SpringMVC plus Hibernate, maintained by Swedish company Citerus.

<http://dddsample.sourceforge.net/>

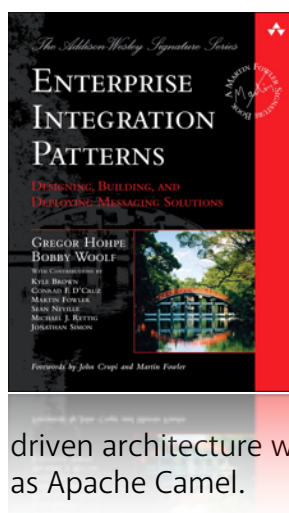
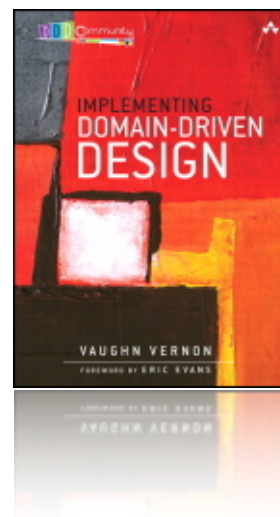
There is a common effort to bring more updated and different implementations available, so ...stay tuned.

Implementing Domain-Driven Design - Vaughn Vernon

Addison Wesley

This book touches most of the hot topics in DDD providing practical advice about how to overcome modeling and practical implementation issues, focusing on implementations on the Java platform, and on state-of-the-art technologies, such as NoSQL, messaging systems and so on. It is probably the closest to a second edition of the blue book. A must read for everybody approaching DDD now.

More info also on Vaughn's [blog](#).



Enterprise Integration Patterns - Gregor Hohpe, Bobby Woolf

Addison Wesley

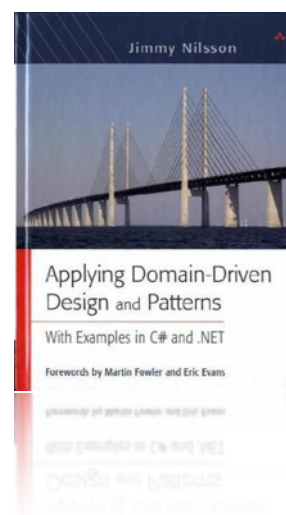
This one didn't look like a book about domain-driven design at first. Just like a damn-good book about message-driven architectures and SOA. However, it is becoming clearer every day that solid foundations are needed to deal with the issues of message oriented architectures. This book demystifies the topic and makes everything look incredibly easy.

Among other things, the book defines a pattern language for message driven architecture which later became part of implementation of messaging middleware such as Apache Camel.

Applying Domain-Driven Design and Patterns - Jimmy Nilsson

Addison Wesley

A more implementation related approach, that focuses on the mechanics of the implementation of tactical DDD with C# and .Net. One of the reasons why DDD took more ground in C# than on the Java platform.





Domain-Driven Design Quickly - Abel Avram & Floyd Marinescu

InfoQ

This is one of the most known entry point for domain-driven design, since the PDF is downloadable for free. However, this is also one of the most dangerous entry-points in the DDD world since it focuses only on the first part of the blue book.

<http://www.infoq.com/minibooks/domain-driven-design-quickly>

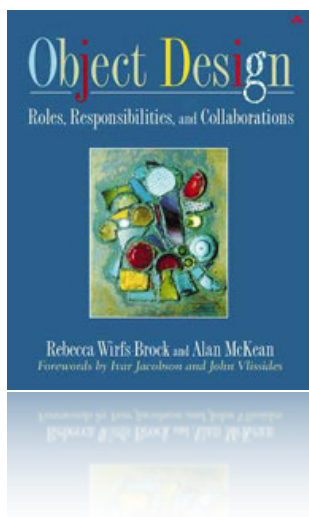
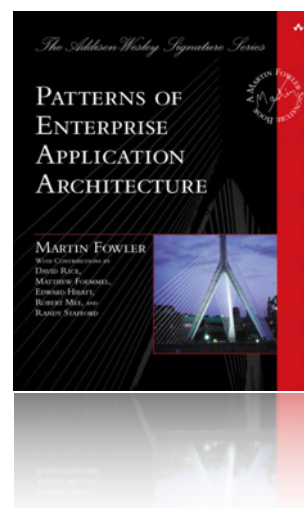
Common Ground

A collection of books that are supposed to be shared foundations among DDD practitioners.

Patterns of Enterprise Application Architecture - Martin Fowler

Addison Wesley

This is probably the “common ground” book. When doing DDD we invariably end up talking about architectural patterns. Martin Makes a fantastic job in explaining the most common architectural patterns and the motivations behind them. If you’re looking for a simple and clear book that explains what a *transaction script* is or why a *domain model* might be a better choice than an *active record* based solution, this is the book for you.



Object Design. Roles, Responsibilities and Collaboration - Rebecca Wirfs-Brock

Addison Wesley

Definitely one of the best books available about Object-Oriented Design.

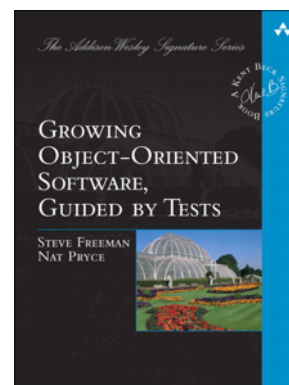
A must read for every practitioner looking for a deep understanding of Object-Oriented Design.

How to design a complex system around meaningful foundations, thinking in terms of roles, responsibilities and collaboration.

Growing Object Oriented Software Guided by Tests – Steve Freeman & Nat Pryce

Addison Wesley

A fresher approach to Test Driven Development. Showing the techniques, the architecture and the project. Unsurprisingly, the Blue Book is one of the most frequently referenced books, since the Domain Model pattern provides better testability, and the tactical patterns perform at their best under heavy testing.



Lean Software Development, An Agile Toolkit – Mary Poppendieck, Tom Poppendieck

Addison Wesley

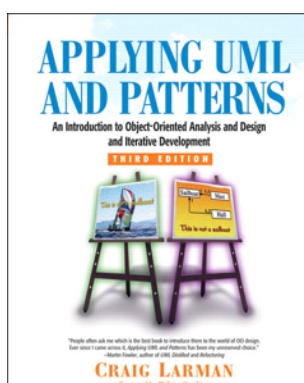
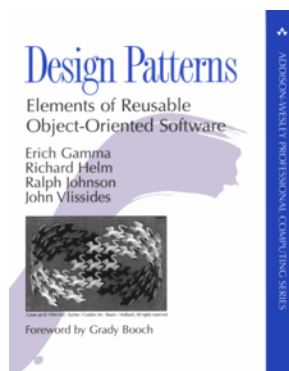
Too often lean is associated only with the over-simplified vision of eliminate waste, while in fact there's a lot more. Deep understanding of your product and of the problem you're trying to solve deeply resonates with the basic concepts of Domain-Driven Design. Moreover, without a clear, business oriented vision it's definitely hard to focus only on the sensible part of the application. Lean and DDD seem distant, but they're

not and the blue book is again frequently cited here.

Design Patterns. Elements of Reusable Object-Oriented Software – Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides

Addison Wesley

When the Blue Book was published, it looked like the relationship with the GoF bible was stricter than it actually was. However, this is definitely expected common ground for every DDD practitioner.



Applying UML and Patterns: An Introduction To Object-Oriented Analysis And Design And Iterative Development – Craig Larman

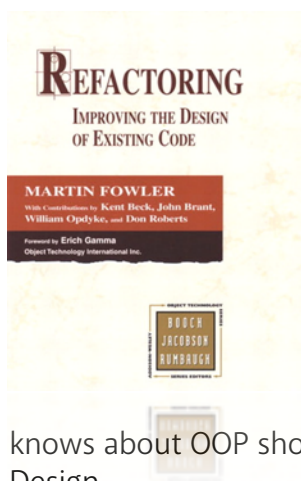
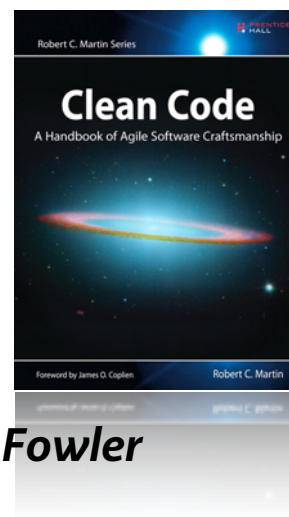
Prentice Hall

This is one of the clearest book ever about software development. Don't be fooled by the size, the internal map structure clarifies the context for every chapter, making it really easy to understand why you're here and which is the problem we're talking about. A great book on basic OOP principles: this is where Craig Larman explains the GRASP patterns, a foundation for every Object Oriented Programmer.

Clean Code - Robert C. Martin

Prentice Hall

Despite being somewhat dogmatic on given topics, the book is a must read. Robert C. Martin explores the dynamics of software writing and evolution showing why and how developers should put extra care in writing software. If you're writing software for the core domain, SOLID principles should definitely be your standard.



Refactoring, Improving the Design of Existing Code - Martin Fowler

Addison Wesley

Refactoring is crucial in DDD. We'll be learning, so we'll have to rewrite our code often along the way.

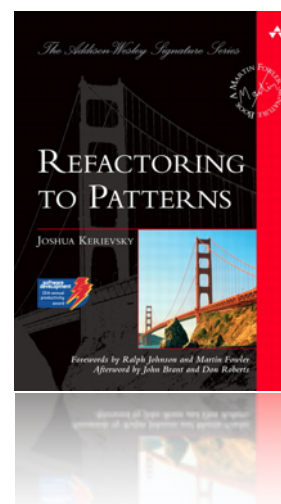
The second part of the book might seem outdated, since many of the mechanics that were highlighted Martin Fowler are now part of the standard features of professional IDEs (thanks again, Martin), but the first 80 pages are still a smasher. Every developer thinking that he/she

knows about OOP should read them and realize how little we know about Object Oriented Design.

Refactoring to Patterns - Joshua Kerievsky

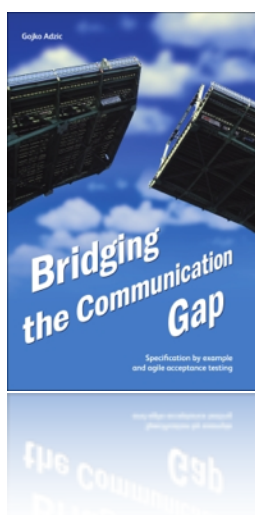
Addison Wesley

Refactoring might be even quicker and more effective if you know where to go. In this book Joshua explains mechanics and motivations for evolutionary design showing how to evolve safely from one implementation style to another. From chaos to elegance, and from one elegant implementation to a more suitable one according to the existing forces.



Domain Discovery

Coding it right is only half of the job. Doing the right thing is probably the hardest thing to do.



Bridging the Communication Gap - Gojko Adzic

Neuri

In this book, Gojko explores the grey area of requirements gathering, interaction with stakeholder and systematic removal of ambiguity. He shows the power of concrete examples and acceptance testing as a way to overcome implicit ambiguities and fallacies in one of the hotspots of effective software development.

By the way, Gojko used to be the official DDD trainer in UK for a while, while writing this book.

Specification by Example, How Successful team deliver the right software

Manning

Gojko goes further exploring the mechanics of acceptance testing at the technical, conversational and process level. Shows examples and stories from several real world projects, of implementation of Agile Acceptance Testing, Behavior Driven Development and Specification by Example. Different names, not so different concepts. Gojko also gave this great presentation some time ago.

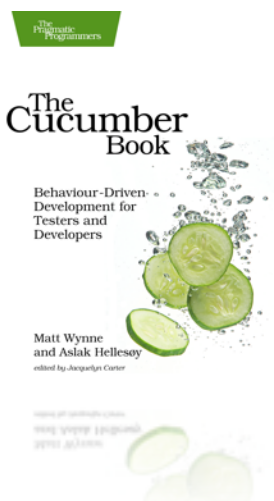


The three amigos: DDD, TDD & BDD

<http://skillsmatter.com/podcast/design-architecture/ddd-tdd-bdd>

Gojko Adzic's Blog

<http://gojko.net/>



The Cucumber Book - Matt Wynne, Aslak Hellesøy

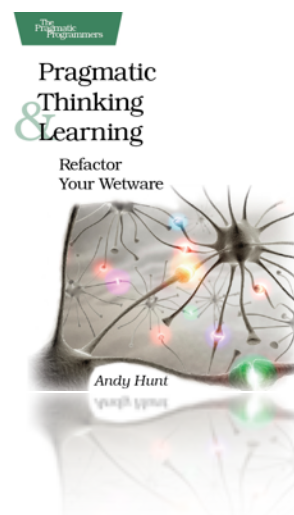
The Pragmatic Bookshelf

Probably the best book on Cucumber and Behavior Driven Development in general, filled with precious tips at the process and modeling level. Crafting a Ubiquitous Language in testing might be tricky before reading this one, and incredibly natural right after.

Pragmatic Thinking and Learning, Refactor your wetware - Andy Hunt

The Pragmatic Bookshelf

Written from a developer for software developers, this books explores and explains the way our brain works focusing particularly on learning and creative problem solving. If you're interested in understanding why the best ideas come under the shower, and how to be more effective in learning without wasting too much water. That's the book for you.



"Everybody knows the problem, we need to be more innovative. Now we've got the solution. Gamestorming. This smart, fun, hands-on book will energize your brain and revolutionize your creativity—and do it using stuff you already have in your office supply closet!"
—Nathan Popkin, author of *Reclaim the Work Week*

Game storming

A Playbook for Innovators, Rulebreakers, and Changemakers



Game Storming - Dave Grey, Sunni Brown, James Macanufo

O'Reilly

Brain storming and group activities could be an incredible source of productivity ...or an incredible source of waste. If you're interested in ways to increase effectiveness of your brainstorming and problem solving sessions, here'll find a lot of tips to fine tune the dynamics of a creative meetings.

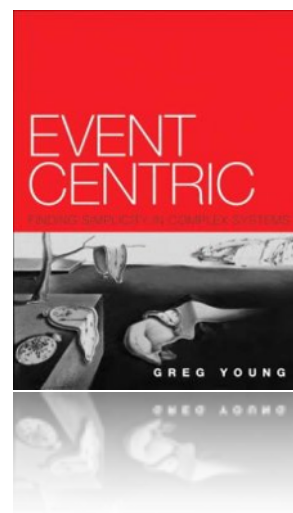
CQRS-ES

The most interesting place to start from is probably the official mailing list:
dddcqrs.googlegroups.com

Event Centric: finding simplicity in complex systems - Greg Young

Addison Wesley

This is the book that should reveal everything to the world. Unfortunately, it hasn't been published yet. Some of the content is anyway already available on Greg's Blog: <http://codebetter.com/gregyoung/> and on EventStore web site documentation. <http://geteventstore.com/docs/>



His presentations definitely deserve a look too:

Innovation and Event Sourcing

<http://skillsmatter.com/podcast/design-architecture/architectural-innovation-eventing-event-sourcing>

People to follow

Udi Dahan

Pioneered CQRS and Event sourcing, some presentations definitely ignite a different way of thinking.

Command Query Responsibility Segregation

<http://www.infoq.com/presentations/Command-Query-Responsibility-Segregation>

Avoid a failed SOA - Presentation by Udi Dahan

<http://www.infoq.com/presentations/SOA-Business-Autonomous-Components>

Rinat Abdullin

Rinat has been pushing forward an amazing implementation of CQRS and Event Sourcing with his company Lokad. He also put the underlying framework code open source. Amazing guy. His blog can be found at <http://abdullin.com/>

Martin Fowler

Martin is kind of an obvious call... But I'd suggest this article in particular. <http://martinfowler.com/bliki/AggregateOrientedDatabase.html>

Tools

Greg Young's Event Store

<http://geteventstore.com/>

Jamie Oliver's Event Store

<https://github.com/joliver/EventStore>

LOKAD.CQRS

Sample code and a basic framework for CQRS and Event Sourcing
<http://lokad.github.com/lokad-cqrs/>

NServiceBus

A C# Service Bus with an eye on SOA and another one on DDD, from Udi Dahan
<http://nservicebus.com/>

AXON Framework

The most popular CQRS Framework in Java: <http://www.axonframework.org>

AKKA

A Scala framework based on the actor model. <http://akka.io/>

Qi4j

A brilliant framework for DDD via DCI in Java.
<http://qi4j.org/>

Disclaimer: all links and descriptions reflect my personal opinion at the time of writing. I am a learning human being so my opinions may change over time (quite often, and in a fuzzy way). I don't think this list is complete nor exhaustive, just ...good enough. I might have forgotten something or someone. If you have any comment, suggestion or mistake to report, feel free to contact me at alberto.brandolini@avanscoperta.it