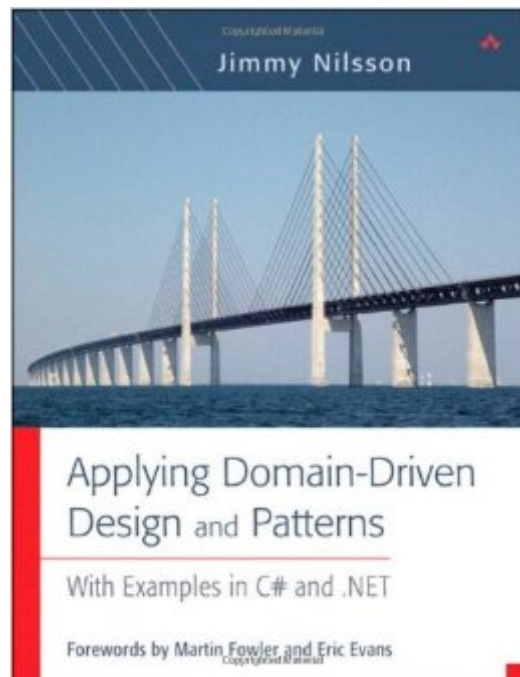


The book was found

Applying Domain-Driven Design And Patterns: With Examples In C# And .NET



Synopsis

Patterns, Domain-Driven Design (DDD), and Test-Driven Development (TDD) enable architects and developers to create systems that are robust and maintainable. While the examples in this guide are in C# and .NET, the principles can be used by developers using any language and IDE.

Book Information

Hardcover: 576 pages

Publisher: Addison-Wesley Professional; 1 edition (May 18, 2006)

Language: English

ISBN-10: 0321268202

ISBN-13: 978-0321268204

Product Dimensions: 7.3 x 1.4 x 9.5 inches

Shipping Weight: 2.1 pounds (View shipping rates and policies)

Average Customer Review: 3.8 out of 5 starsÂ Â See all reviewsÂ (25 customer reviews)

Best Sellers Rank: #655,201 in Books (See Top 100 in Books) #211 inÂ Books > Computers & Technology > Programming > Microsoft Programming > .NET #274 inÂ Books > Computers & Technology > Networking & Cloud Computing > Data in the Enterprise > Client-Server Systems #283 inÂ Books > Computers & Technology > Programming > Languages & Tools > C#

Customer Reviews

I was surprised that this book slipped under my radar for almost 3 months. I've been on the lookout for just such a unifying tome of knowledge that relates patterns and domain-driven design (DDD) to a practical .NET example for quite some while. The book delivers well on its promises, significantly surpassing the only other real competitor, Foundations of Object-Oriented Programming Using .NET 2.0 Patterns. The pros and cons, as I see them, are outlined below:

PROS* Combines the ideas of Domain Driven Design (Evans) with Patterns of Enterprise Application Architecture (Fowler). These books are pretty much mandatory reading prior to diving into this book.* Draws upon a myriad of other well-known sources, including materials from Refactoring to Patterns and the GoF, work from Johnson and Lowy, as well as a rare reference to Naked Objects. The more experienced and better read you are, the more this stuff will make sense.* Rare .NET coverage of advanced concepts like Plain Old CLR Objects (POCOs), persistence ignorant (PI) objects, O/R mapping with NHibernate, Dependency Injection, Inversion of Control, and Aspect-Oriented Programming.

CONS* While some sections are really insightful and could contain more interesting materials, other sections seem to drone on too long. The work on defining the NUnit tests, in particular, flows like a stream of

consciousness and doesn't really add a lot of structured value to understanding DDD, patterns, or TDD for that matter.* Embedded comments in the text adopt from the style used in Framework Design Guidelines. It worked very well for Cwalina / Abrams in their book because it seemed planned in from the outset. Comments like "one reviewer commented on the code with the following, more succinct version" seem like editorial comments left in and not collaborative authoring by design. All-in-all a very solid book that fills a unique market niche, leaving it pretty much without peers. If it had a 4.5 stars rating, Applying DDD would get it. As a secondary reference book, it doesn't offer the earth shattering insights of some of the innovative source materials found in the Fowler Signature Series, for example. It does, however, weave together an interesting example of how to tie all of these concepts together for the .NET architect looking to take their understanding to the next level.

I was expecting this book for a looong time. For those who are new to DDD and want to be a good practitioner, I think this is a must read. The reason is not only that there are not many books in this topic (check out Eric Evan's DDD book if you haven't done so) but also there are not many "down-to-earth" books available. In this book you will find many "real world" examples where the author discusses the pros and cons. I like the books that discuss the trade-offs instead of the ones that try to give "universal" answers; as "it depends" is usually the answer to most of the questions in software development. Apart from DDD, if you are also new to TDD, PEAA (Patterns of Enterprise Application Architecture by Martin Fowler, another great book), O/RM (NHibernate to be specific), Mocking frameworks (NMock to be specific), SOA, AOP etc you will find introductory level information in the book which is just enough to get started. With this book the link between the PEAA and DDD is clearer than ever. It does a great job on how to use PEAA and DDD in a complementary way. I should also mention the format of the book; it is easy to read and grasp. No need to mention that the idea of having guest authors for specific topics is just great. And also as readers we might be subject to a new trend; having "product placements" in the book :) Some Swedish brands made it to the book as the author being a Swedish guy, which I think totally fair :) I want to thank Jimmy and all the coauthors for this great work.

The reviewer saying it sounds like a tired colleague's random ramblings is spot on. It's very hard to read, and not because the matter is difficult, but because it's terribly written. Fowler's and Evans' books are masterpieces. Very clean, focused, inspiring and readable. This one is nothing like that. A bit of TDD, a bit of NHibernate, but little concrete core. What's worse, very often I find myself in

complete disagreement with how the author is using various techniques. It hurt to read some of the TDD pieces, and it seems the author is yet to discover SOLID. Eventually the domain model looks like a complex, tangled ball of mud, not a consistent, elegant piece of art I expected.

At the start the author says that this book is for "a wide target audience" and that if you don't have some knowledge of "object-orientation and C#" interest and enthusiasm will compensate for any lack of prior knowledge. I've been a web dev for over 10 years and know more than something about object-orientation etc. but this book is NOT for a beginner or even an intermediate programmer. On page 4 he starts discussion something called "case focus" with no definition and moves right into Domain-Driven Design Focus, again without explaining what this means. I found the book quite full of jargon and buzz words with a large presumption that one has already had experience in these topics. He says he is trying to build a bridge between users and developers. Most users will glaze over after the first chapter without any clue as to what he is talking about. If you are an advanced object-oriented programmer familiar with UML and other design technologies then this might be the book for you. Sorry. But a book this complex is not a bridge between users and developers.

[Download to continue reading...](#)

Applying Domain-Driven Design and Patterns: With Examples in C# and .NET Domain Names For Profit: How to Play The Domain Name Game & Make Money Domain-Driven Design: Tackling Complexity in the Heart of Software MCAD/MCSD Self-Paced Training Kit: Developing Windows®-Based Applications with Microsoft® Visual Basic® .NET and Microsoft Visual C#® .NET, Second Ed: ... C#(r) .Net, Second Ed (Pro-Certification) Parallel Programming with Microsoft® .NET: Design Patterns for Decomposition and Coordination on Multicore Architectures (Patterns & Practices) N-Layered Domain-Oriented Architecture Guide with .NET 4.0 Corporate Taxation: Examples And Explanations (Examples & Explanations) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process (2nd Edition) What Customers Want: Using Outcome-Driven Innovation to Create Breakthrough Products and Services: Using Outcome-Driven Innovation to Create Breakthrough Products and Services The Domain Theory: Patterns for Knowledge and Software Reuse The Ultimate Question 2.0 (Revised and Expanded Edition): How Net Promoter Companies Thrive in a Customer-Driven World The Ultimate Question 2.0 (Revised and Expanded Edition): How Net Promoter Companies Thrive in a Customer-Driven World (Your Coach in a Box)

Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries Safari Animal Patterns: 30 Exotic Safari Animal Patterns to Feel the Wildlife World (Safari Animal Patterns, animal designs, zendoodle) Database Programming with Visual Basic .NET and ADO.NET: Tips, Tutorials, and Code The .NET and COM Interoperability Handbook (Integrated .Net) Building Web Solutions with ASP.Net and ADO.NET (Developer Reference) MCPD Self-Paced Training Kit (Exams 70-536, 70-528, 70-547): Microsoft® .NET Framework Web Developer Core Requirements: Microsoft .Net Framework Web ... Requirements (Microsoft Press Training Kit)

[Dmca](#)