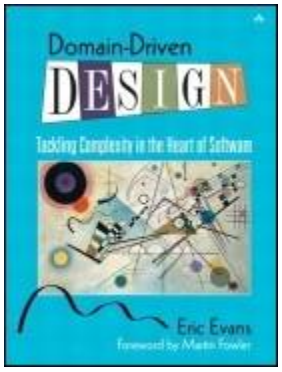


• Table of Contents

Domain-Driven Design: Tackling Complexity in the Heart of Software

By Eric Evans

Start Reading

Publisher: Addison Wesley
Pub Date: August 20, 2003
ISBN: 0-321-12521-5
Pages: 560

The software development community widely acknowledges that domain modeling is central to software design. Through domain modeling, software developers are able to express rich functionality and translate that functionality into software implementation that truly serves the needs of its users. Despite its obvious importance, however, there are few practical resources that show how to incorporate effective domain modeling into the software development process.

Domain-Driven Design fills that need. It offers readers a systematic approach to domain-driven design, presenting an extensive set of design best practices, experience-based techniques, and fundamental principles that facilitate the development of software projects facing complex domains. Intertwining design and development practice, Domain-Driven Design incorporates numerous examples in Java-case studies taken from actual projects that illustrate the application of domain-driven design to real-world software development.

Readers will find an overview of domain-driven design that highlights key principles, terms, and implications. The book presents a core of best practices and standard patterns that provide a common language for the development team. In addition, it highlights how refactoring in domain modeling, integrated with the frequent iterations of Agile development, leads to deeper insight into domains and enhanced communication between domain expert and programmer. Building on this foundation, the book then addresses domain-driven design for complex systems and larger organizations.

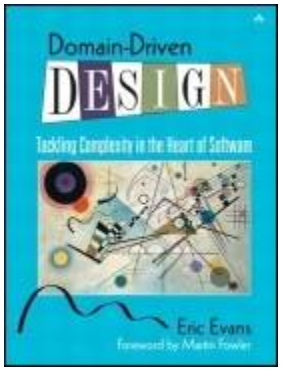
Specific topics covered include:

- Isolating the domain
- Entities, value objects, services, and modules
- The lifecycle of a domain object
-
-

- Representing processes as domain objects
- Creating functions free of side effects
- Conceptual contours
- Standalone classes
- Extending specifications
- Applying analysis patterns
- Relating design patterns to the model
- Maintaining model integrity
- Formulating the domain vision statement
- Choosing refactoring targets
- Responsibility layers
- Creating a pluggable component framework
- Bringing together large-scale structures and bounded contexts

With this book in hand, object-oriented developers, system analysts, and designers will have the guidance they need to think deeply about domains, create rich and accurate domain models, and transform these models into high-quality, long-lasting software implementations.

NEXT



- Table of Contents

Domain-Driven Design: Tackling Complexity in the Heart of Software

By Eric Evans

Start Reading

Publisher: Addison Wesley
Pub Date: August 20, 2003
ISBN: 0-321-12521-5
Pages: 560