Introduction to Domain-Driven Design (DDD) (TT1190)

Master DDD to improve collaboration, increase productivity and produce higher quality software

Course Snapshot

- **Course: Introduction to Domain Driven Design (TT1190)**
- **Duration**: 3 days
- **Audience & Skill-Level**: This introductory-level course is designed for anyone working in IT who wishes to learn Domain Driven Design.
- **Hands-on Learning:** This course is a lecture-based, workshop style event that includes pen and paper style labs and group activities. This is not a coding class although having knowledge of software development and basic programming skills is useful and recommended.
- **Customizable**: This course agenda, topics and labs can be further adjusted to target your specific training skills objectives, tools and learning goals. Please ask for details.

Overview

In today's highly competitive and rapidly changing software industry, the ability to design and develop software that aligns with the business needs is essential. By adopting a domain-driven approach to software development, organizations can achieve a better alignment between the software and the business needs, leading to improved collaboration, increased productivity, and better quality of software. Additionally, DDD can help developers reduce complexity, improve maintainability, and increase flexibility in their codebase.

Introduction to Domain-Driven Design is a three-day, workshop style course designed to provide students with a comprehensive understanding of DDD and its application in real-world projects, offering an engaging learning experience for those interested in software development and design. Throughout the course, you'll explore the key concepts and principles of DDD, including building blocks, designing bounded contexts, strategic design, tactical design, and learn how to apply DDD in real-world projects. You'll learn how to identify a domain model, design entities and aggregates, implement domain services and repositories, and create context maps for different subdomains.

Learning Objectives

Working in an interactive learning environment, led by our expert facilitator, you will learn how to:

- Identify a domain model and design entities, value objects, and aggregates for a given domain using domain-driven design (DDD) principles.
- Design bounded contexts, create context maps, and apply strategic and tactical design patterns in DDD to implement software systems that align with the business needs.
- Apply DDD in real-world projects, including identifying and overcoming challenges, and implementing best practices.
- implement domain services and repositories, and apply domain events and event sourcing, and CQRS using DDD principles and tactics.
- Analyze business requirements and identify core concepts and relationships within a domain to develop software that aligns with the business needs.
- Develop collaboration and communication skills necessary for working in a team environment in software development projects.
- Apply DDD to reduce complexity, improve maintainability, and increase flexibility in software systems.
- Identify anti-patterns and pitfalls in DDD and avoid common mistakes when applying DDD principles in practice.

Need different skills or topics? If your team requires different topics or tools, additional skills or custom approach, this course may be further adjusted to accommodate.

Audience & Pre-Requisites

The intended audience is software developers, software engineers, technical architects, and anyone involved in software development and design. This course is also suitable for project managers and business analysts who work closely with development teams.

The roles that would benefit from this course include:

- Software developers who want to learn how to apply DDD principles to their projects and write software that aligns with the business needs.
- Software engineers who want to deepen their understanding of DDD and learn how to design and develop software that is maintainable, flexible, and scalable.
- Technical architects who want to incorporate DDD into their design strategies and lead the development of software systems that align with the business needs.
- Project managers who want to gain a better understanding of the software development process and how to collaborate more effectively with development teams.
- Business analysts who want to learn how to identify and map out the core concepts and relationships within a business domain and translate them into software requirements.

Overall the course is suitable for anyone interested in learning how to apply DDD principles to software development and design. Whether you are a seasoned developer or just starting your career, this course offers a unique and valuable learning experience that will help you advance your skills and career in software development.

Pre-Requisites

In order to benefit from the course you should have a basic understanding of software development concepts, including programming languages, software architecture, and design patterns. It is recommended that students have at least 1-2 years of experience in software development or a related field.

Ideally, students should have the following knowledge and skills:

- Proficiency in at least one programming language, such as Java, C#, or Python.
- Familiarity with software architecture concepts, such as MVC, microservices, or event-driven architecture.
- Basic understanding of software design patterns, such as Factory, Singleton, and Observer.
- Knowledge of database concepts, such as SQL and NoSQL databases.

Enhanced Learning Services: Please also ask about our Pre-Training Class OnRamp & Prep / Primer offerings, Skills Gap Assessment Services, Case Studies, Knowledge Check Quizzes, Skills Immersion Programs & Camps, Collaborative Mentoring Services and Extended Learning Support & Post Training services.

Course Topics / Agenda

Please note that topics, agenda, labs and timing listed below are estimated and may adjust during live delivery based on student experience, interests and participation.

Day 1: Introduction to Domain-Driven Design

1. **What is Domain-Driven Design?**

- Definition of DDD
- Why DDD is important
- Overview of the DDD process
- Lab: Identifying a Domain Model - Students will work together to identify a domain model for a given problem, using pen and paper to map out entities, attributes, and relationships.

2. **Building Blocks of DDD**

- Entities, Value Objects, and Aggregates
- Bounded Contexts and Context Maps
- Ubiquitous Language
- Lab: Identifying Entities, Value Objects, and Aggregates - Students will work together to identify the entities, value objects, and aggregates for a given domain model, using pen and paper to map out their properties and relationships.

Day 2: Implementing Domain-Driven Design

3. **Designing Bounded Contexts**
- Context Maps and Subdomains
- Context Mapping Patterns
- Context Mapping Anti-Patterns
- Lab: Creating a Context Map
- Description: Students will work together to create a context map for a given domain, using pen and paper to identify the subdomains and their relationships.

4. **Strategic Design**
- Strategic Design Principles
- Domain Vision and Core Domain
- Designing Entities and Aggregates
- Domain Events
- Lab: Designing Entities and Aggregates - Students will work together to design the entities and aggregates for a given bounded context, using pen and paper to map out their properties and relationships.

Day 3: Applying Domain-Driven Design

5. **Tactical Design**
- Repository Pattern
- Domain Services
- Domain Events and Event Sourcing
- CQRS
- Lab: Implementing Domain Services and Repositories - Students will work together to implement domain services and repositories for a given bounded context, using pen and paper to map out their interactions.

6. **Applying DDD in Real-World Projects**
- Challenges of applying DDD in practice
- Success stories of applying DDD
- Best practices for applying DDD in a project
- Lab: Applying DDD in a Real-World Project - Students will team to apply DDD principles to a real-world project, using pen and paper to map out the domain model, design bounded contexts, and implement tactical design patterns.

For More Information

For more information about our dedicated skills-focused training services (instructor-led, self-paced or blended), collaborative coaching services, robust Learning Experience Platform (LXP) solutions, Career Experiences, public course schedule, partner programs, courseware licensing options or to see our complete list of course offerings, training solutions and special offers please visit us at www.triveratech.com, email Info@triveratech.com or call us toll free at **844-475-4559.** Our pricing and services are always satisfaction guaranteed.

TRIVERA TECHNOLOGIES • Collaborative IT Training, Coaching & Skills Development Solutions
www.triveratech.com • toll free +1-844-475-4559 • Info@triveratech.com • Twitter TriveraTech

ONSITE, ONLINE & BLENDED TRAINING SOLUTIONS • PUBLIC / OPEN ENROLLMENT COURSES • COURSEWARE LICENSING & DEVELOPMENT
MENTORING • ASSESSMENTS • LEARNING PLAN DEVELOPMENT • SKILLS IMMERSION PROGRAMS / RESKILLING / NEW HIRE / BOOT CAMPS
PARTNER & RESELLER PROGRAMS • CORPORATE TRAINING MANAGEMENT • VENDOR MANAGEMENT SERVICES

Trivera Technologies is a Woman-Owned Small-Business Firm