ONE PAGE SUMMARY

IT & Software › Software Architecture › Domain Driven Design

Introduction to Domain Driven Design

Mastering Domain Driven Design for Effective Software Solutions

DEFINITION

> Domain Driven Design (DDD) is an approach to software development that emphasizes collaboration between technical and domain experts to create a shared understanding of the problem space. It focuses on modeling complex software systems based on the business domain.

KEY CONCEPTS

Bounded Context

A bounded context defines the boundaries within which a particular model is applicable, ensuring clarity and reducing ambiguity.

Ubiquitous Language

A common language used by both developers and domain experts to ensure clear communication and understanding of the domain.

Entities

Objects that have a distinct identity and lifecycle, representing core business concepts within the domain.

Value Objects

Objects that describe certain aspects of the domain but do not have a unique identity, often used to convey attributes.

Aggregates

A cluster of domain objects that can be treated as a single unit, ensuring consistency and integrity of changes.

EXAMPLES

→ An e-commerce system where 'Order' is an entity and 'Shipping Address' is a value object.

→ Using a shared vocabulary between developers and marketing teams to describe customer interactions.

→ Defining a bounded context for user management separate from product management in a large application.

MEMORY TIPS

★ Think of 'Bounded Context' as a 'zone' where specific rules apply.

★ Remember 'Ubiquitous Language' as the 'common tongue' for all stakeholders.

★ Visualize 'Entities' as 'people' in a story, and 'Value Objects' as 'attributes' describing them.

COMMON MISTAKES

✗ Confusing entities with value objects by overlooking identity.

✗ Neglecting to establish clear bounded contexts, leading to ambiguity.

✗ Failing to use ubiquitous language, causing miscommunication among team members.

QUICK RECAP

Domain Driven Design focuses on aligning software development with business needs through shared understanding. Key elements include bounded contexts, ubiquitous language, and the distinction between entities and value objects.