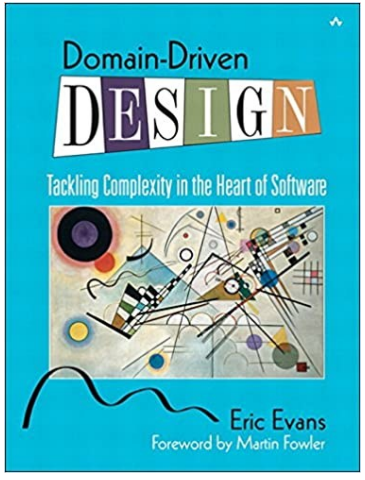
HSLU Hochschule Luzern

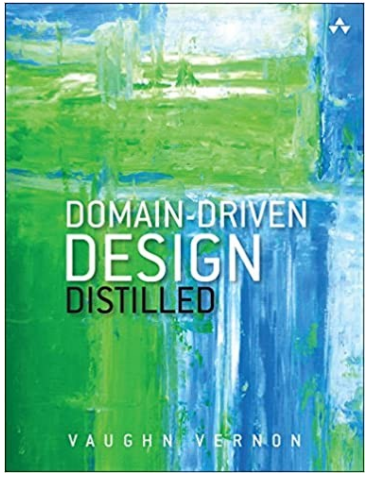
Software Architecture and Techniques

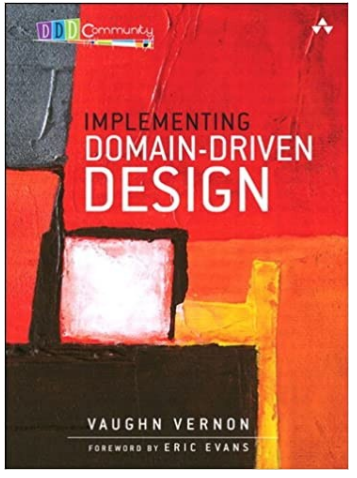
Domain Driven Design Workshop

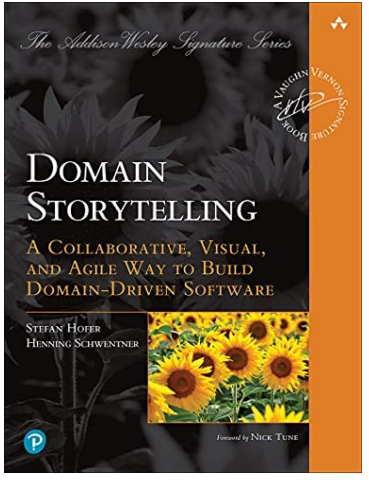
Marcel Baumann, tangly llc

Domain-Driven Design



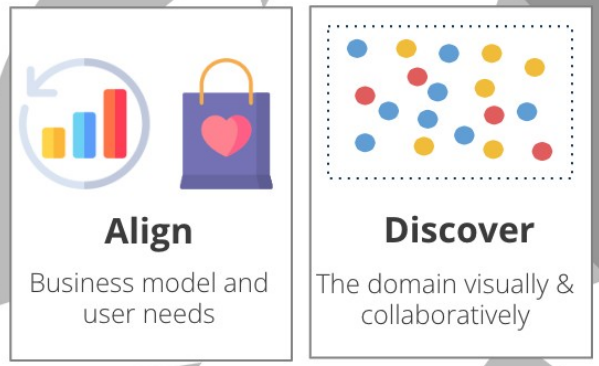






Domain-Driven Design starter modeling process

A starter process for beginners, not a rigid best-practice.
DDD is continuous, evolutionary and iterative design.

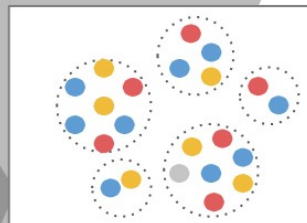


Align

Business model and user needs

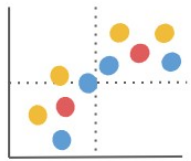
Discover

The domain visually & collaboratively



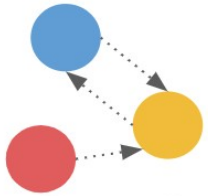
Decompose

The domain into sub-domains



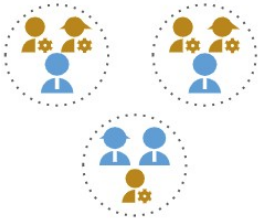
Strategize

Business differentiating core-domains



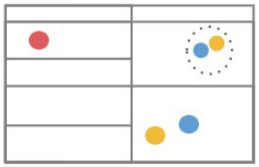
Connect

Sub-domains to form a loosely coupled architecture



Organize

Teams around bounded contexts



Define

Roles & responsibilities for bounded contexts

Code

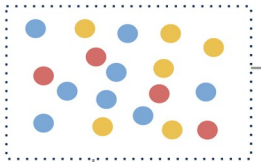
Your bounded context with tactical patterns

Domain-Driven Design

- GitHub DDD Crew

Domain Driven Design On A Page

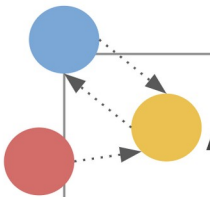
DDD is a philosophy for developing software systems that encourages Domain Thinking at each step of the Software Development Lifecycle



Domain Discovery

Exploratory DDD

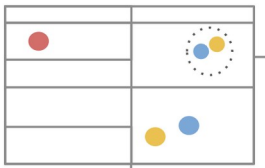
- Model as-is, to-be, and could-be states of the domain
- Model collaboratively and visually so that the whole team* learns the domain and contributes to solution ideas
- Example: Big Picture EventStorming



Software Architecture

Strategic DDD

- Bounded Contexts: Split a large software system into specialised models aligned to areas of the domain
- Integrate bounded context using domain events**
- Identify strategically significant core domains



Software Design

Tactical DDD

- Create models in code which align to the team's shared understanding of the domain
- Use appropriate patterns in each context: entities aggregates, event sourcing, etc

* whole team: all roles involved in product development including business experts
** domain events: business-relevant happening communicated via (technical) events or commands