

Resenha dos Capítulos 4, 5 e 6 – *Domain-Driven Design Reference*

Capítulo 4 – Context Mapping for Strategic Design

O quarto capítulo introduz o conceito de **Context Mapping**, uma peça fundamental do Design Estratégico no DDD. Evans explica que, em sistemas complexos, **múltiplos Bounded Contexts coexistem**, cada um com seu próprio modelo. Como a comunicação entre esses contextos é inevitável, torna-se essencial **mapear as relações** entre eles para minimizar ruídos, inconsistências e dependências frágeis.

O autor apresenta diversos **padrões de relacionamento** entre contextos, cada um descrevendo dinâmicas organizacionais e técnicas:

- **Partnership** – dois times cooperam intensamente e compartilham riscos e decisões.
- **Shared Kernel** – pequenos trechos de código e modelo são compartilhados, exigindo disciplina e forte alinhamento.
- **Customer/Supplier Development** – um contexto fornece funcionalidades que outro consome; o time fornecedor precisa considerar necessidades do cliente.
- **Conformist** – o time consumidor é obrigado a seguir o modelo externo, sem liberdade para negociá-lo.
- **Anticorruption Layer** – camada de tradução para evitar que modelos externos contaminem o interno.
- **Published Language** – criação de uma linguagem compartilhada, formalizada e estável, para integração.
- **Open-Host Service** – definição de um serviço aberto com protocolos de interação claros.
- **Separate Ways** – quando integrar custa mais do que separar, os times seguem caminhos independentes.
- **Big Ball of Mud** – antipadrão para modelos desorganizados e sem fronteiras claras.

A essência do capítulo é mostrar que **DDD vai além de modelagem interna**: ele alcança a **gestão das relações entre equipes, sistemas e fronteiras de conhecimento**. Evans reforça que entender essas relações é tão importante quanto projetar os domínios internos.

Capítulo 5 – Distillation for Strategic Design

O quinto capítulo aprofunda como identificar o que realmente importa dentro de um grande domínio. Evans chama isso de **Distillation**: separar o que é essencial do que é genérico ou secundário.

O objetivo é ajudar equipes a **investirem energia onde há vantagem competitiva**, definindo:

- **Core Domain** – a parte central do sistema, responsável por gerar maior valor estratégico.
- **Generic Subdomains** – áreas necessárias, mas não estratégicas; podem ser compradas, terceirizadas ou padronizadas.
- **Domain Vision Statement** – uma declaração clara que orienta o projeto e explicita o propósito do domínio.
- **Highlighted Core** – técnica de destacar no modelo o que é realmente essencial para direcionar esforços.
- **Cohesive Mechanisms** – mecanismos altamente especializados que resolvem problemas centrais.
- **Segregated Core** – quando o Core Domain é isolado do restante para reduzir ruído e dependências.
- **Abstract Core** – extração de princípios fundamentais quando o domínio é muito complexo para modelagem explícita.

Evans argumenta que, sem destilação, um projeto perde foco e transforma-se em grandes modelos confusos e ineficientes. O capítulo enfatiza que **saber o que ignorar é tão importante quanto saber o que modelar**.

Capítulo 6 – Large-Scale Structure for Strategic Design

No sexto capítulo, Evans discute como **estruturar sistemas grandes** de forma que continuem compreensíveis, coesos e evolutivos. Ele apresenta padrões de organização que dão ao sistema um “**macrodesign**”, ajudando a equipe a enxergar o todo.

Os principais padrões são:

- **Evolving Order** – ordem arquitetural que emerge gradualmente, guiada por princípios e correções constantes.
- **System Metaphor** – metáfora global que orienta decisões de design (herdado da XP).
- **Responsibility Layers** – divisão em camadas de responsabilidade que orientam o fluxo de decisões e dependências.
- **Knowledge Level** – separação entre regras operacionais e definições conceituais que controlam essas regras.
- **Pluggable Component Framework** – construção de componentes intercambiáveis que formam uma arquitetura flexível.

A proposta do capítulo é oferecer **ferramentas de organização macroestrutural**, permitindo que o sistema cresça sem se tornar caótico. Evans destaca que essas estruturas não são rígidas; elas precisam ser adaptadas ao contexto e podem coexistir.

Síntese crítica dos três capítulos

Os capítulos 4, 5 e 6 aprofundam o Design Estratégico dentro do DDD, mostrando que **o verdadeiro desafio de sistemas complexos está nos relacionamentos, prioridades e estruturação em larga escala**. Evans demonstra que:

- **Não basta um bom modelo** → é necessário coordenar equipes e contextos (Cap. 4).
- **Não basta coordenar equipes** → é necessário saber onde colocar energia (Cap. 5).
- **Não basta foco estratégico** → é preciso estruturar o sistema inteiro de forma compreensível (Cap. 6).

Juntos, esses capítulos revelam a visão madura do DDD: **um conjunto de práticas para alinhar arquitetura, modelagem e organização humana**. Evans não prescreve uma fórmula, mas propõe padrões que podem ser combinados, adaptados e refinados conforme o domínio evolui.