构建领域驱动设计知识体系

张逸
http://zhangyi.xyz

DDD CHINA

关于我

- 架构编码实践者
- 领域驱动设计布道师
- 大数据平台架构师
- 敏捷转型咨询师

@逸言

AGENDA

01

领域驱动设计的历史回溯

- 里程碑之一
- 里程碑之二
- 里程碑之三
- 里程碑之四

02

对领域驱动设计的新定位

- 领域驱动设计魔方
- 丰富领域驱动设计方法

03

领域驱动设计参考过程模型

04

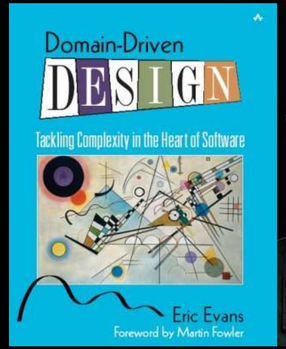
领域驱动设计能力评估模型

01 领域驱动设计的历史回溯

DDD CHINA

里程碑之一

诞生



Domain-Driven Design

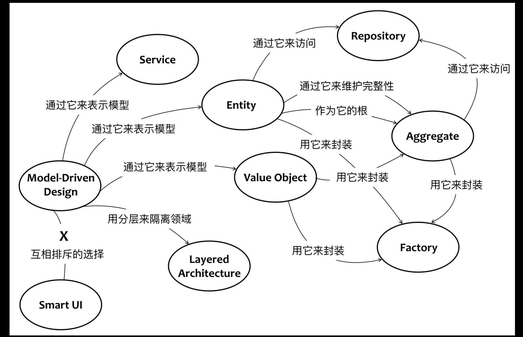
Tackling Complexity in Software

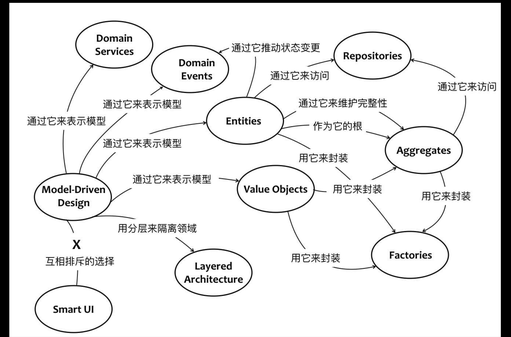
2004年

DDD CHINA

里程碑之二

领域事件的引入





DDD CHINA

里程碑之二

领域事件的引入

Domain Event

CQRS

Event Sourcing

Event Store

EDA

Reactive Programming

Functional Programming

建模范式的改变

对象范式

以“**对象**”为中心

Entity
Value Object
Aggregate
Domain Service
Repository
Factory

事件范式

以“事件”为中心

Domain Event
Event Soucing
Event Store
Application Event
Publisher-Subscriber

函数范式

以“函数”为中心

Algebraic Data Type
Pure Function
Combinator
Monad

架构风格的改变

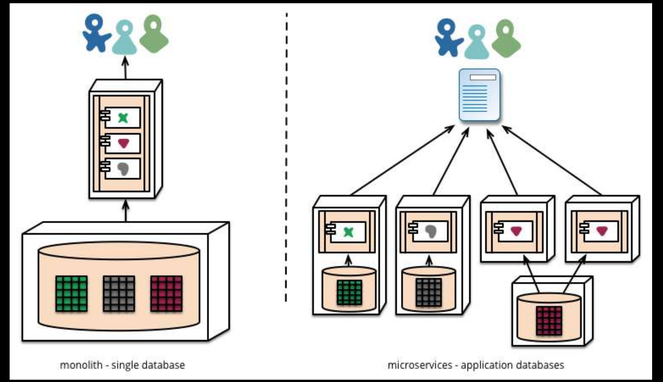
对象范式
分层架构

事件范式
事件驱动架构
CQRS

函数范式
响应式架构

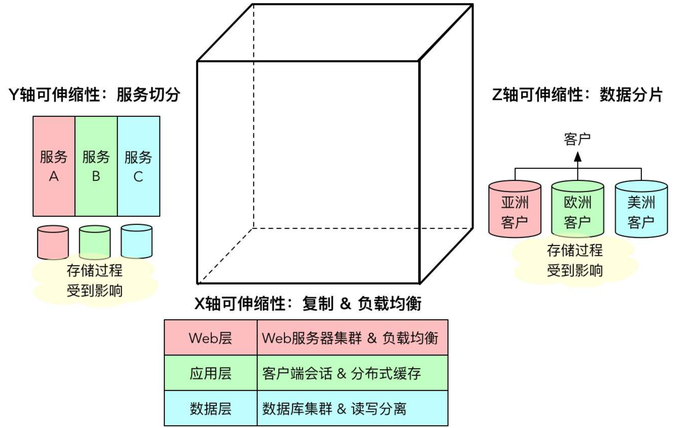
DDD CHINA

里程碑之三
微服务的引入



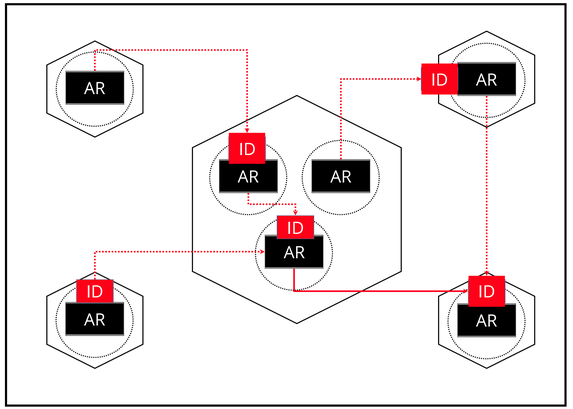
A **microservices** architecture puts each element of functionality into a seperate service and scales by distributing these services accross servers, replicating as needed.

设计理念的改变



数据模型驱动设计
不适合微服务

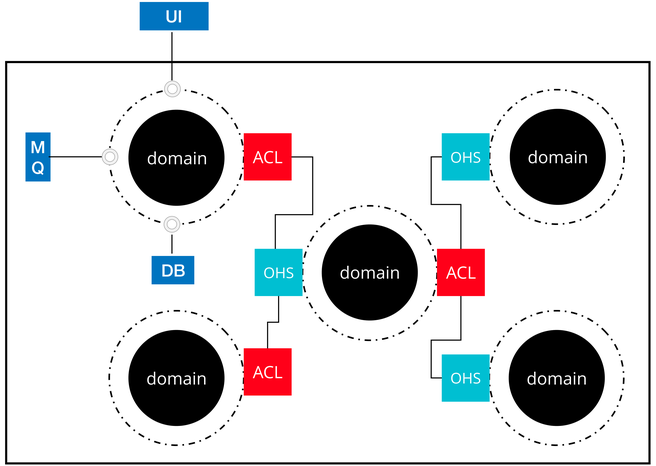
设计理念的改变



- 限界上下文的边界可以是微服务的边界
- 聚合的边界更加稳定，通过ID引用聚合，有利于限界上下文边界的调整，改变通信方式

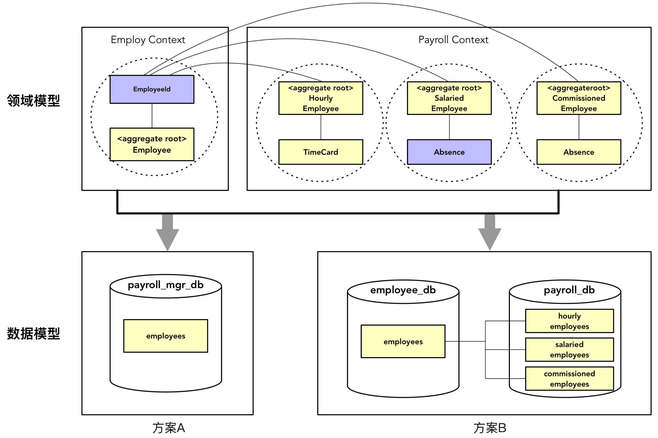
DDD CHINA

设计理念的改变



- 维护好限界上下文边界，有利于从单体架构迁移到微服务架构

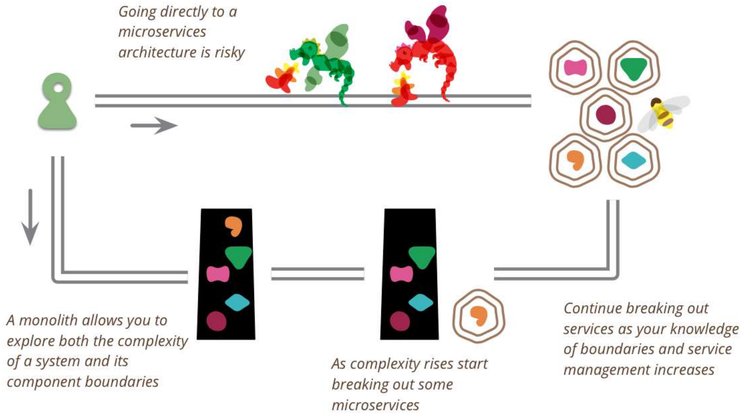
设计理念的改变



- 领域模型与数据模型的分离，有利于从单体架构迁移到微服务架构

DDD CHINA

领域驱动设计带来价值



- 领域驱动设计的模式与实践降低了从单体架构迁移到微服务架构的风险

天作之合

Microservices Architecture

Domain Driven Design

DDD CHINA

DDD CHINA

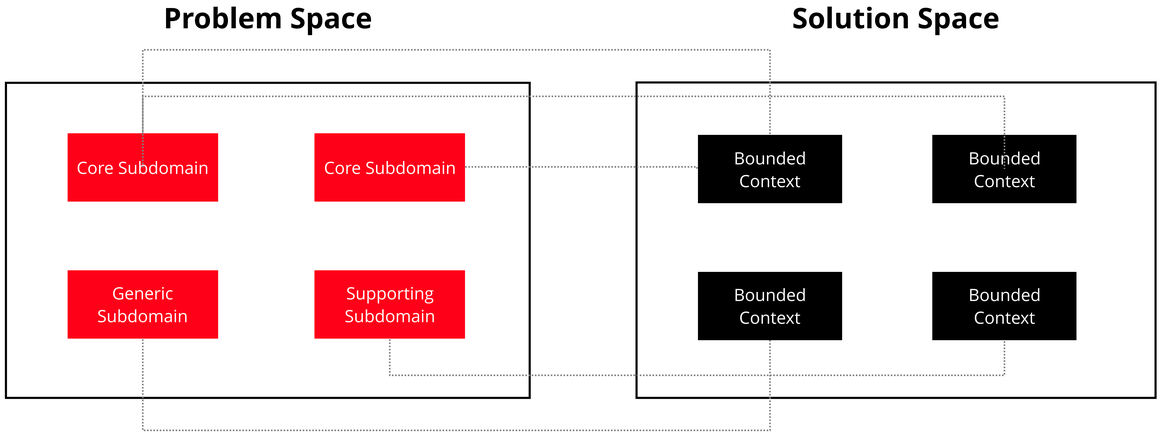
里程碑之四

中台战略的引入

企业级能力复用平台

- 企业级：定义了中台的范围，区分开了单系统的服务化与微服务；
- 能力：定义了中台的主要承载对象，能力的抽象解释了各种各样中台的存在；
- 复用：定义了中台的核心价值；
- 平台：定义了中台的主要形式。

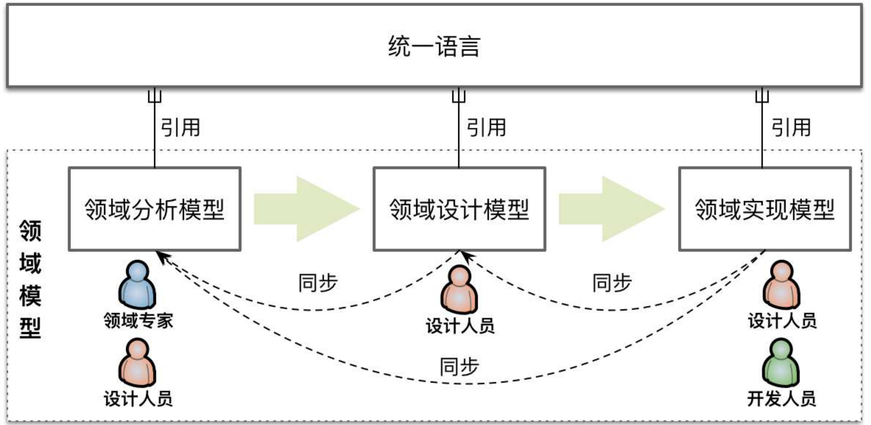
企业级
与能力



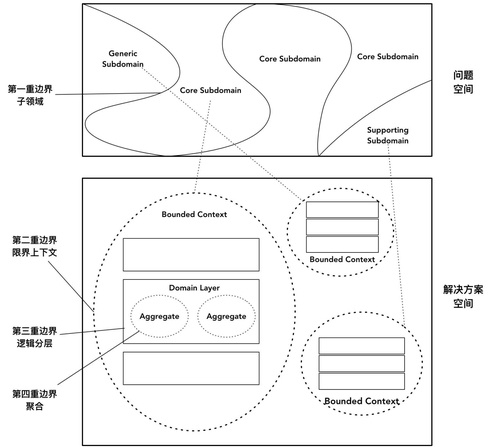
核心子领域：企业核心价值

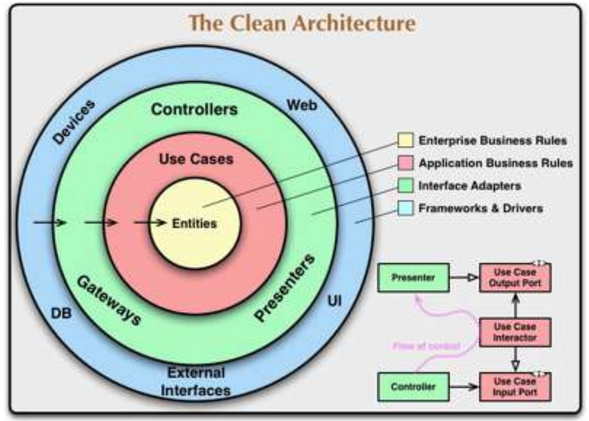
限界上下文：业务能力

领域模型的复用



边界控制与平台沉淀





DDD CHINA

探索.....

ZhongTai Strategy

Domain Driven Design

DDD CHINA

02

对领域驱动设计的新定位

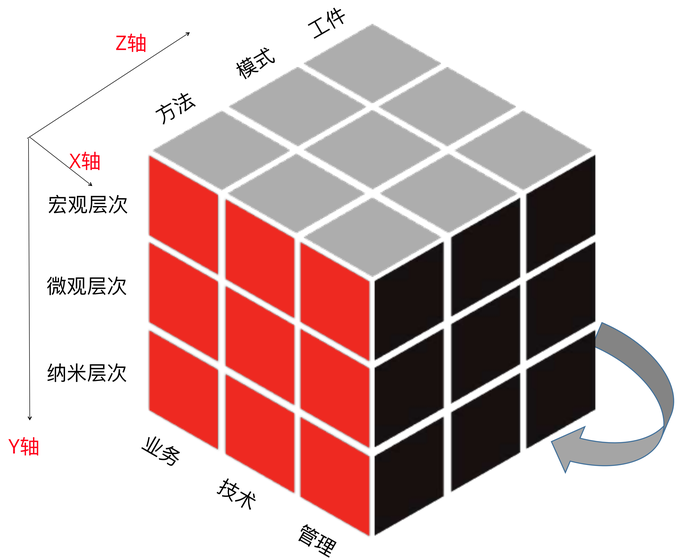
新的定位

Domain Driven Design ~~Technology~~ Philosophy

DDD CHINA

DDD CHINA

领域驱动设计魔方



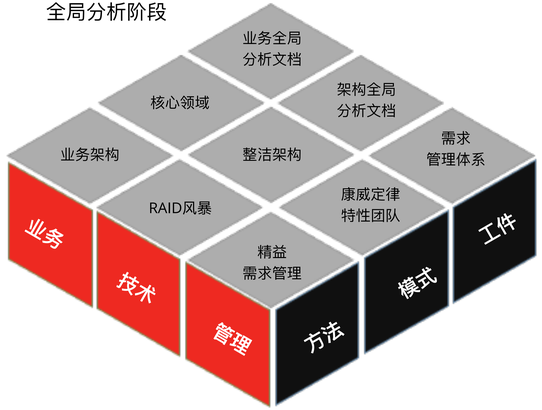
X轴：限定领域驱动设计的内容
Y轴：分离领域驱动设计的层次
Z轴：蕴含了领域驱动设计的实践

宏观层次

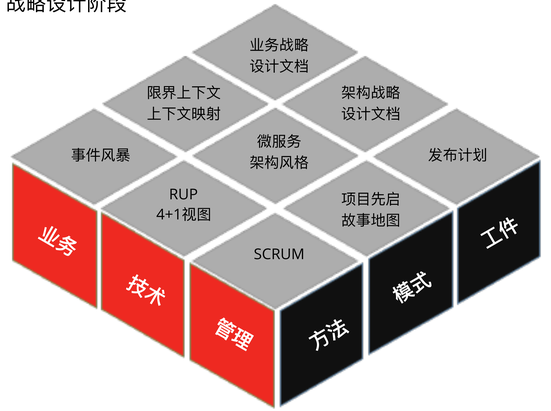
宏观层次是针对整个软件系统开展的战略宏图规划与战略概要设计，通常分为两个阶段：

- 全局分析阶段
- 战略设计阶段

宏观层次
全局分析阶段



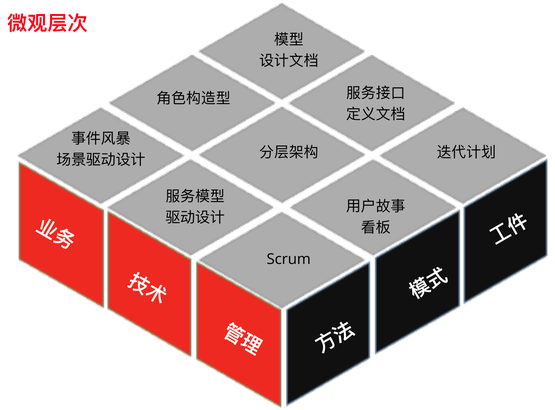
宏观层次
战略设计阶段



DDD CHINA

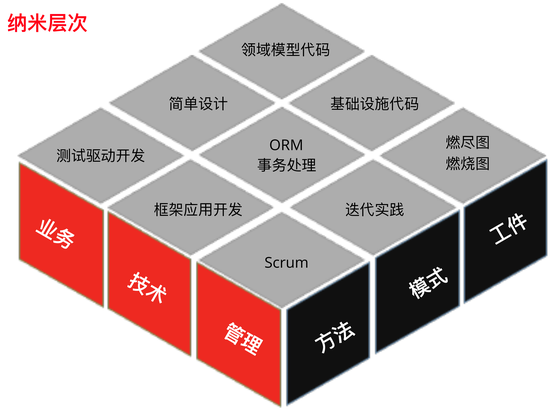
微观层次

微观层次是承上启下的关键环节，是领域驱动设计在团队中落地的重要前提，这个层次输出的工件可以为团队成员提供直接的指导与参考价值。



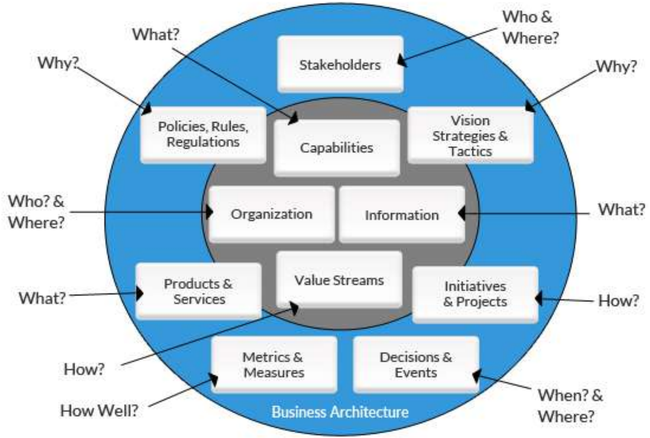
纳米层次

纳米层次对应于软件开发过程的实现阶段。



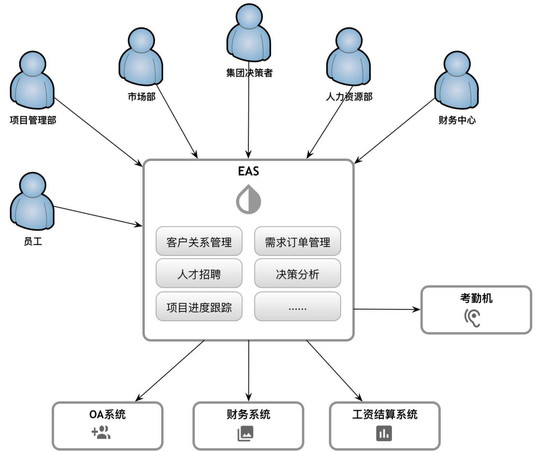
引入
-业务架构

业务架构（Business Architecture）呈现全面的、多维度的业务视角，包括：业务能力、端到端的价值交付、信息和组织结构，以及这些业务视角之间的关系，还包括它们与战略、产品、政策、项目和Stakeholder之间的关系。



引入
-C4模型的 System Context

引入**系统上下文**（System Context）确定系统的边界，了解当前系统与利益相关人之间的关系，并确定它的外部环境，包括与其集成的第三方系统与基础设施。



DDD CHINA

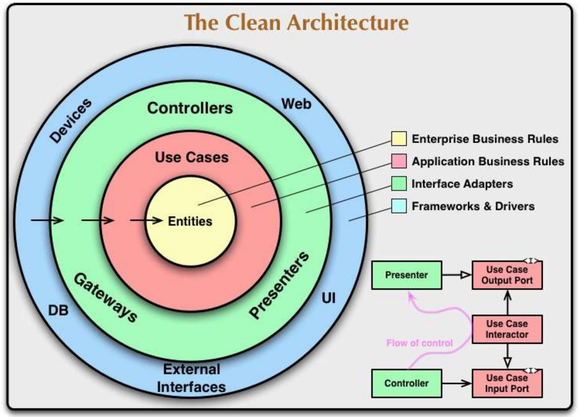
引入

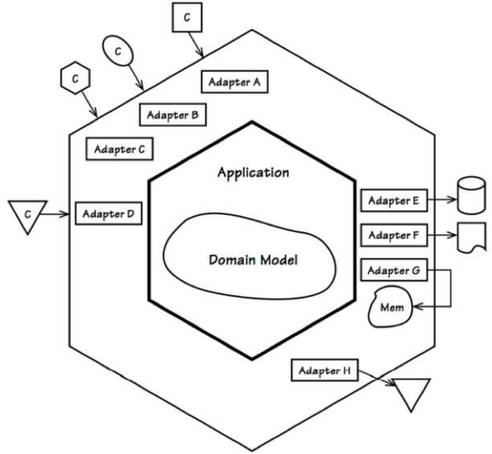
-事件风暴

DDD CHINA

引入

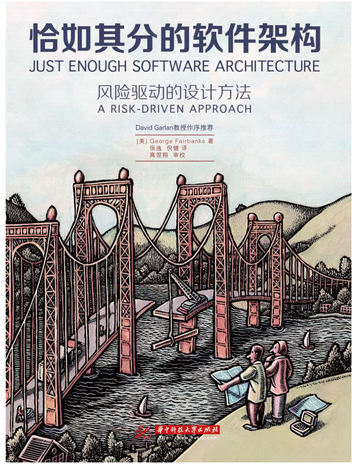
-架构模式


The Clean Architecture
Devices
Controllers
Web
Use Cases
Entities
Gateways
Presenters
DB
UI
External Interfaces
Enterprise Business Rules
Application Business Rules
Interface Adapters
Frameworks & Drivers
Presenter
Use Case Output Port
Use Case Interactor
Flow of control
Controller
Use Case Input Port


C
C
C
C
Adapter A
Adapter B
Adapter C
Application
C
Adapter D
Domain Model
Adapter E
Adapter F
Adapter G
Mem
Adapter H

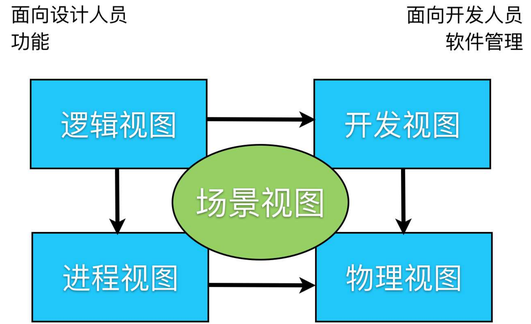
引入

-RAID风暴



引入
-RUP 4+1 视图

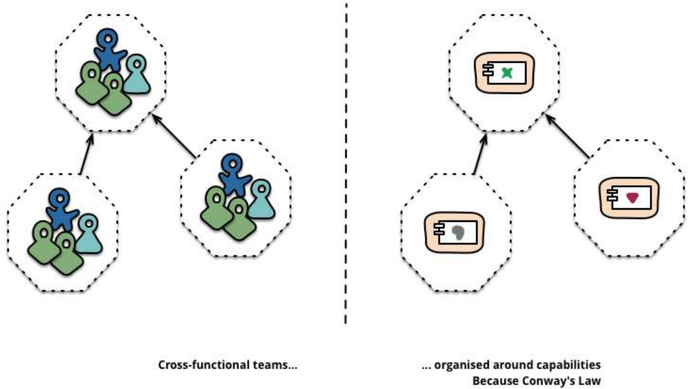
领域驱动设计并没有明确给出架构的设计过程与设计交付物，限界上下文、分层架构、上下文映射仅仅作为战略设计的模式而存在。可以引入**RUP 4+1视图**与领域驱动设计的战略设计结合。



引入

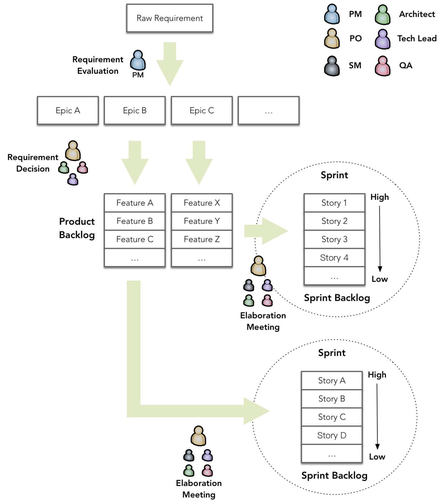
-康威定律

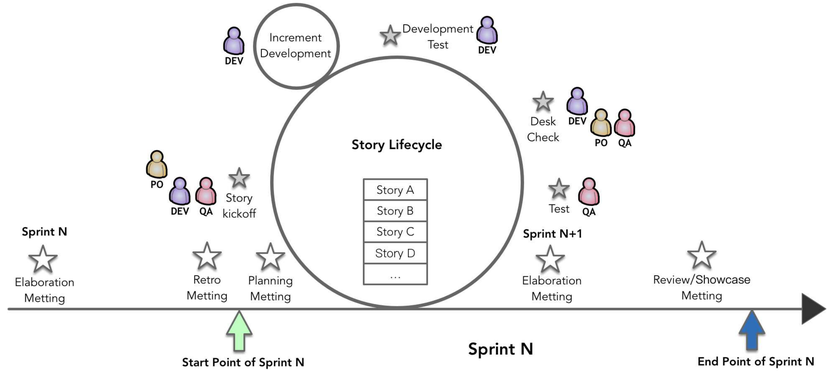
Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations.



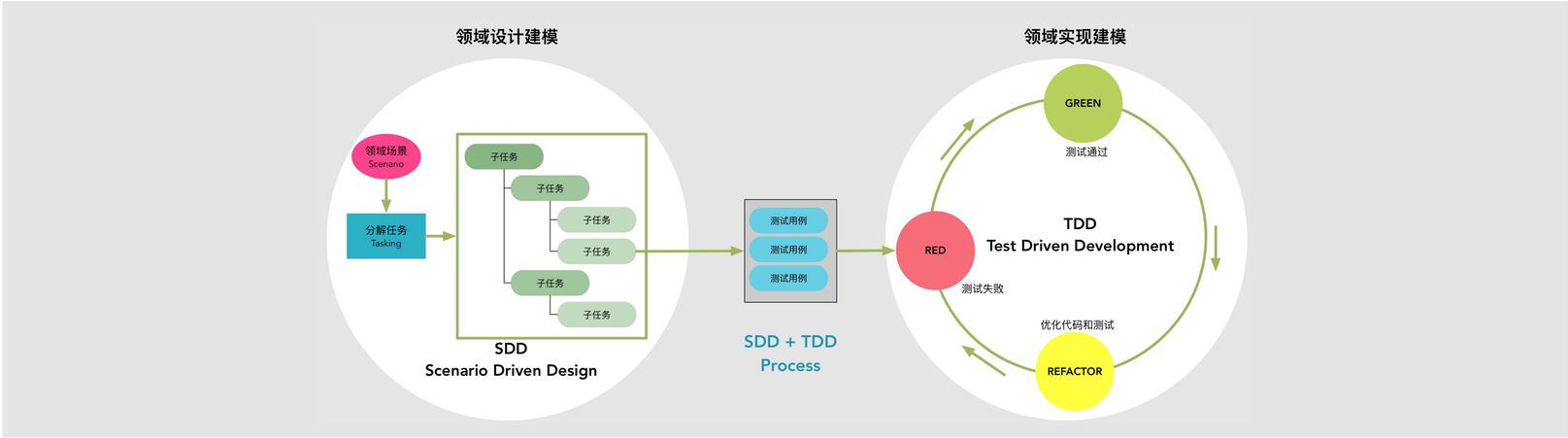
引入

-精益需求管理与敏捷过程管理



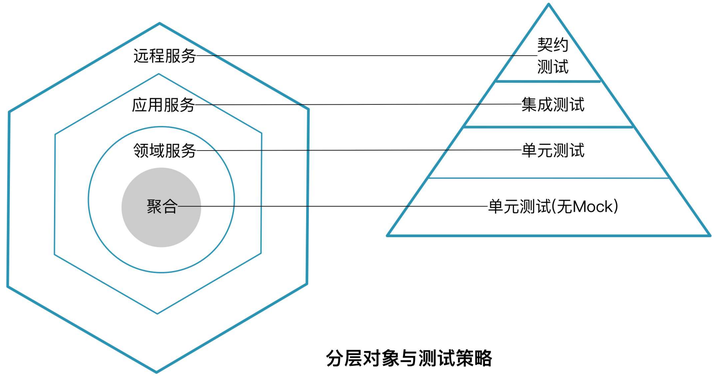


引入
-场景驱动设计与测试驱动开发



引入
-测试策略

领域驱动设计架构的每个逻辑层都定义了自己的控制边界，领域驱动设计的角色构造型位于不同层次。不同的设计元素，决定了它们不同的职责和设计的粒度。层次、职责和粒度的差异，恰好可以与**测试策略**形成一一对应的关系。



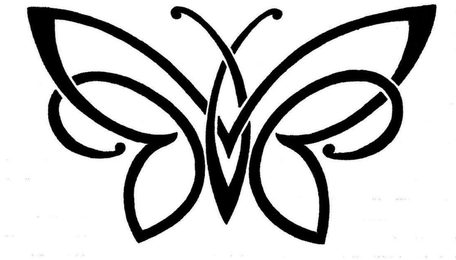
分层对象与测试策略

03

领域驱动设计参考过程模型

参考过程
模型-固化与简化

固化领域驱动设计的过程，提供简单有效的实践方法，建立具有目的性和可操作性的研发过程



DDD CHINA

领域驱动设计参考过程模型

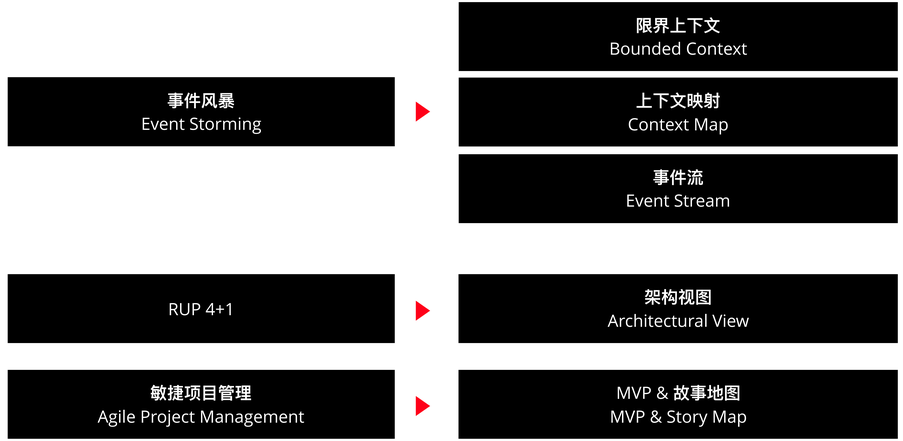
全局分析阶段

价值链 Value Stream	**核心子领域** Core Subdomain
C4模型 C4 Model	**系统上下文** System Context
风险驱动设计 Risk Driven Design	RAID Risks, Assupmtion, Issue, Dependency
精益需求管理 Lean Requirement Management	**需求故事** Epic - Feature - User Story

DDD CHINA

领域驱动设计参考过程模型

战略设计阶段



DDD CHINA

领域驱动设计参考过程模型

领域模型
驱动设计阶段

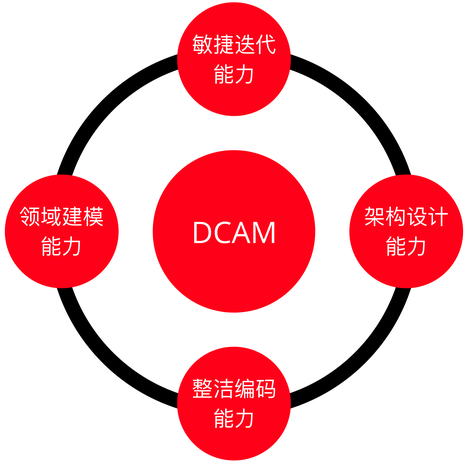
迭代实践模式 Iteration Pratice Pattern	▶	User Story | Kick Off | Desk Check
事件风暴 Event Storming	▶	领域分析模型 Domain Analysis Model
场景驱动设计 Scenario Driven Design	▶	领域设计模型 Domain Design Model
测试驱动开发	▶	领域实现模型 Domain Implement Model

DDD CHINA

04

领域驱动设计能力评估模型

能力评估模型介绍



领域驱动设计能力评估模型

Domain-driven design Capability Assesment Model，DMAM

DCAM是一套评估模型，提供了对软件产品实施领域驱动设计的评估指标，指导团队进行能力的培养和提升。

目前，DCAM仅限于对象范式的领域驱动设计。

DDD CHINA

能力评估模型介绍

DCAM并非一个标准或一套认证体系，更非事先制定和强制执行的评估框架。建立这套模型的目的仅仅是为了更好地实施领域驱动设计，它是一个能够不断演化的评估框架。

根据能力水平，分为三个等级层次。

初始级　　成长级　　成熟级

DCD CHINA

评估维度

敏捷迭代能力

等级	团队	需求	过程
初始级	组件团队，缺乏定期的交流制度	没有清晰的需求管理体系	每个版本的开发周期长，无法快速响应需求的变化
成长级	全功能的特性团队，每日站立会议	定义了产品待办项和迭代待办项	采用了迭代开发，定期交付小版本
成熟级	自组织的特性团队，团队成员定期轮换，形成知识共享	建立了故事地图、建立了史诗故事、特性与用户故事的需求体系	建立了可视化的看板，由下游拉动需求的开发，消除浪费

评估维度

领域建模
能力

等级	领域建模
初始级	采用数据建模，建立以数据表关系为基础的数据模型
成长级	采用领域建模，建模工作只限于少数资深技术人员，并凭借经验完成建模
成熟级	采用事件风暴、四色建模等建模方法，由领域专家与开发团队一起围绕核心子领域开展领域建模

评估维度

架构设计能力

等级	架构	设计
初始级	采用传统三层架构，未遵循整洁架构，整个系统缺乏清晰的边界	采用贫血领域模型，业务逻辑主要以事务脚本实现
成长级	领域层作为分层架构的独立一层，并为领域层划分了模块	采用了富领域模型，遵循面向对象设计思想，但未明确定义聚合和资源库
成熟级	建立了系统层次与限界上下文层次的系统架构，遵循了整洁架构，建立了清晰的限界上下文与领域层边界	建立了以聚合为核心的领域设计模型，职责合理地分配给聚合、资源库与领域服务

评估维度

整洁编码
能力

等级	编码	自动化测试
初始级	编码以实现功能为唯一目的	没有任何自动化测试
成长级	方法和类的命名都遵循了统一语言，可读性高	为核心的领域产品代码提供了单元测试
成熟级	采用测试驱动开发编写领域代码，遵循简单设计原则	具有明确的测试战略，单元测试先行

DDD CHINA

与我交流

「逸言」
公众号

「TOP DDD」
知识星球

DDD CHINA