

IPC 2019: Multiple Models - Strategic Design with DDD

@hschwentner

Autor von "Domain Driven Design Kompakt" :)

<https://photos.app.goo.gl/TYFM7VNTzotyuxW7>

Model ist in der Domain verwurzelt

Domain Model - wir modellieren die existierende reelle Domain

Domain Modell hat keine technischen Worte sondern fachliche (rich domain behaviour)

wächst über die Zeit - One Big Model

Realität: verschiedene gemixte Modelle

verschiedene mentale Modelle in verschiedenen Entwickler-Gehirnen

One Big Model -> Big Ball Of Mud

große Modelle führen zu "Whack a mole" Issues - eins erledigt ploppt ein neues auf

(Max & Moritz ist in Niederlande nicht so bekannt)

Model ist ein Tool die Welt zu verstehen, zumindest den essentiellen Teil davon
verschiedene Modelle für verschiedene Kontexte

Bsp.: verschiedene Modelle um die Welt *geografisch* abzubilden

gibt keine Karte/Modell die *alle* Dinge auf einer Karte abbilden (Ressourcen, Flüsse, Berge, Städte, Straßen etc.)

Bounded Context = Software Modul

je BC eine Codebasis

je BC max. ein Team

je BC eine DB

es braucht eine Weile die Schönheit dahinter zu verstehen

Strategic Design

zerschneiden in einzelne BCs und Models

hat man es nicht baut man ein schwarzes Loch

(zeigt man Klassen und ihre Verbindungen in einen Schema, bilden die kreuzenden Linien eine Fläche in der Mitte...)

das hat so viel Gravitation dass es alle neuen Klassen sofort verschluckt

Crossfunktionales-Team ist in sich Full-Stack und ohne andere Teams handlungsfähig

Layers:

namespace Foo\Bar{UI, Application, Domain, Infrastructure}

namespace Foo\Bar -> hier liegen die Interfaces für BC-Interop

Core Domain

nicht alle Bestandteile können mit guter Architektur, Coverage, Testabdeckung gebaut werden

welche ist besonders wichtig? Wo findet die Wertschöpfung statt?

Supporting Domains

unterstützende Domains, z.b. Login, Mails

Wie starten?

Nicht alles auf einmal, aber irgendwann starten

evolutionär weiterentwickeln, ist ein langer Weg

nach jedem Schritt muss die Software lebensfähig sein

Monolith -> Microservice => Distributed Big Ball of Mud

Monolith -> Modulith (-> Microservice)

1. Model your domain
2. Split
3. ?

LeasingNinja.io (Java, später PHP)