Domain Driven Design damals und heute

FrOSCon 2017 / 20.08.17

Christoph Baudson / @sustainablepace

REWE digital

Christoph Baudson

- **Organisator** des **Domain Driven Design Meetups** Köln/Bonn
- @sustainablepace
- **Softwareentwickler** bei **REWE Digital** seit 08/2015

REWE

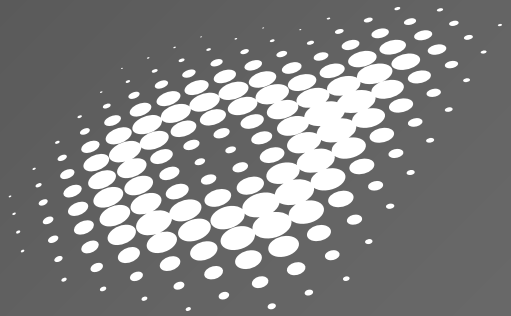
Turnover
>54 bn

Employees
>330.000

Shops
>15.000

Industries
Food Retail, Tourism, DIY

History
almost
90 years

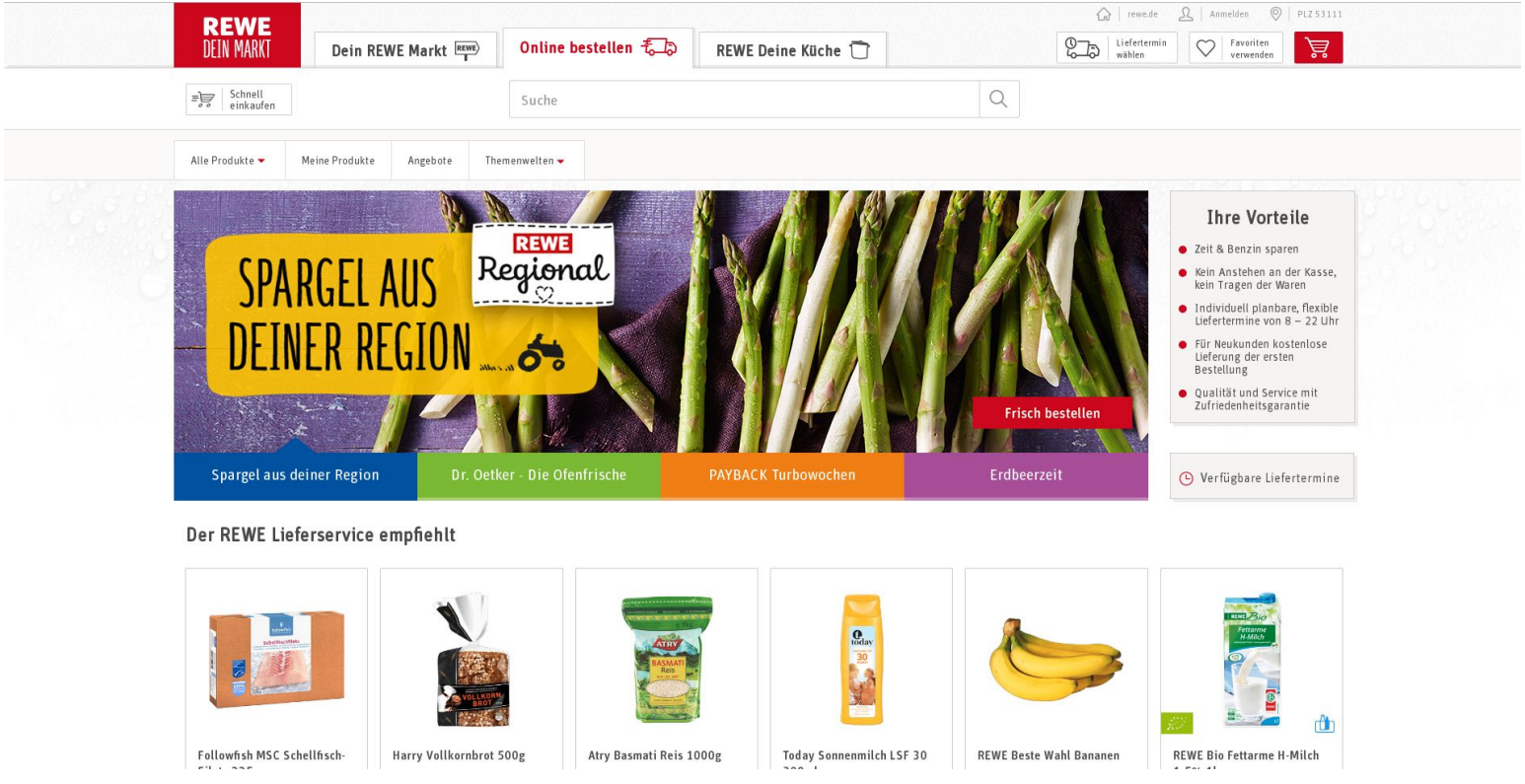


REWE digital

https://rewe-digital.com/jobs.html

REWE Lieferservice für Lebensmittel

→ shop.rewe.de



REWE digital

Woher kommt Domain Driven Design?

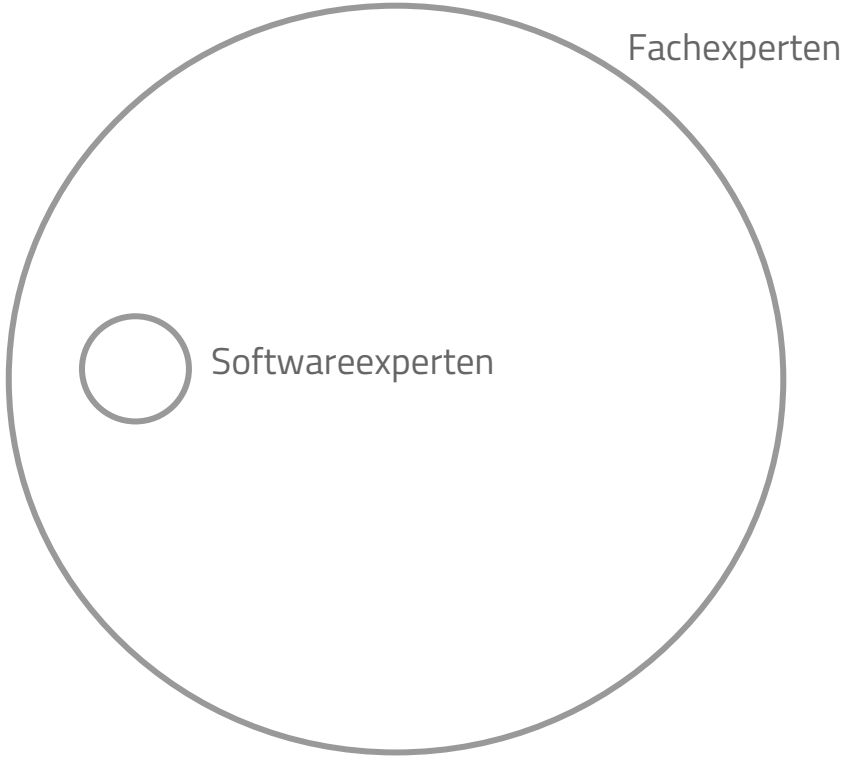
(1960s-90s)

David West - The Past and Future of Domain-Driven Design

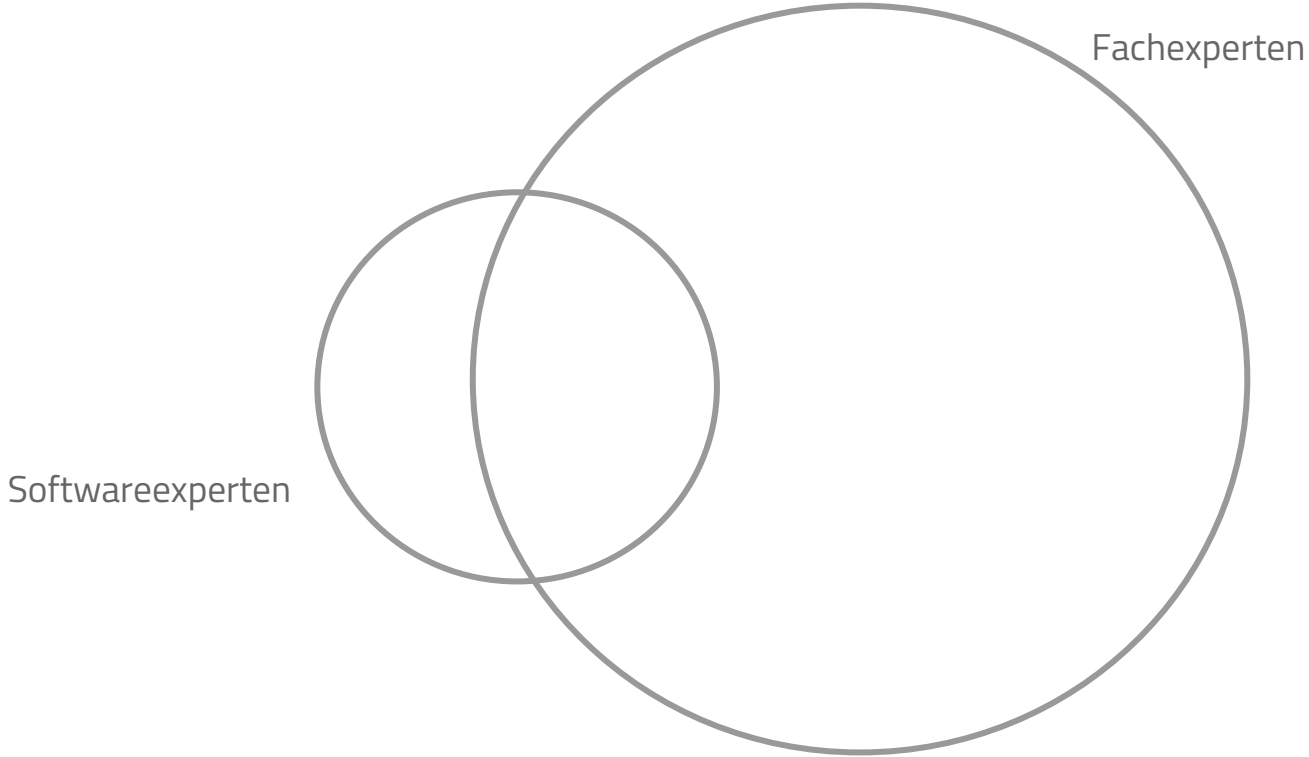
Domain-Driven Design Europe

Abonniert 2.500

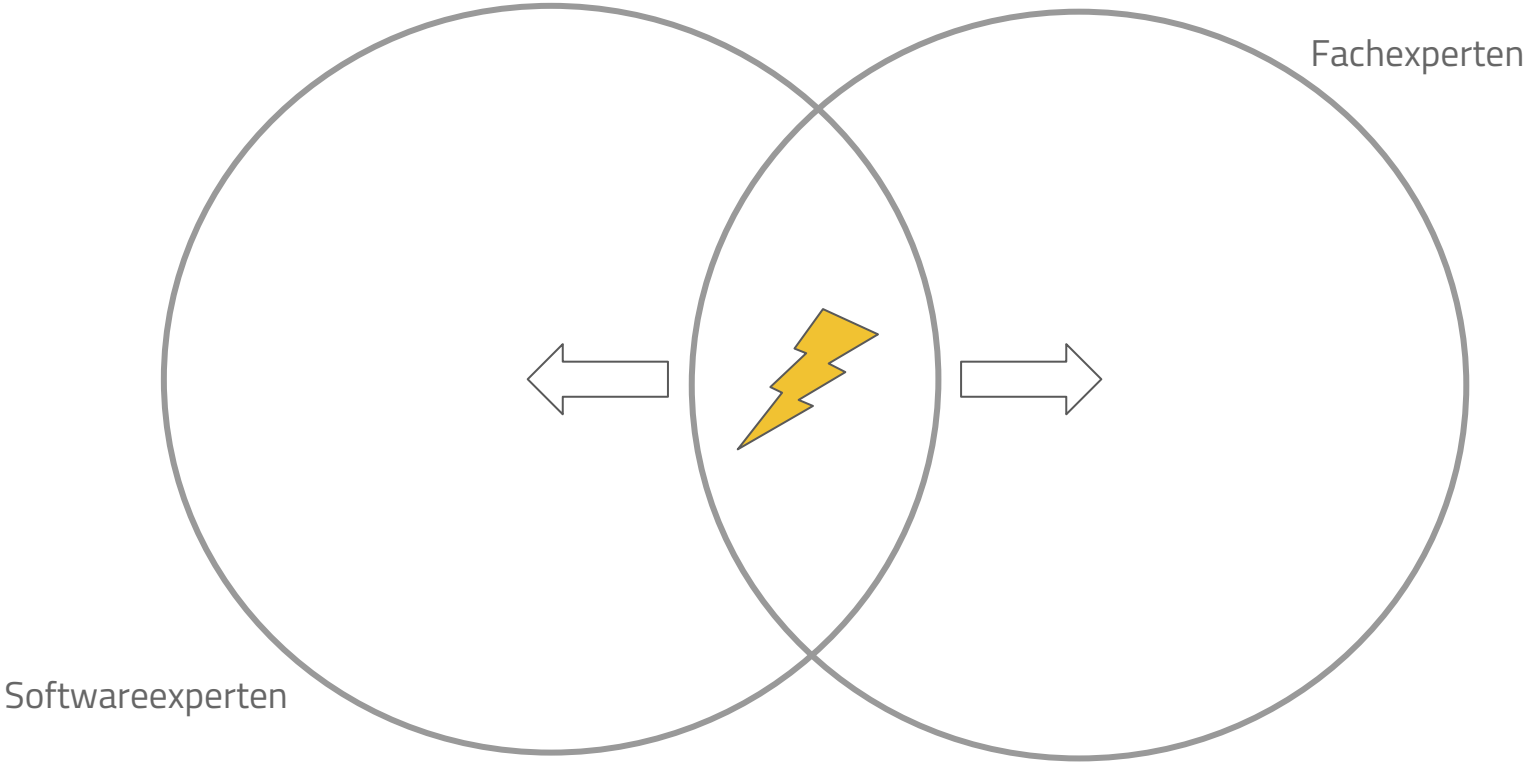
3.020 Aufrufe



Software wird ausschließlich von Fachexperten geschrieben



IT Sektor wächst - zu wenig Fachexperten mit Know How



Informatik

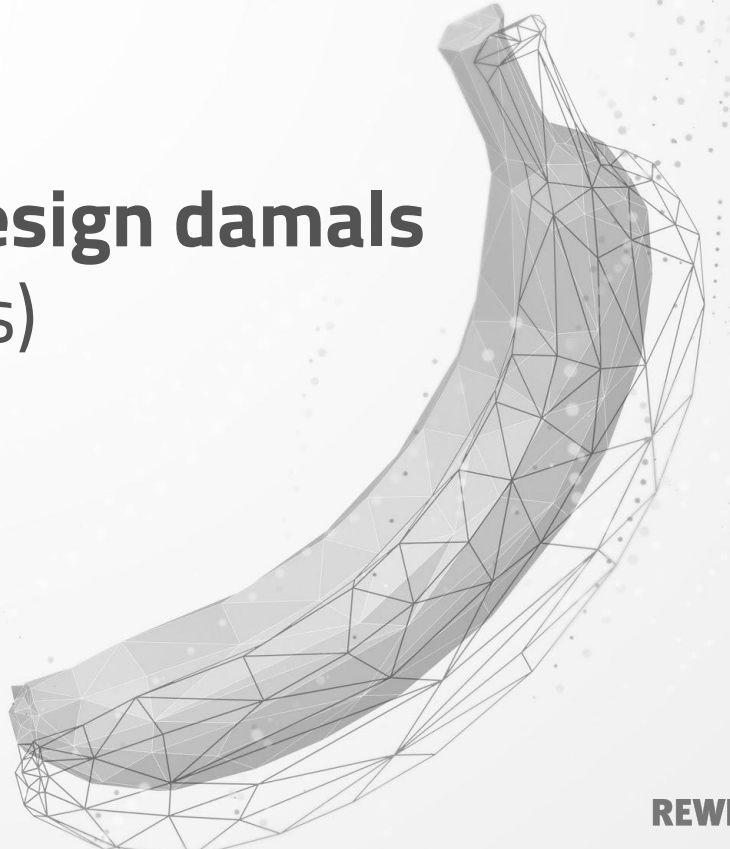
Professionalisierung

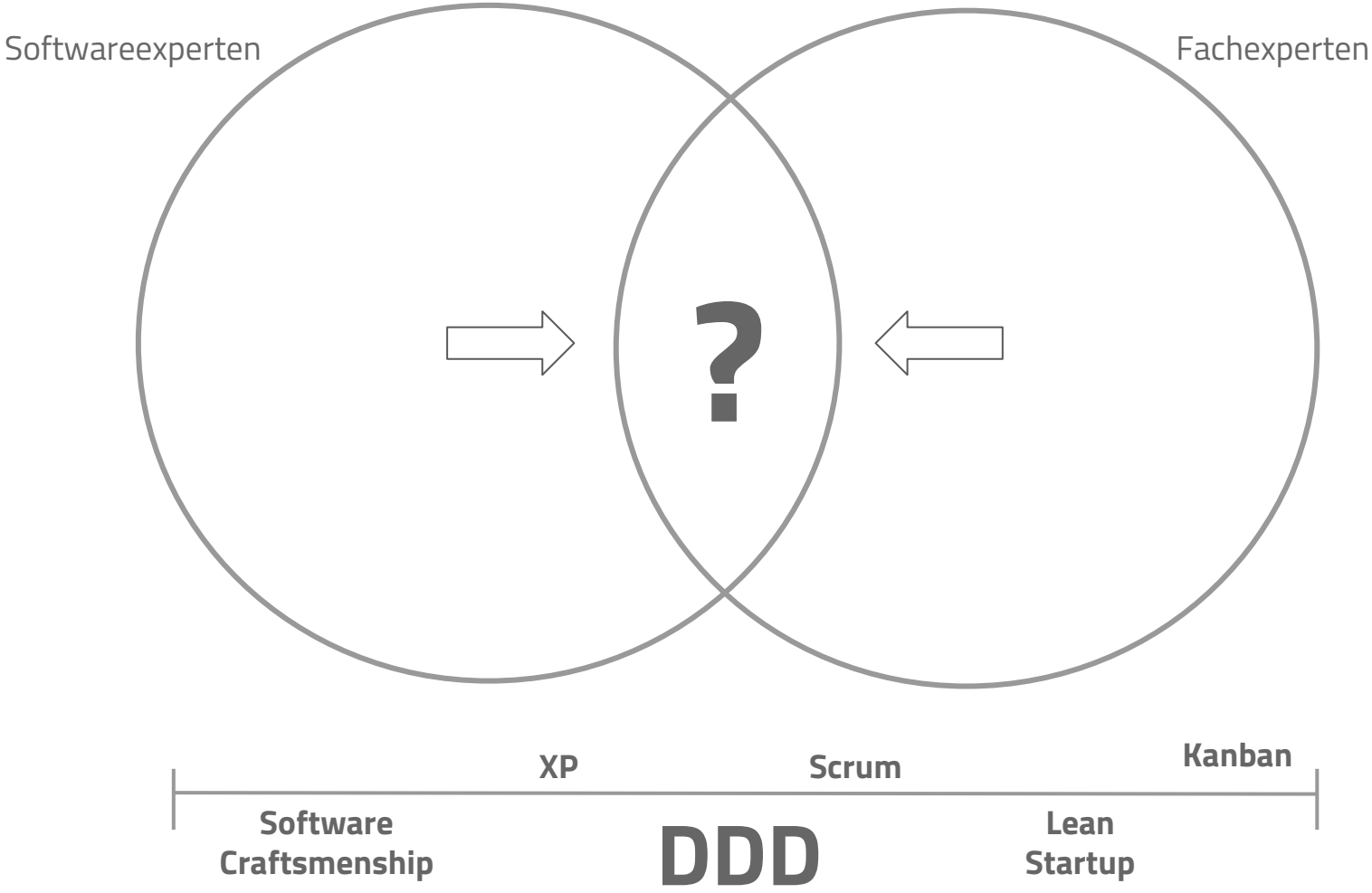
Outsourcing

Hindernis

REWE digital

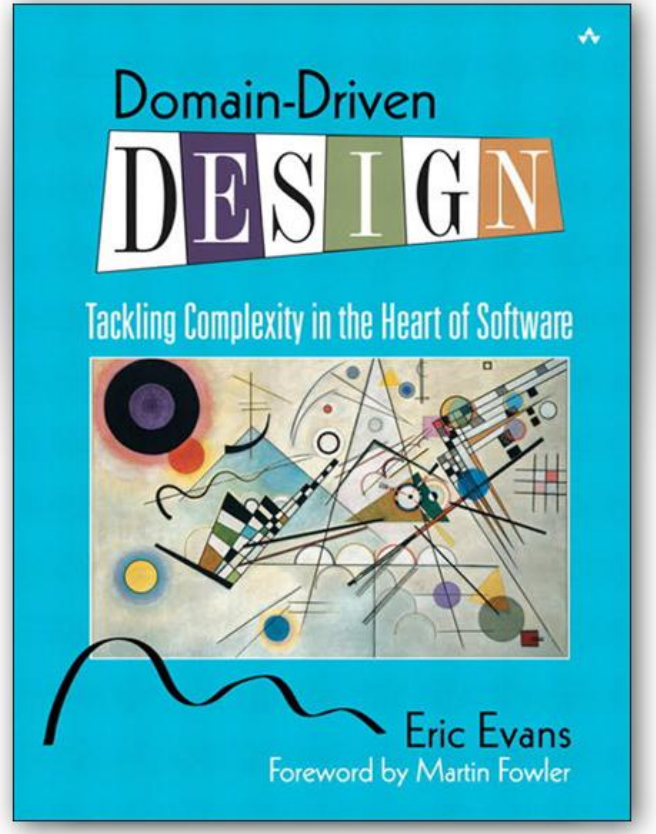
Domain Driven Design damals
(2000s)




Softwareexperten
Fachexperten
?
XP
Scrum
Kanban
Software Craftsmenship
DDD
Lean Startup

Domain Driven Design
(2004)

Eric Evans



Domain Driven Design 1x1

- **Domain**
- Domain Model
- Ubiquitous Language

- Beispiel

Domain

- **Fachlichkeit / Fachlogik** eines Geschäftsfelds

- **Anwendungsgebiet / Einsatzbereich** der Software

Domain @ REWE

Domain Driven Design 1x1

- Domain ✔
- **Domain Model**
- Ubiquitous Language

- Beispiel

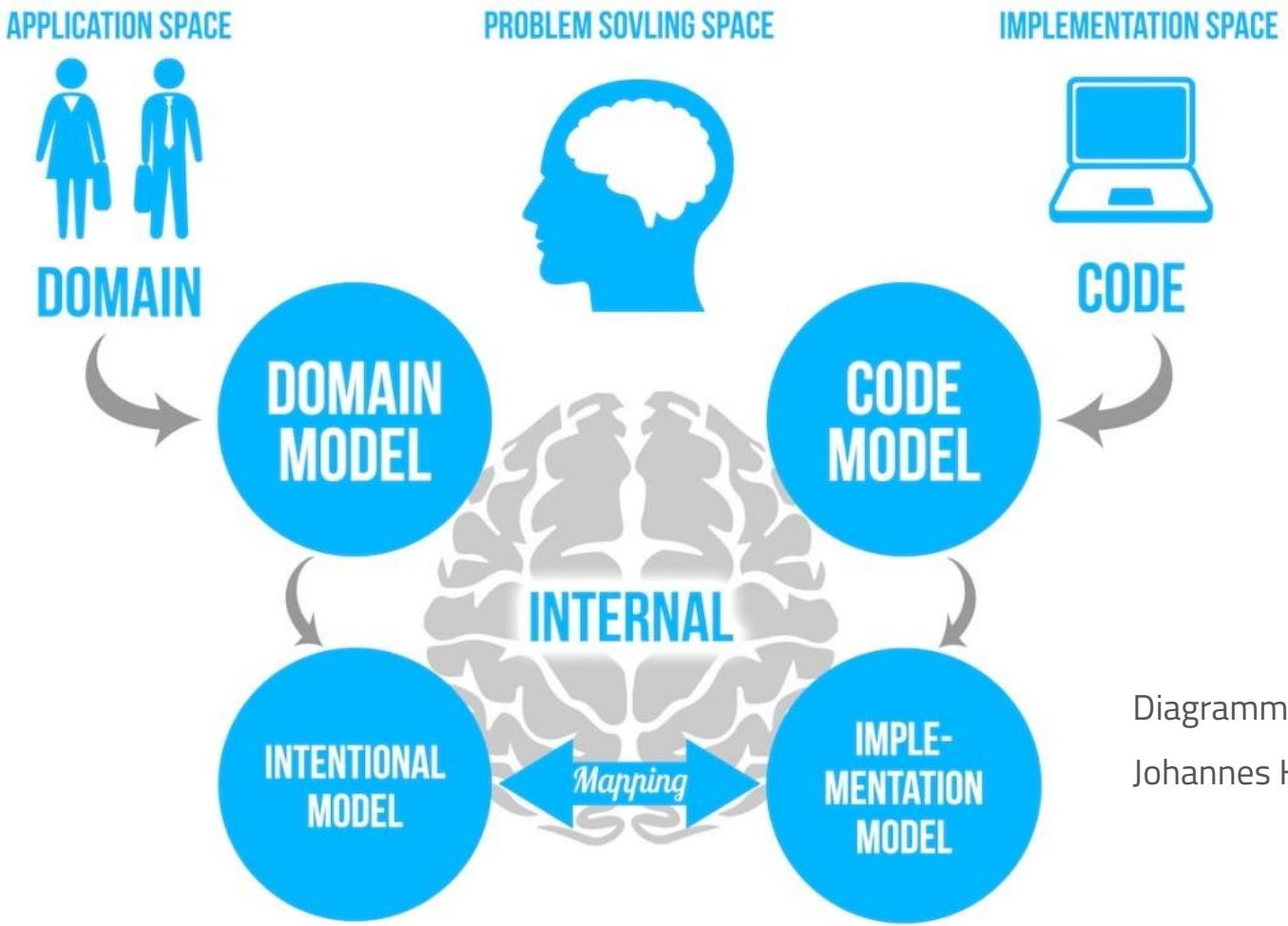


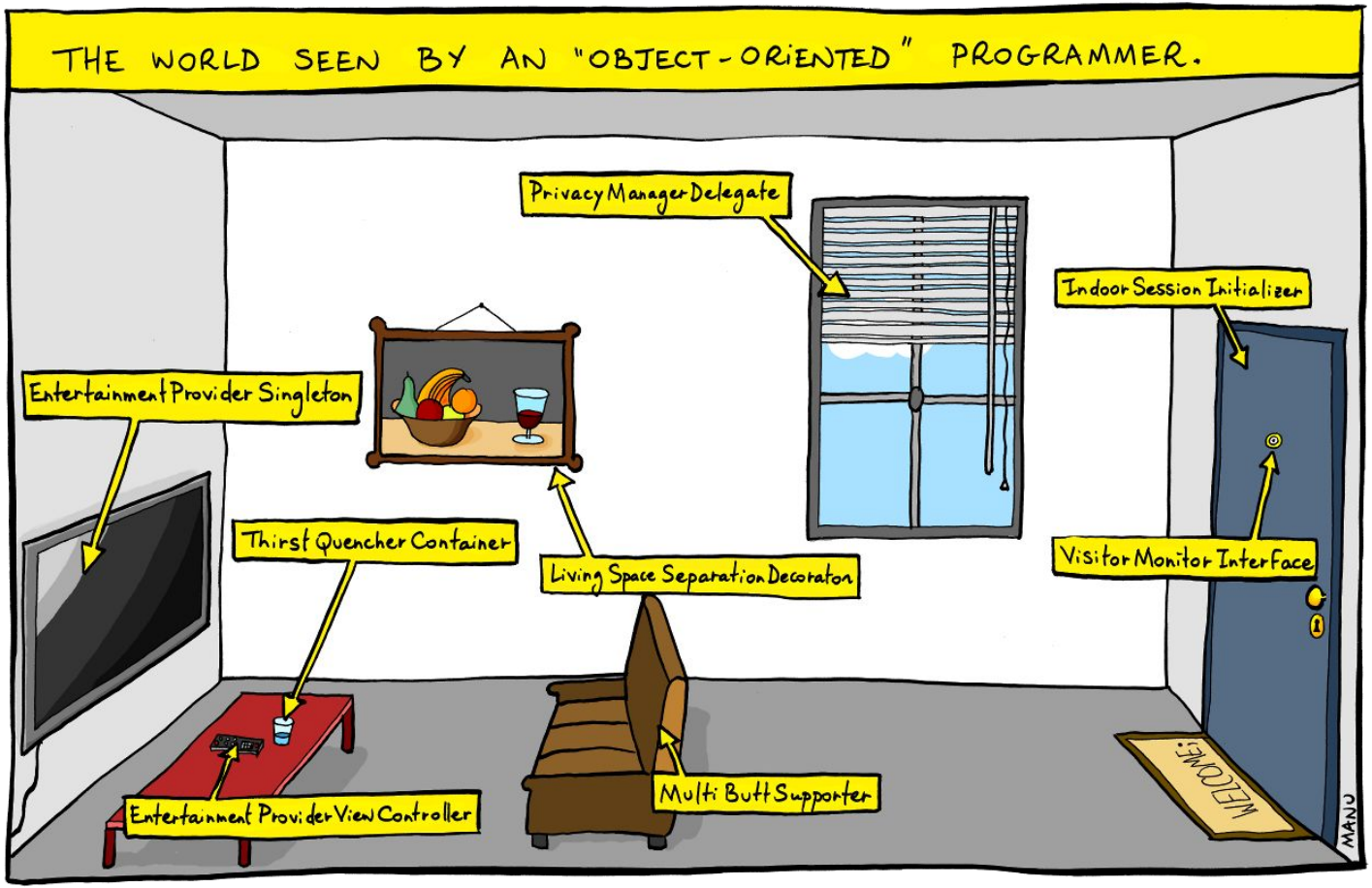
Diagramm von

Johannes Hofmeister

Domain Driven Design 1x1

- Domain ✔
- Domain Model ✔
- **Ubiquitous Language**

- Beispiel

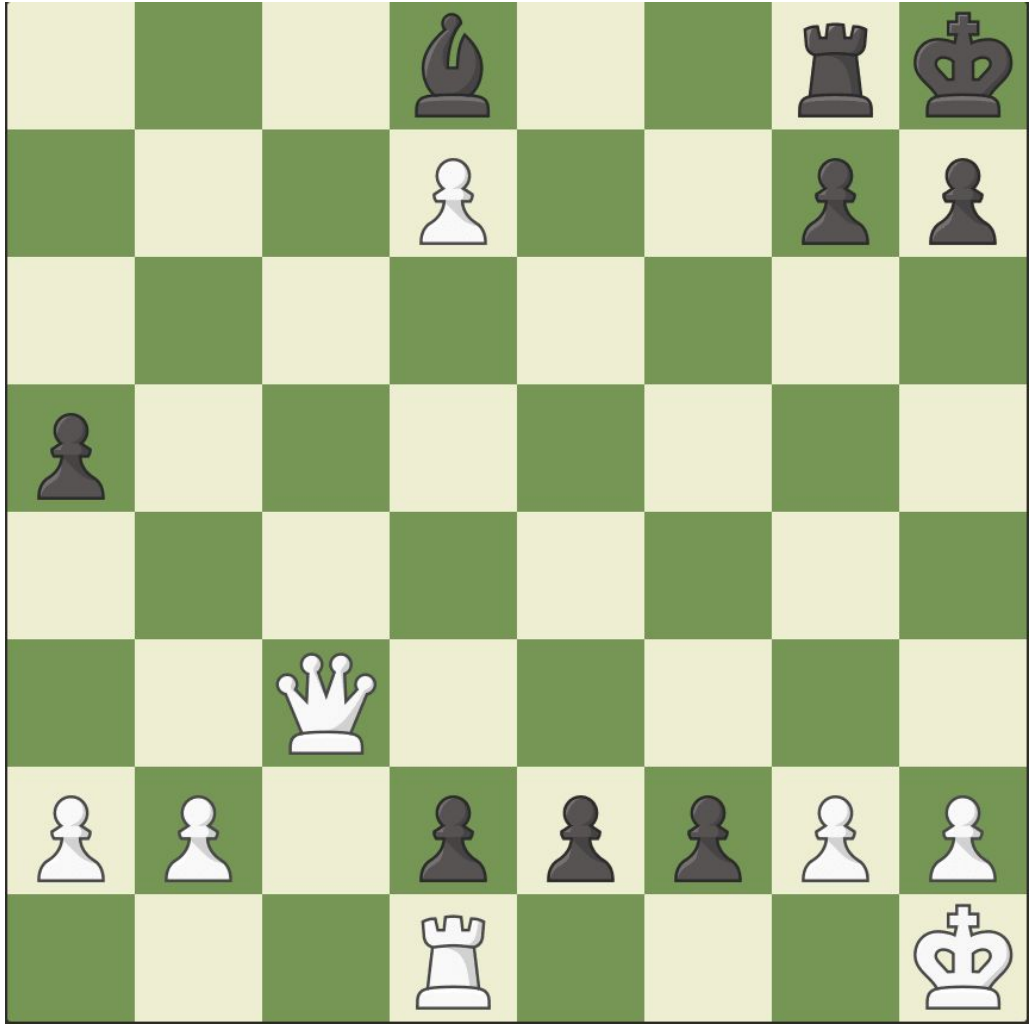


http://bonkersworld.net/object-world

Domain Driven Design 1x1

- Domain ✔
- Domain Model ✔
- Ubiquitous Language ✔

- **Beispiel**



Domain Modellierung

Abstraktion: Domain ⇒ Modell

- Partie

Domain Modellierung

Abstraktion: Domain ⇒ Modell

- Partie
  - Brett

Domain Modellierung

Abstraktion: Domain ⇒ Modell

- Partie
  - Brett
    - Felder

Domain Modellierung

Abstraktion: Domain ⇒ Modell

- Partie
  - Brett
    - Felder[a..h][1..8]

Domain Modellierung

Abstraktion: Domain ⇒ Modell

- Partie
  - Brett
    - Felder[a..h][1..8]
      - Optional: Figur { König, Dame, Turm, Springer, … }

Domain Modellierung

Abstraktion: Domain ⇒ Modell

- Partie
  - Brett
    - Felder[a..h][1..8]
      - Optional: Figur { König, Dame, Turm, Springer, … }
        - Weiß | Schwarz
        - Gangart

Domain Modellierung

Abstraktion: Domain ⇒ Modell

- Partie
  - Stellung (Default: Grundstellung)
  - Spieler { Weiß, Schwarz }
  - { Spieler } ist am Zug (Default: Weiß)

Domain Modellierung

Abstraktion: Domain ⇒ Modell

- Partie
  - Stellung (Default: Grundstellung)
  - Spieler { Weiß, Schwarz }
  - { Spieler } ist am Zug (Default: Weiß)
  - *ziehen*(Figur, Feld von, Feld nach)

Domain Modellierung

Abstraktion: Domain ⇒ Modell

- Partie
  - Stellung (Default: Grundstellung)
  - Spieler { Weiß, Schwarz }
  - { Spieler } ist am Zug (Default: Weiß)
  - *ziehen*(Zug)

Domain Modellierung

Abstraktion: Domain ⇒ Modell

- Partie
  - Stellung (Default: Grundstellung)
  - Spieler { Weiß, Schwarz }
  - { Spieler } ist am Zug (Default: Weiß)
  - *ziehen*(Zug)
  - Patt, Matt, Remis

Domain Modellierung

Abstraktion: Domain ⇒ Modell

Konzepte die man z.B. **nicht** modellieren möchte

- Bedenkzeit, Schachuhr
- Aufgeben, Remis anbieten
- Spielanalyse: Eröffnung, Springergabel, Zugzwang, Abzugsschach, Opfer, Freibauer, ...
- ...

Domain Modellierung

Integrität des Modells schützen

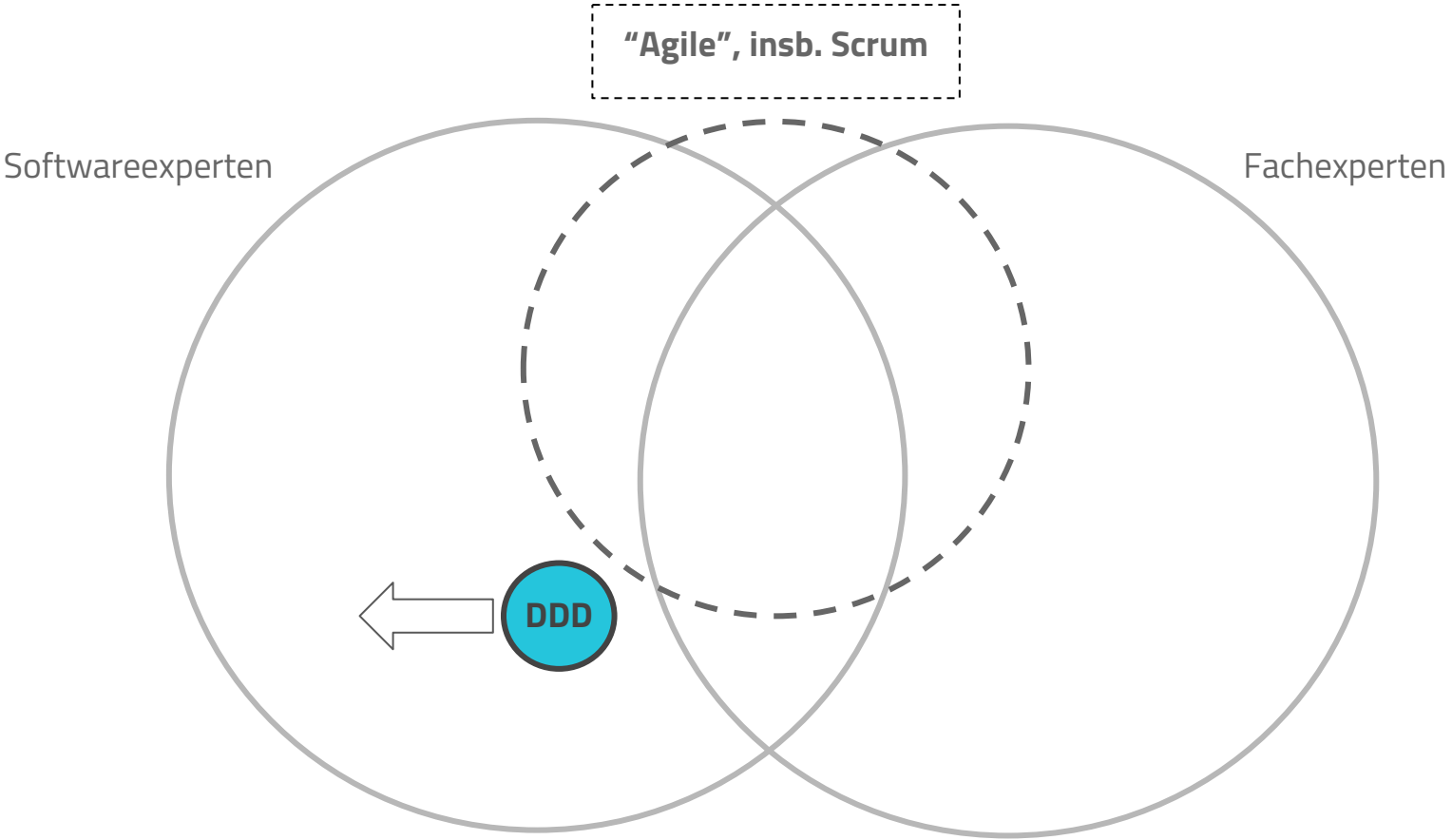
z.B. in der Methode *ziehen(Zug)*

- Figuren dürfen sich nur ihrer Gangart entsprechend bewegen
- Der eigene König darf nicht ins Schach geraten
- Ein Spieler darf nicht zweimal hintereinander ziehen
- Immer höchstens eine Figur auf einem Feld

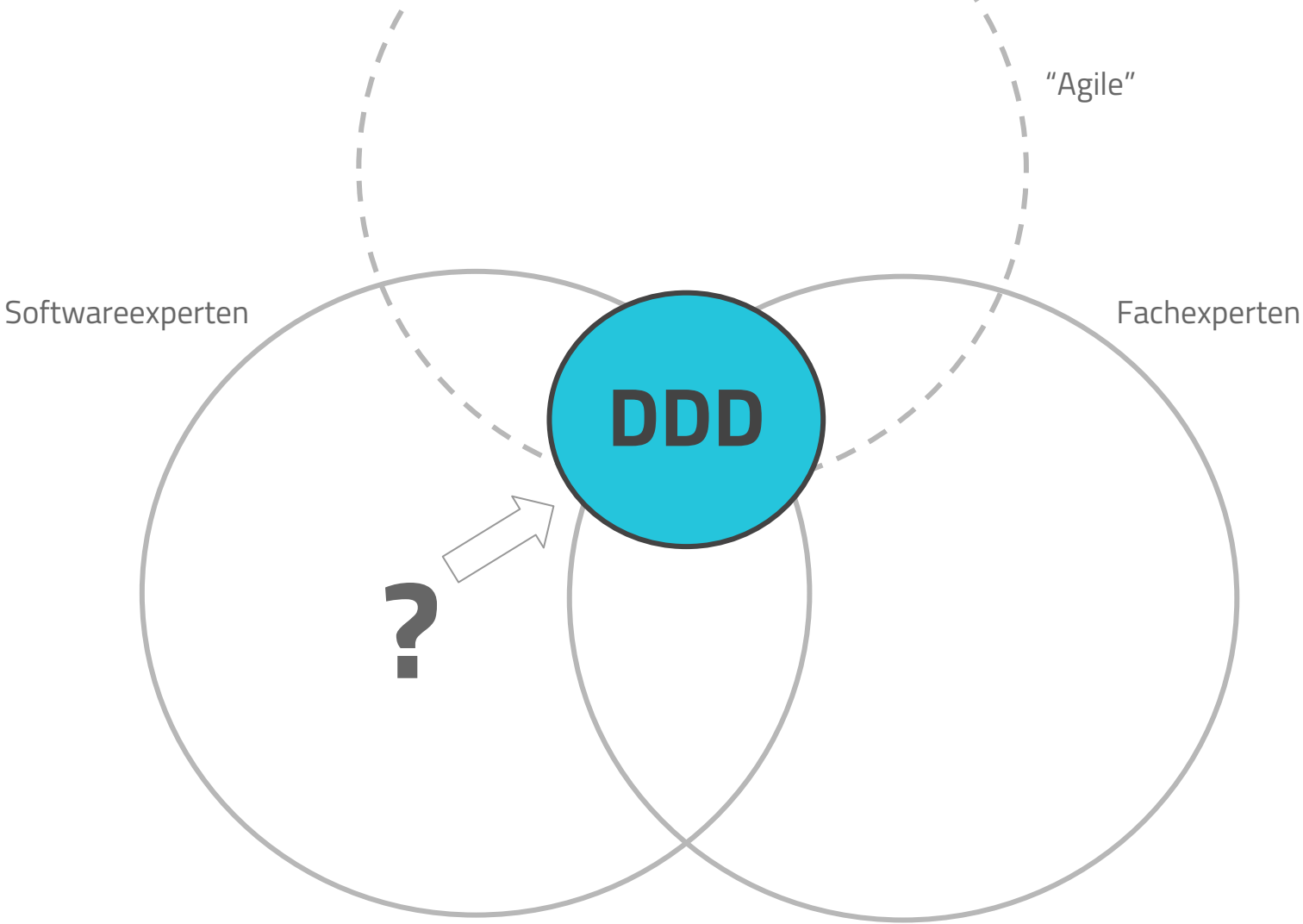
Domain Driven Design 1x1

- Domain ✔
- Domain Model ✔
- Ubiquitous Language ✔

- Beispiel ✔


“Agile”, insb. Scrum
Softwareexperten
Fachexperten
DDD

Domain Driven Design heute
(2000s-10s)

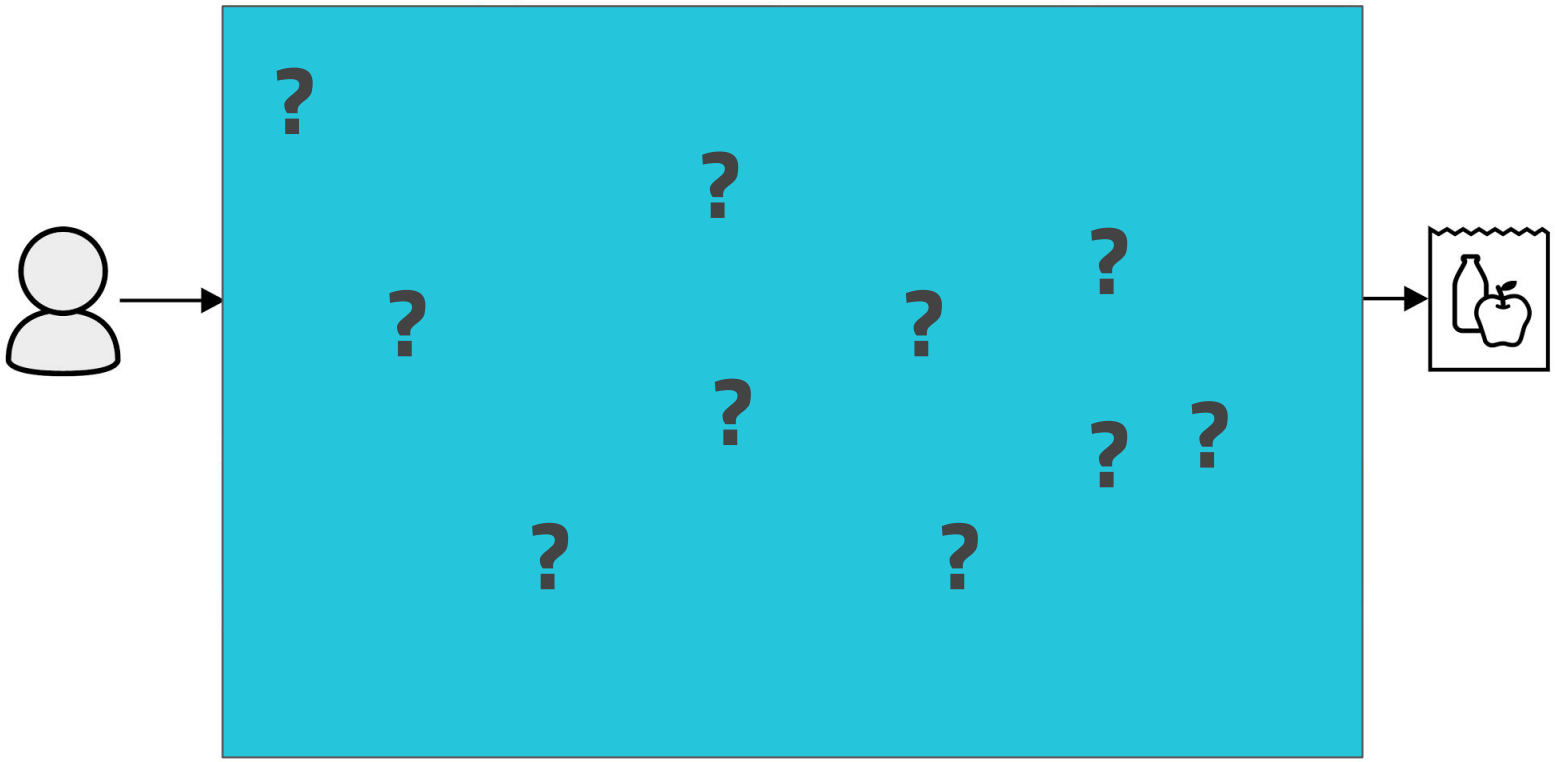

“Agile”
Softwareexperten
Fachexperten
DDD
?

Gründe für die DDD Renaissance

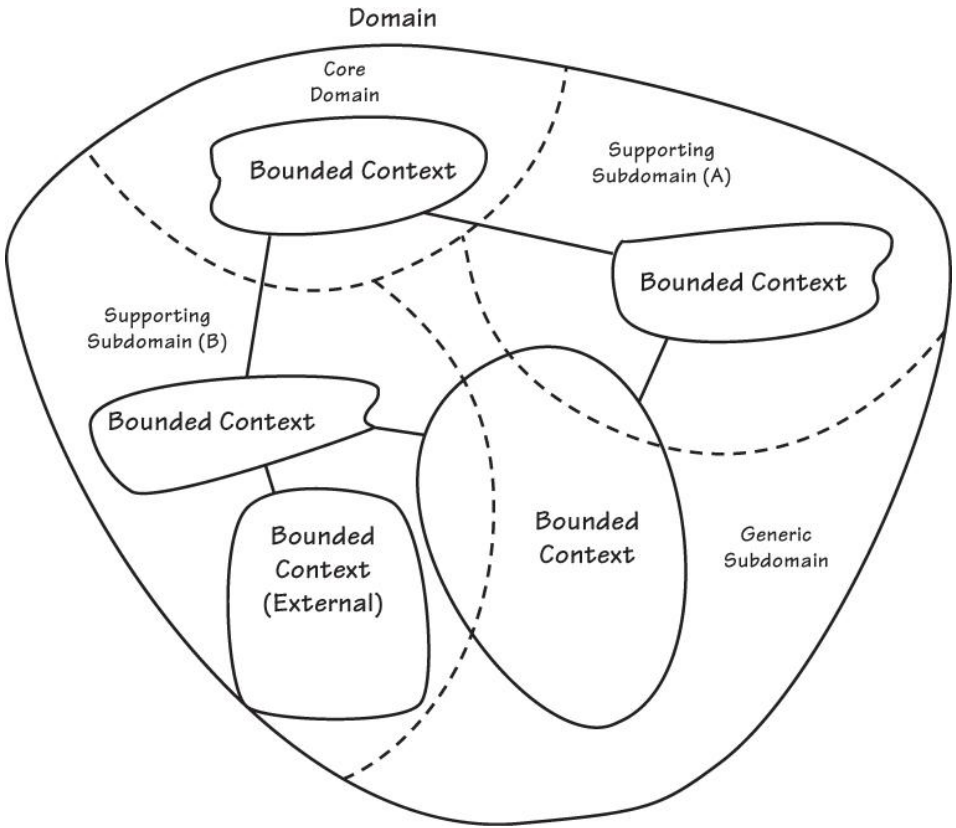
- **Bounded Context**
- Domain Event

Domain @ REWE

Subdomains @ REWE



Subdomains und Bounded Contexts



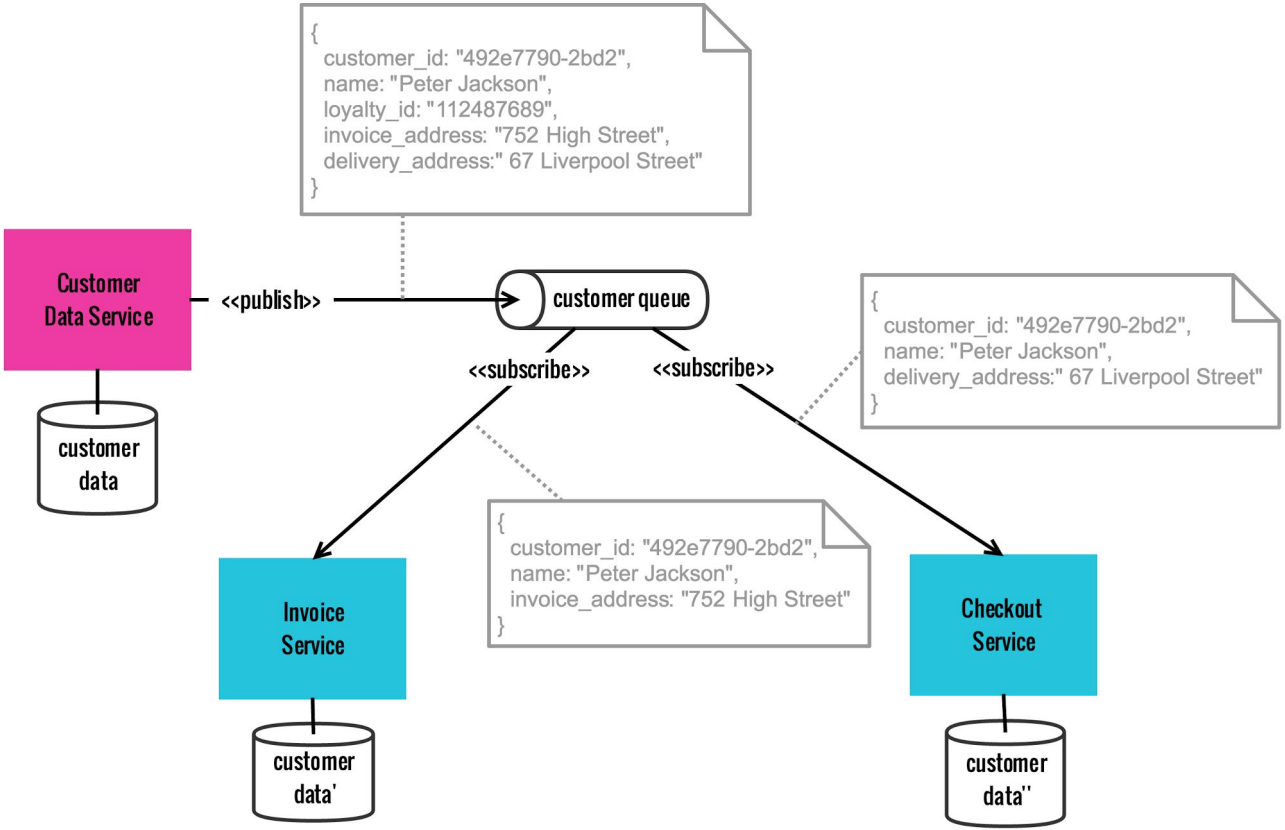
- **Subdomain = Problemraum**
- **Bounded Context = Lösungsraum**

Pro **Bounded Context**

- Anwendung eines Domain Models
- Eigene Ubiquitous Language

Im **Idealfall Überlappung** von Subdomains und Bounded Contexts

Übersetzen in andere Bounded Contexts



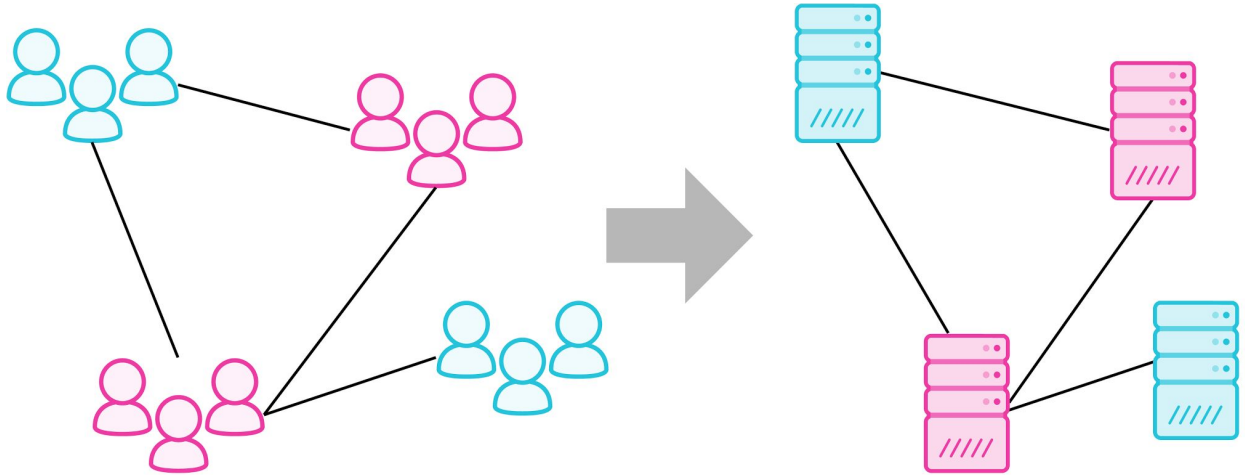
Gründe für die DDD Renaissance

- Bounded Context ✔
  - **Conway’s Law**
  - Self Contained Systems
- Domain Event

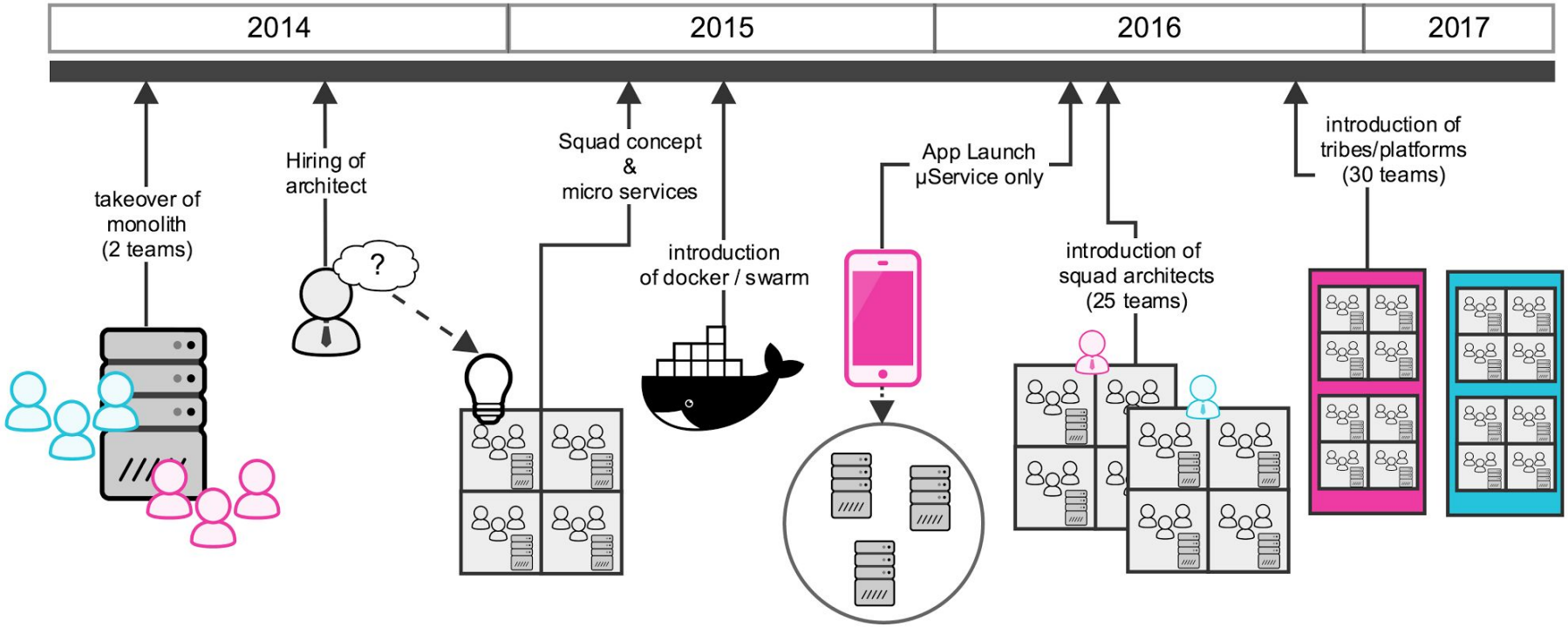
Conway's law

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

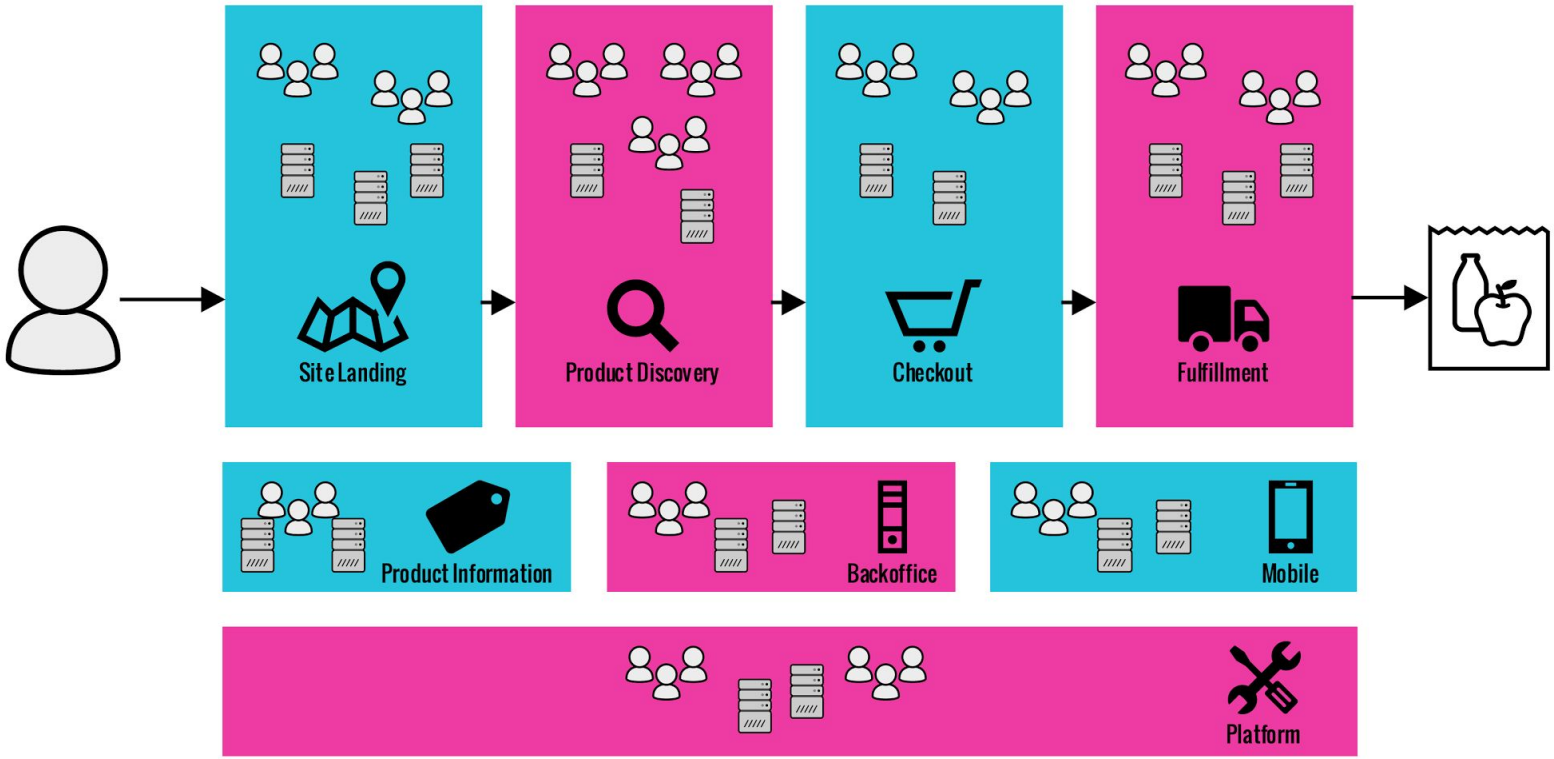
Melvin Conway (1967)



Conway's law @ REWE



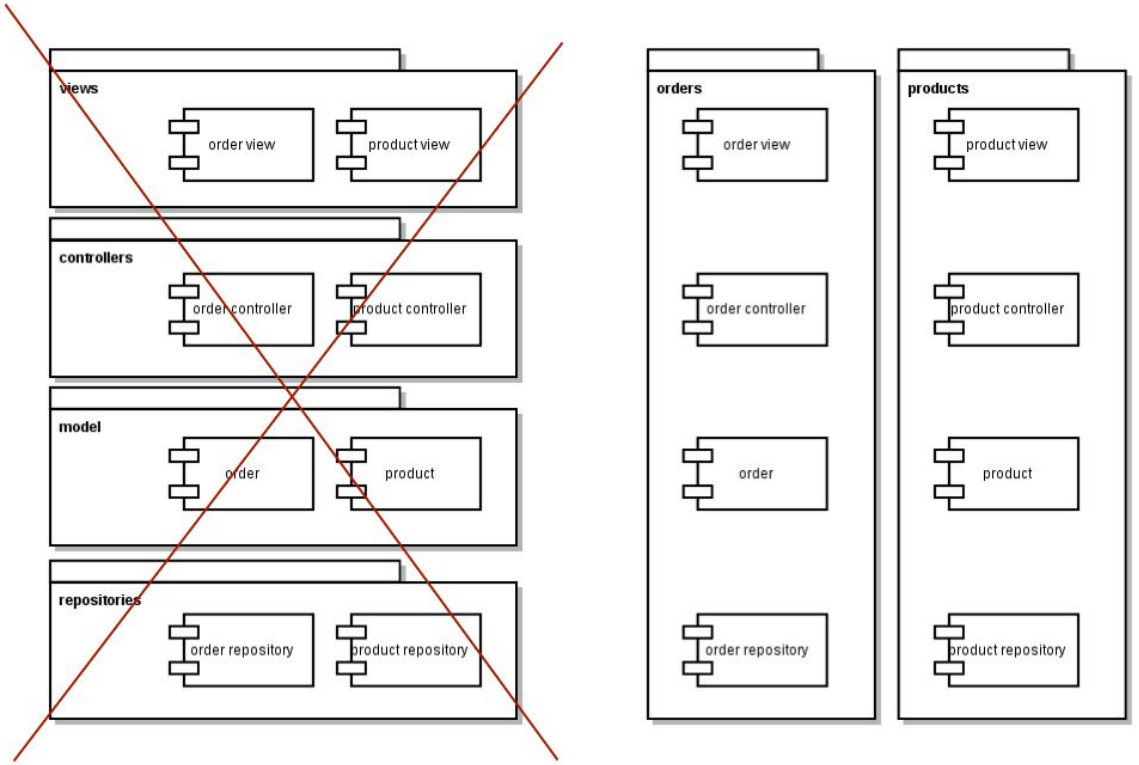
Squads/Subdomains @ REWE



Gründe für die DDD Renaissance

- Bounded Context ✔
  - Conway's Law ✔
  - **Self Contained Systems**
- Domain Event

Self-contained systems (SCS)

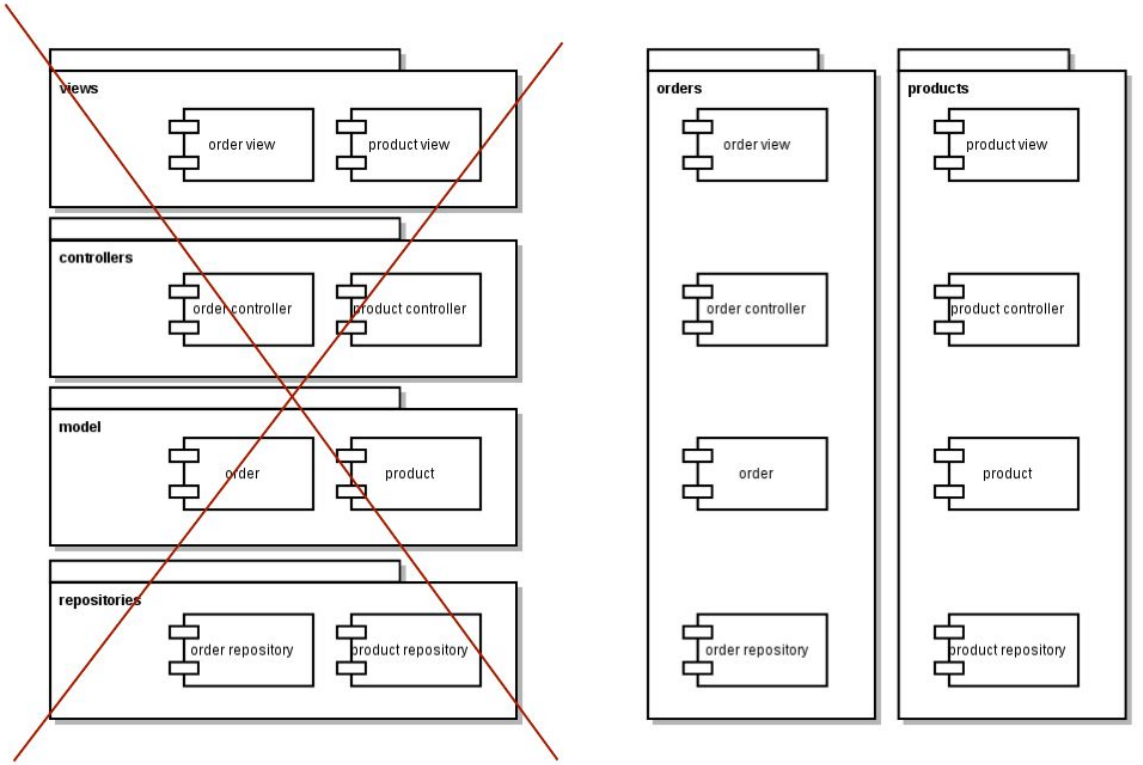


- SCS sind eine Spielart von **µServices**, zugeschnitten auf ein bestimmtes Szenario

- **1 SCS** wird betreut von **1 Team**
- **1 Team** betreut **n SCS**
- **1 Bounded Context** = **1 SCS**

- **Keine geteilte Fachlogik!**

Self-contained systems (SCS)



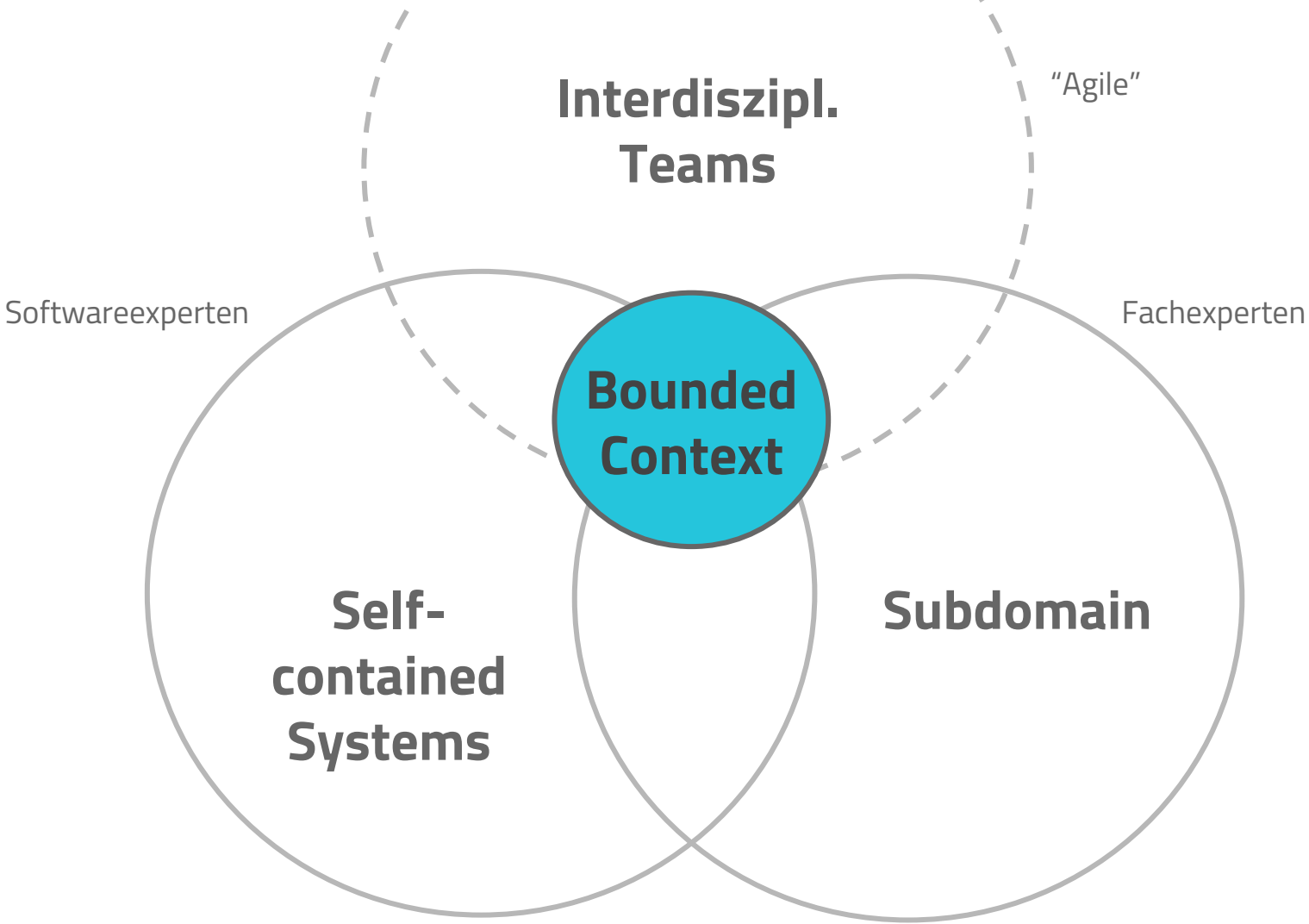
Abgrenzung zu µServices

- size(µService) < size(SCS)
- #µService > #SCS
- kommunizieren nicht synchron miteinander
- haben eine UI, Integration mehrerer SCS auf UI Ebene

http://scs-architecture.org/vs-ms.html

Gründe für die DDD Renaissance

- Bounded Context ✔
  - Conway’s Law ✔
  - Self Contained Systems ✔
- Domain Event

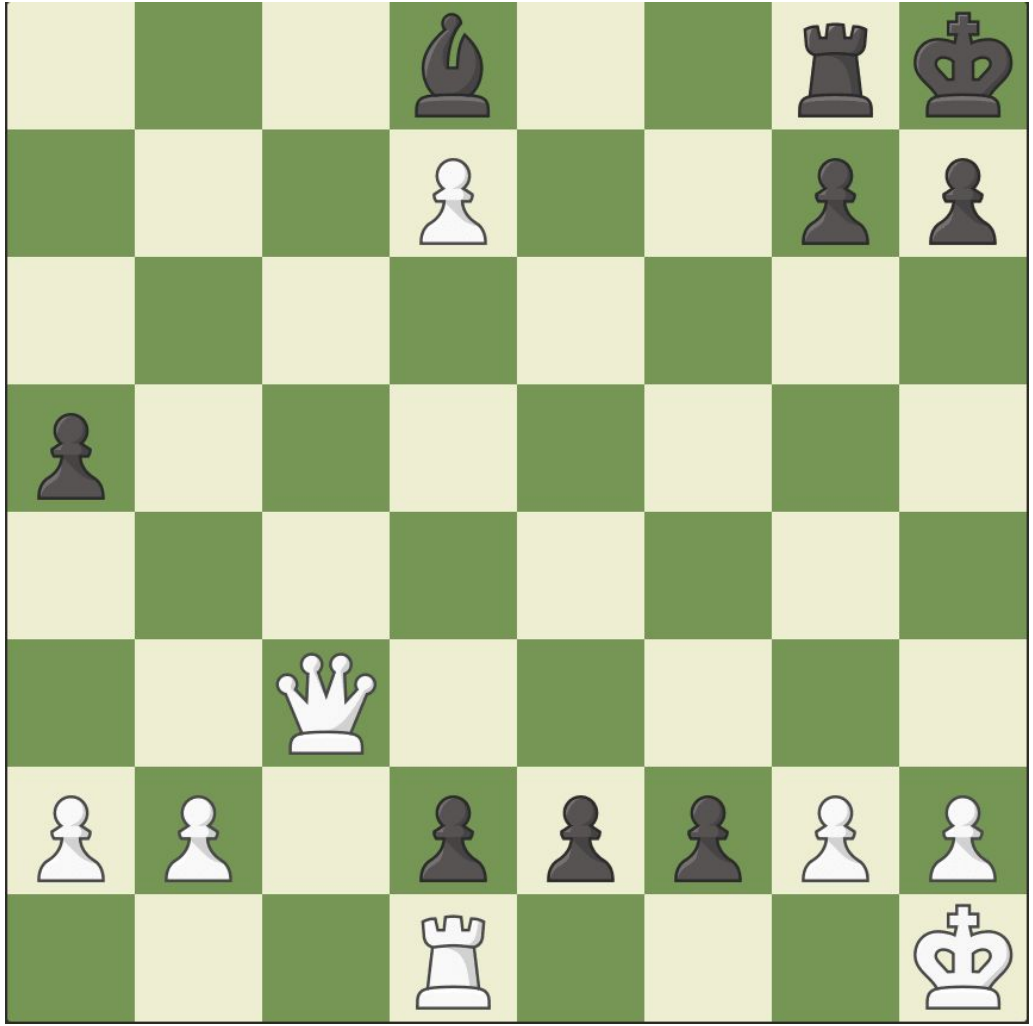

Interdiszipl.
Teams
“Agile”
Softwareexperten
Fachexperten
Bounded
Context
Self-
contained
Systems
Subdomain

Gründe für die DDD Renaissance

- Bounded Context ✔
- **Domain Event**
  - Event Sourcing
  - CQRS
  - Event Storming

Domain Event

- Ein **unveränderbarer Fakt**
- Liegt in der **Vergangenheit**
- Ist für **Domain Experten** von Interesse



Gründe für die DDD Renaissance

- Bounded Context ✔
- Domain Event ✔
  - **Event Sourcing**
  - CQRS
  - Event Storming

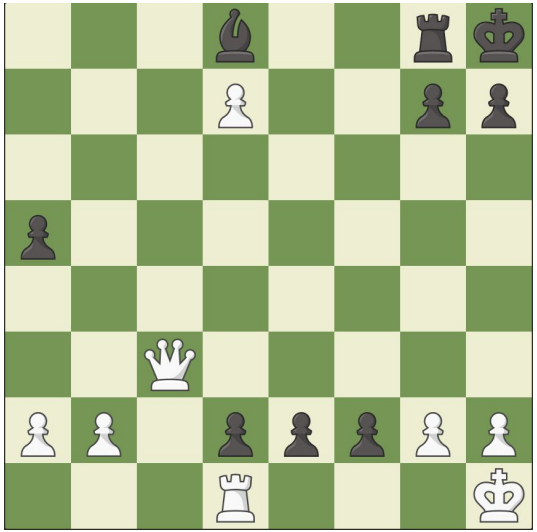
Domain Modellierung

Abstraktion: Domain ⇒ Modell

- Partie
  - **Stellung (Default: Grundstellung)**
  - Spieler { Weiß, Schwarz }
  - { Spieler } ist am Zug (Default: Weiß)

Klassisches CRUD

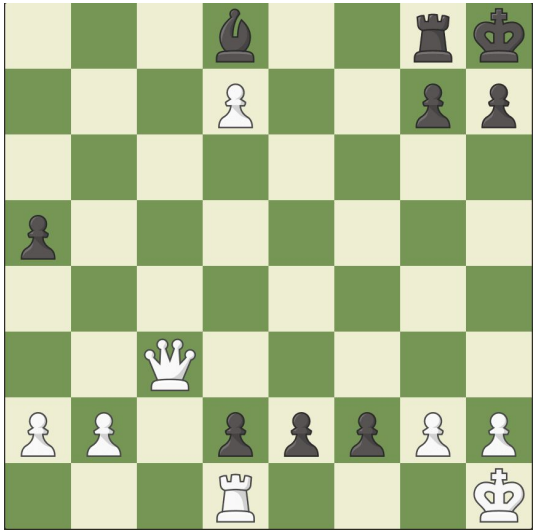
- Bei Änderungen wird die jeweils **aktuelle Stellung** gespeichert
- Modellierung z.B. in Forsyth-Edwards-Notation (FEN)



3b2rk/3P2pp/8/p7/8/2Q5/PP1pppPP/3R3K w - - 0 38

Event Sourcing

- Aktuelle Stellung wird aus der **Historie aller Domain Events** erzeugt
- Modellierung z.B. in Portable Game Notation (PGN)



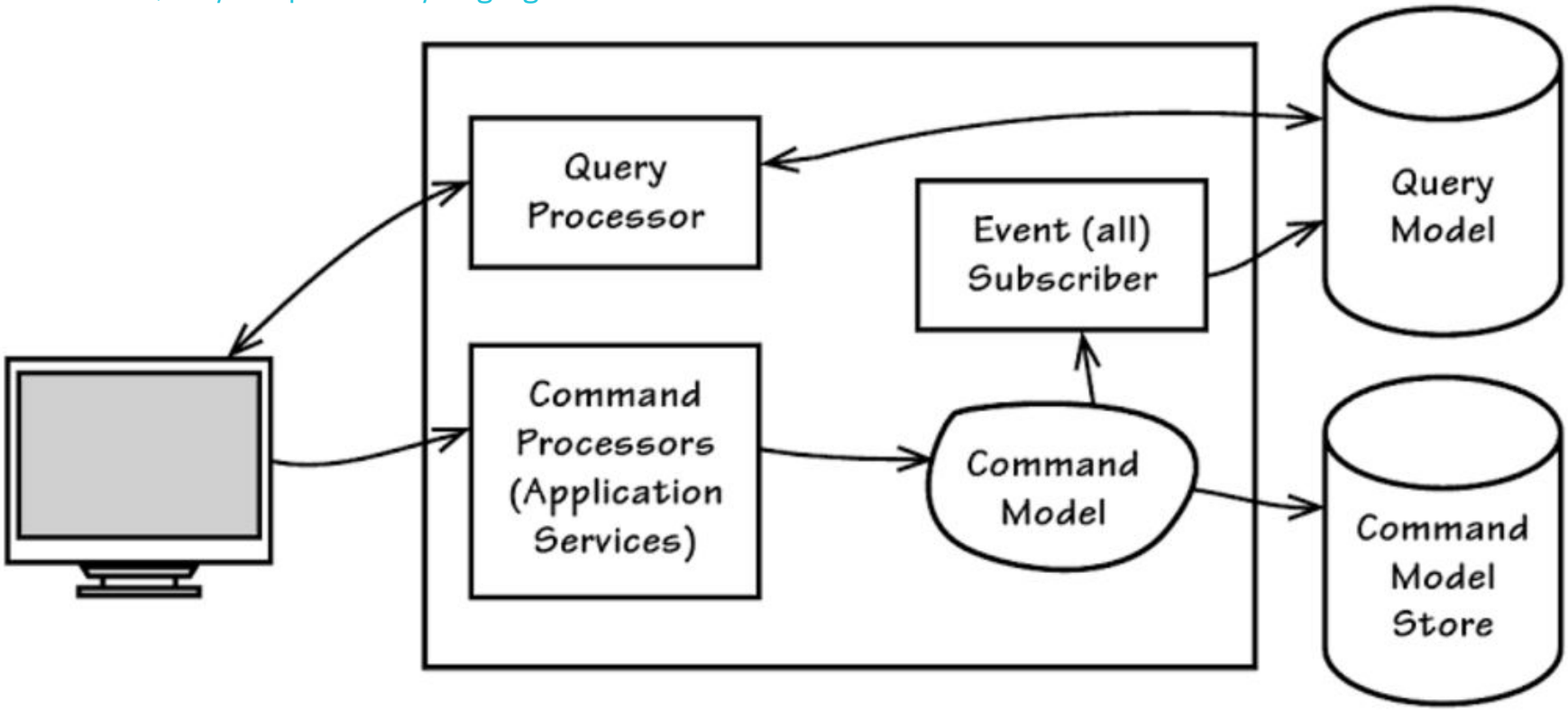
1.e4 c5 2.Nf3 Nc6 3.d4 cxd4 4.Nxd4 e5 5.Nxc6 bxc6 6.Bc4 Nf6 7.Bg5 Be7 8.Qe2 d5 9.Bxf6 Bxf6 10.Bb3 O-O 11.O-O a5 12.exd5 cxd5 13.Rd1 d4 14.c4 Qb6 15.Bc2 Bb7 16.Nd2 Rae8 17.Ne4 Bd8 18.c5 Qc6 19.f3 Be7 20.Rac1 f5 21.Qc4+ Kh8 22.Ba4 Qh6 23.Bxe8 fxe4 24.c6 exf3 25.Rc2 Qe3+ 26.Kh1 Bc8 27.Bd7 f2 28.Rf1 d3 29.Rc3 Bxd7 30.cxd7 e4 31.Qc8 Bd8 32.Qc4 Qe1 33.Rc1 d2 34.Qc5 Rg8 35.Rd1 e3 36.Qc3 Qxd1 37.Rxd1 e2

Gründe für die DDD Renaissance

- Bounded Context ✔
- Domain Event ✔
  - Event Sourcing ✔
  - **CQRS**
  - Event Storming

CQRS

Command **Q**uery **R**esponsibility **S**egregation

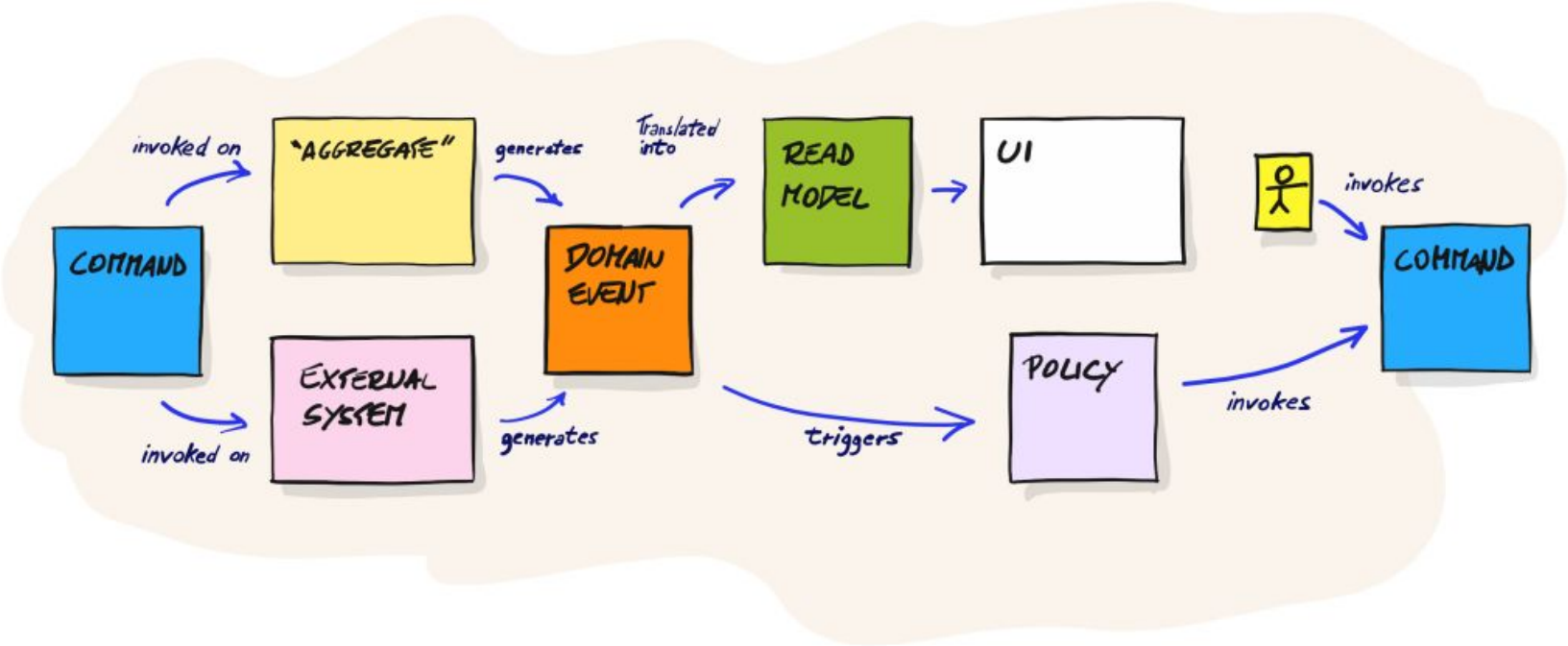


Gründe für die DDD Renaissance

- Bounded Context ✔
- Domain Event ✔
  - Event Sourcing ✔
  - CQRS ✔
  - **Event Storming**

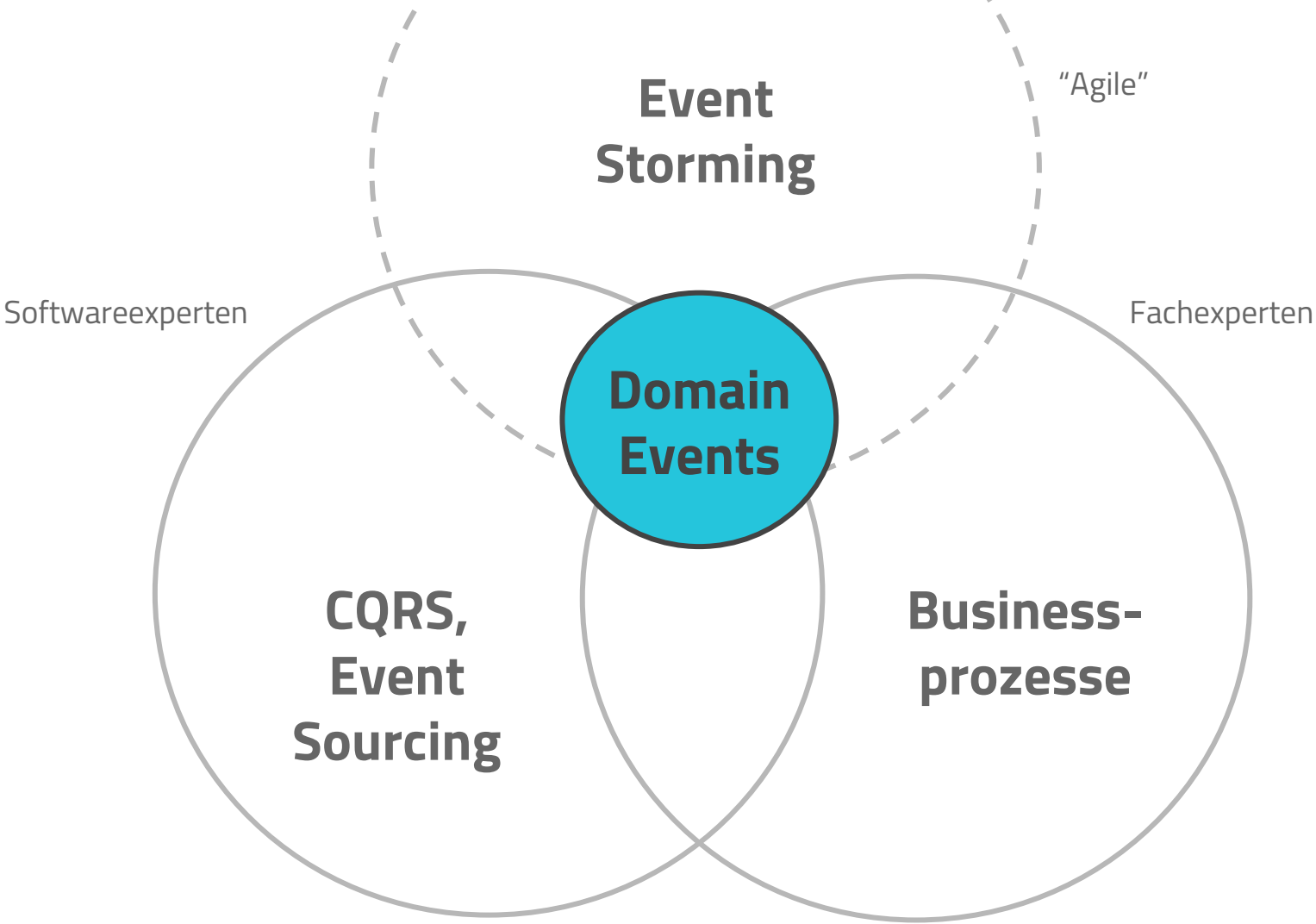
Event Storming

Kollaboratives Erforschen und Modellieren von Businessprozessen



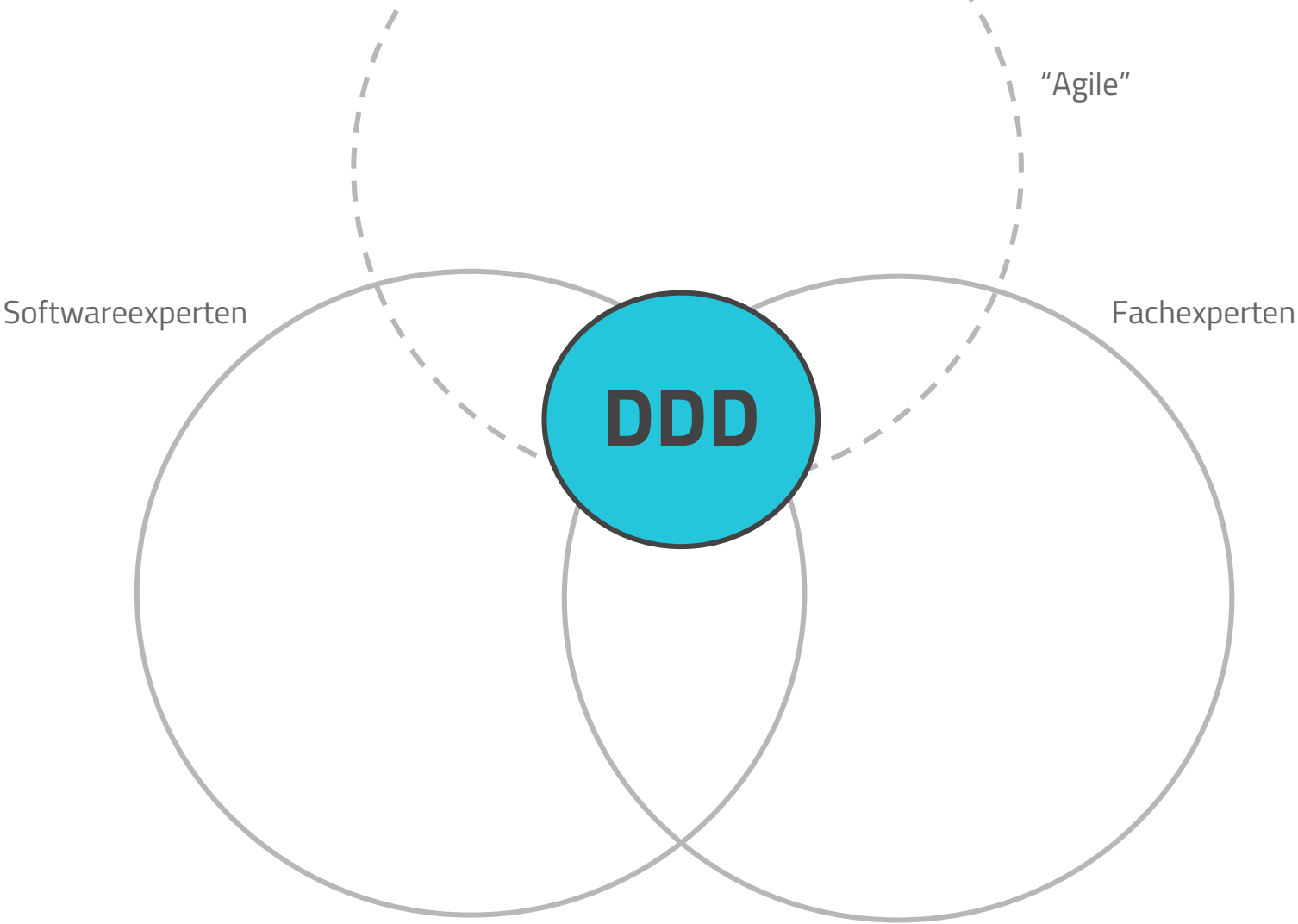
Gründe für die DDD Renaissance

- Bounded Context ✔
- Domain Event ✔
  - Event Sourcing ✔
  - CQRS ✔
  - Event Storming ✔

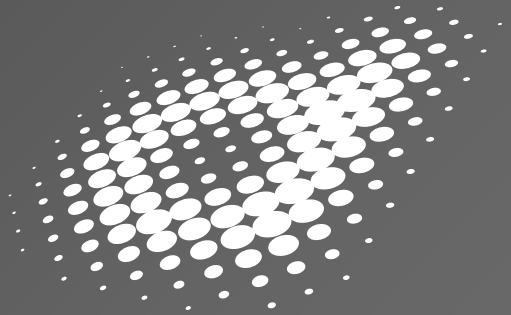

Event Storming
“Agile”
Softwareexperten
Fachexperten
Domain Events
CQRS, Event Sourcing
Business-prozesse

Gründe für die DDD Renaissance

- **Bounded Context** ✔
  - Conway's Law ✔
  - Self Contained Systems ✔
- **Domain Event** ✔
  - Event Sourcing ✔
  - CQRS ✔
  - Event Storming ✔


“Agile”
Softwareexperten
Fachexperten
DDD

Interesse an Domain Driven Design?



REWE digital

https://rewe-digital.com/jobs.html

Vielen DDDank :)

Christoph Baudson / @sustainablepace