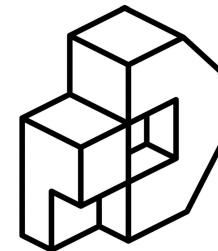


# EventStorming

Uma abordagem  
sociotécnica para  
mapeamentos de  
Domínios



**DEVPIRA**

**Erik Aceiro Antonio**

# aboutMe ()



Formado em Ciência da Computação (MACKENZIE)

- Componente adaptativo e colaborativo

Mestre em Engenharia Elétrica e Óptica & Fotônica (MACKENZIE)

- WebLab & LabView

Doutor em Engenharia de Software (UFSCar)

- Atividades de VV&T para Sistemas Embarcados UML/SysML

Arquiteto de Soluções (Raízen & Shell Box)

Professor, Pesquisador, Amante de Gatos & Vegetariano :)

# howToFindMe()



<https://www.linkedin.com/in/erik-antonio-aa018925>



<https://github.com/aceiro>



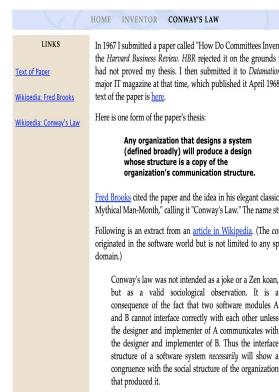
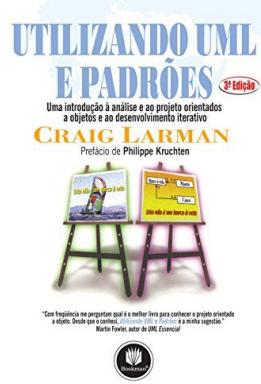
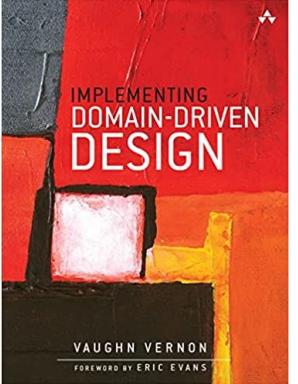
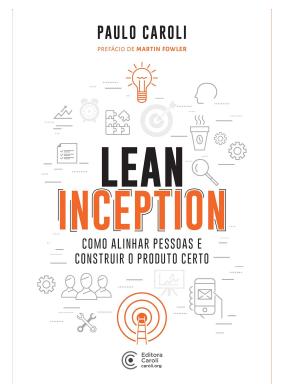
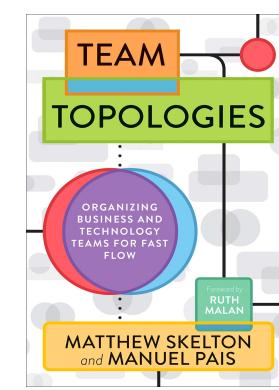
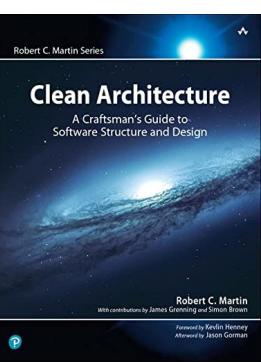
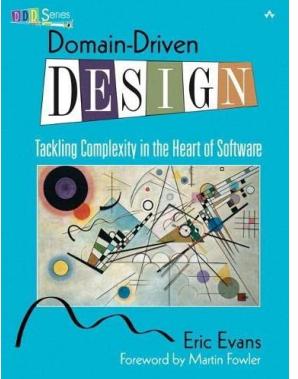
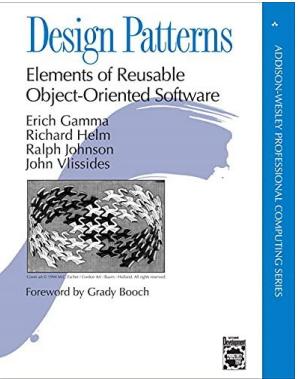
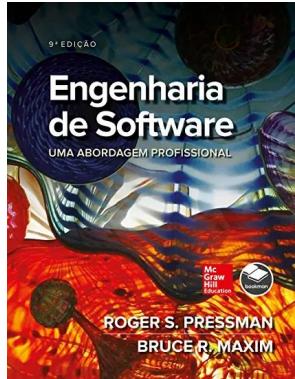
<https://medium.com/ereflections>



aceiro@gmail.com

# Agenda

- Desafios na Engenharia de Software
- Domínios, Contexto Delimitados
- Topologia de Times
- Lei de Conway
- Arquitetura Sociotécnica
- EventStorming



Conway's law was not intended as a joke or a Zen koan, but as a valid sociological observation. It is a consequence of the fact that two software modules A and B cannot interface correctly with each other unless the designer and implementer of A communicates with the designer and implementer of B. Thus the interface structure of a software system necessarily will show a congruence with the social structure of the organization that produced it.

In 1967 I submitted a paper called "How Do Committees Invent?" to the *Harvard Business Review*. *HBR* rejected it on the grounds that I had not proved my thesis. I then submitted it to *Datamation*, the major IT magazine at that time, which published it April 1968. The text of the paper is [here](#).

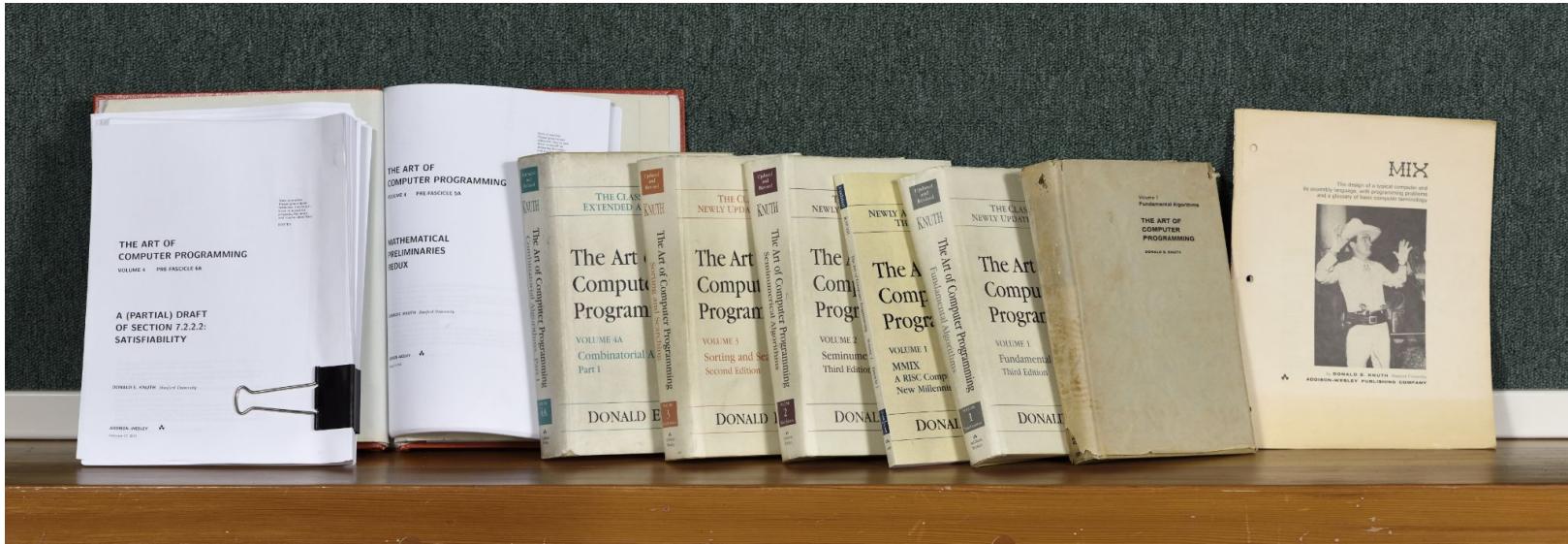
Here is one form of the paper's thesis:

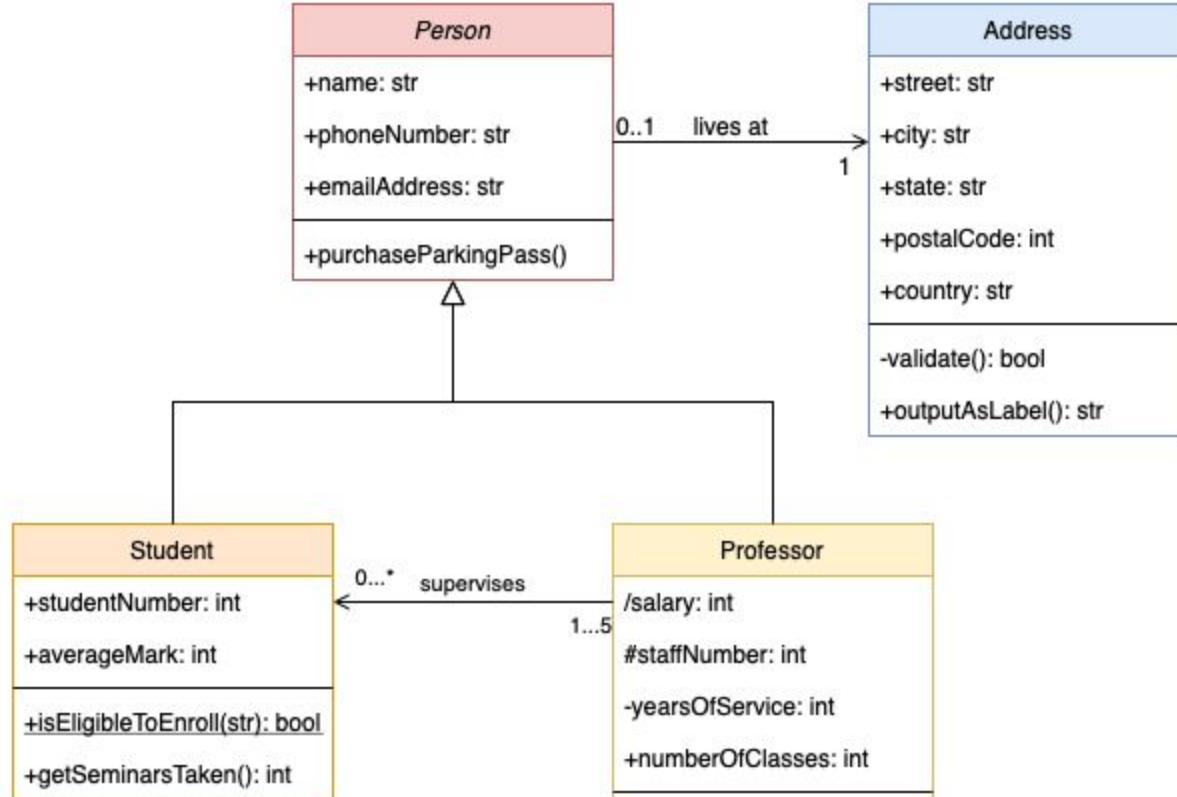
Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.

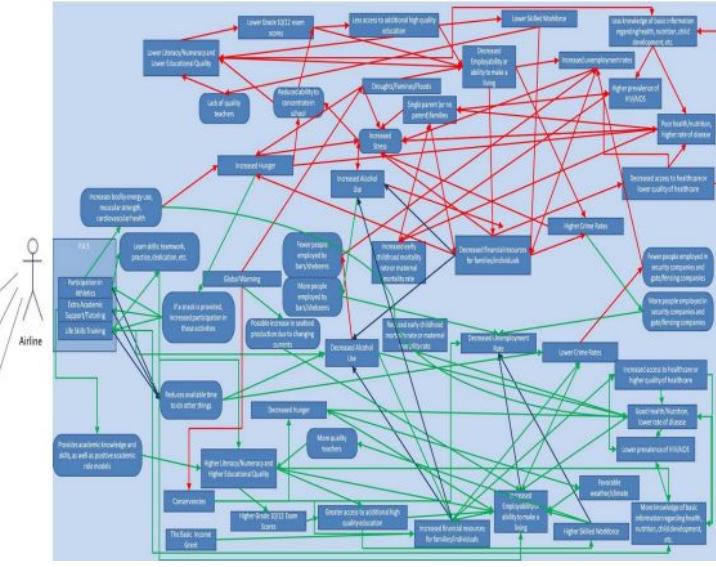
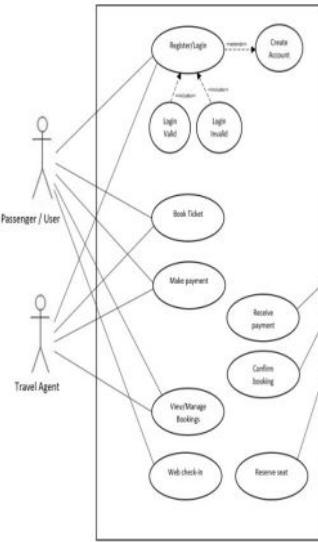
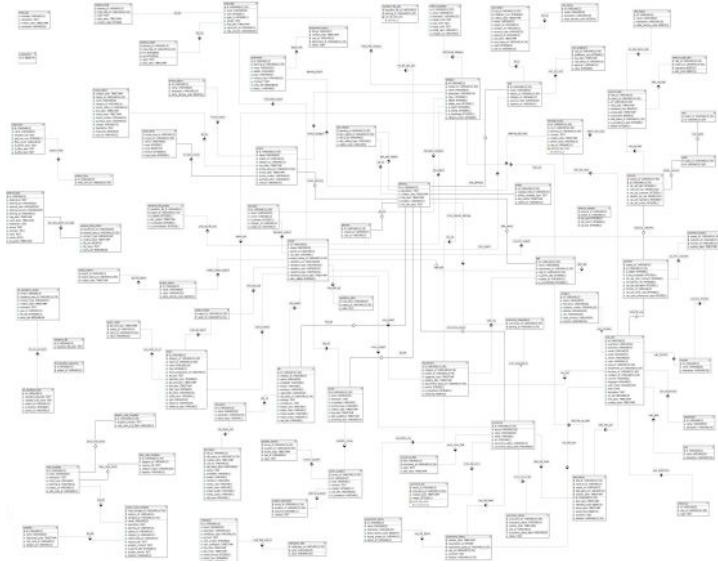
Fred Brooks cited the paper and the idea in his elegant classic "The Mythical Man-Month," calling it "Conway's Law." The name stuck.

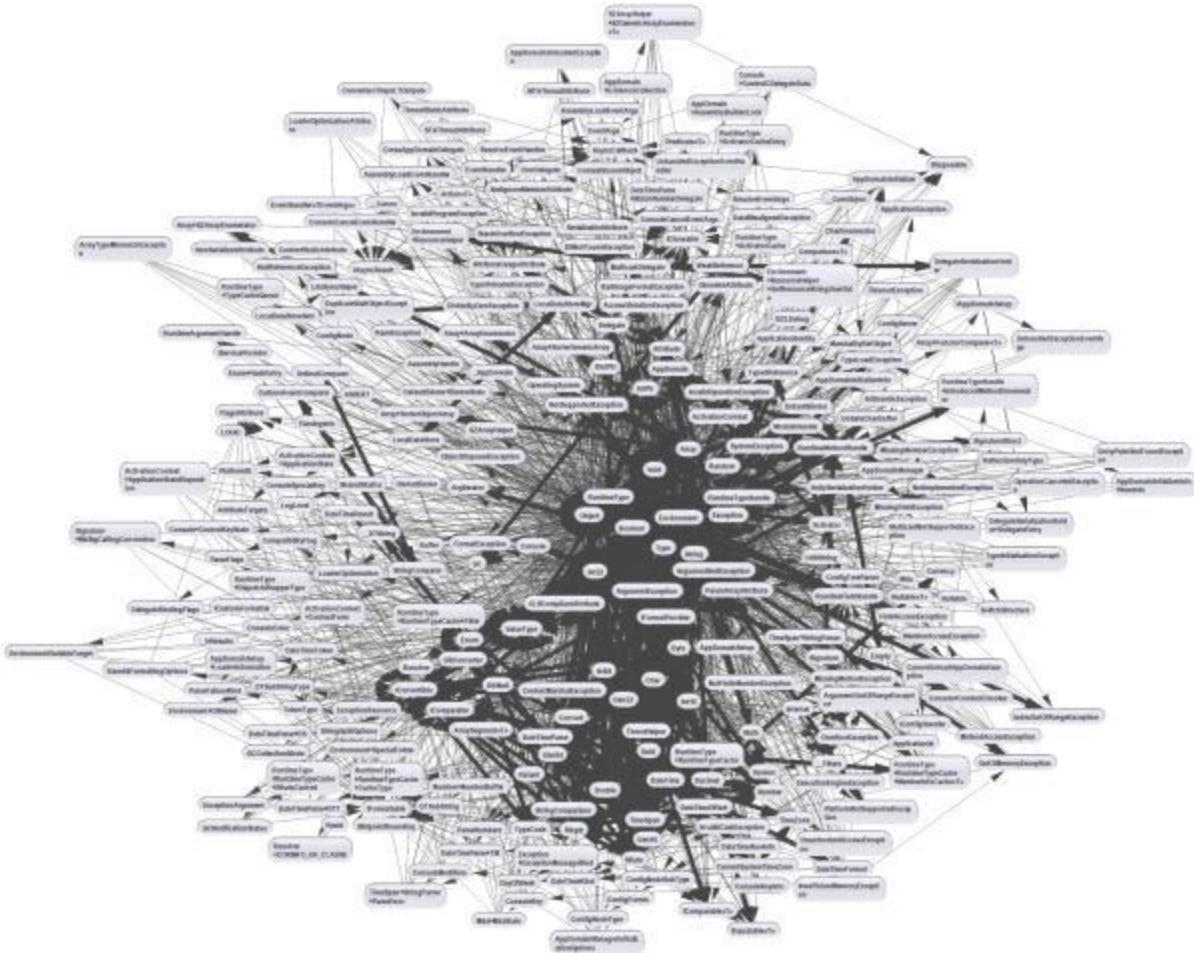
Following is an extract from an [article in Wikipedia](#). (The concept originated in the software world but is not limited to any specific domain.)

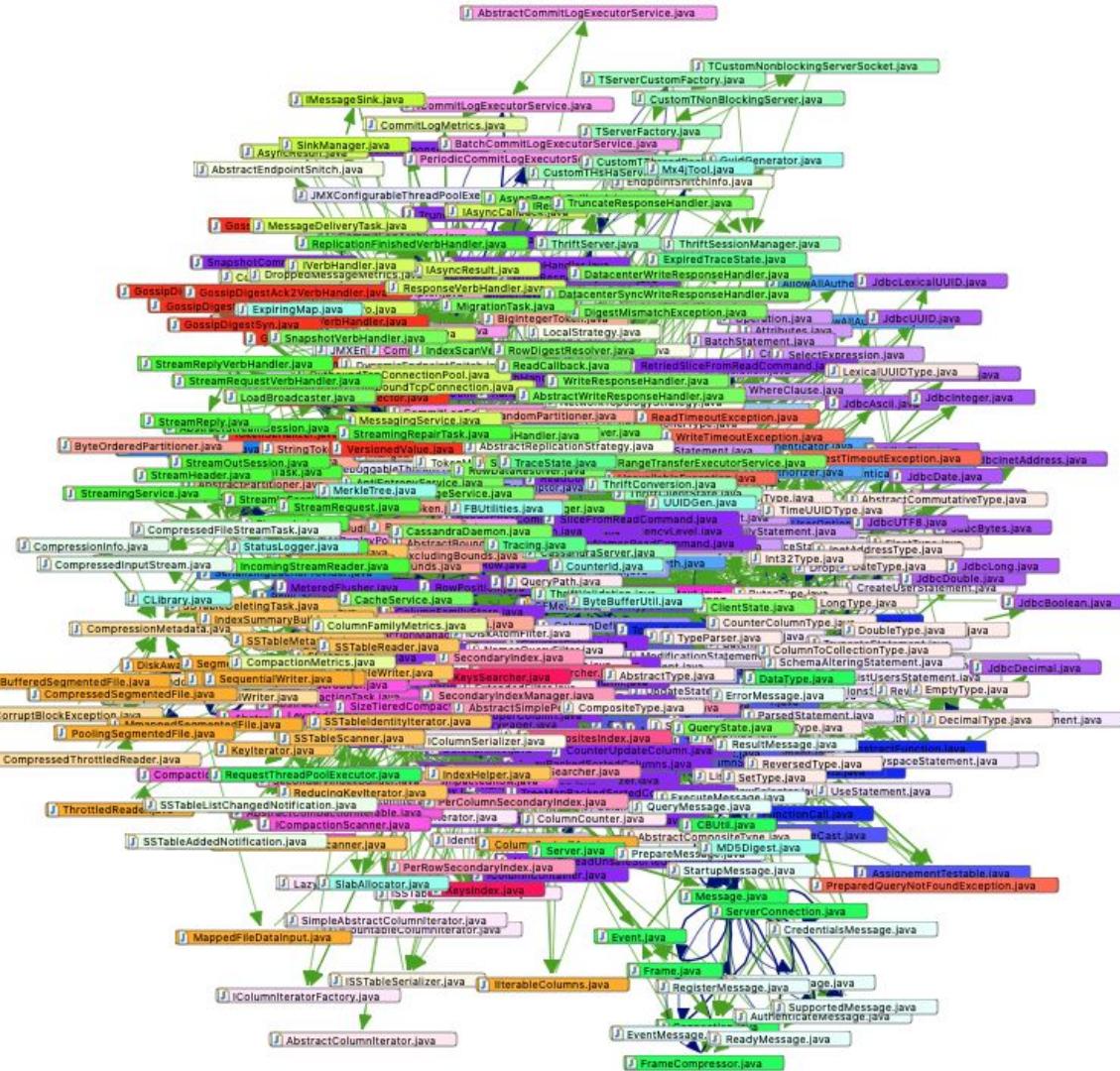
# The art of build software (Donald Knuth)



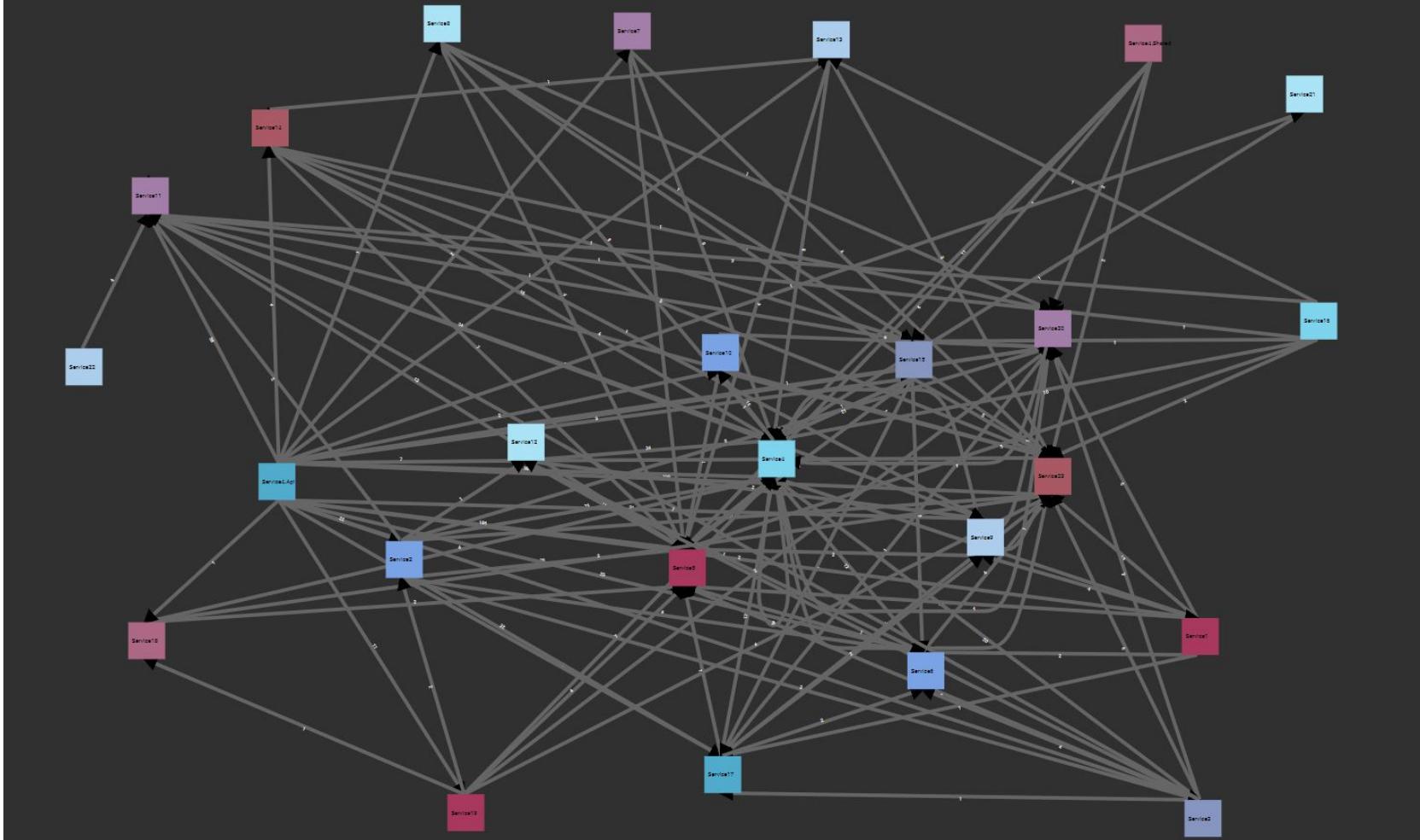










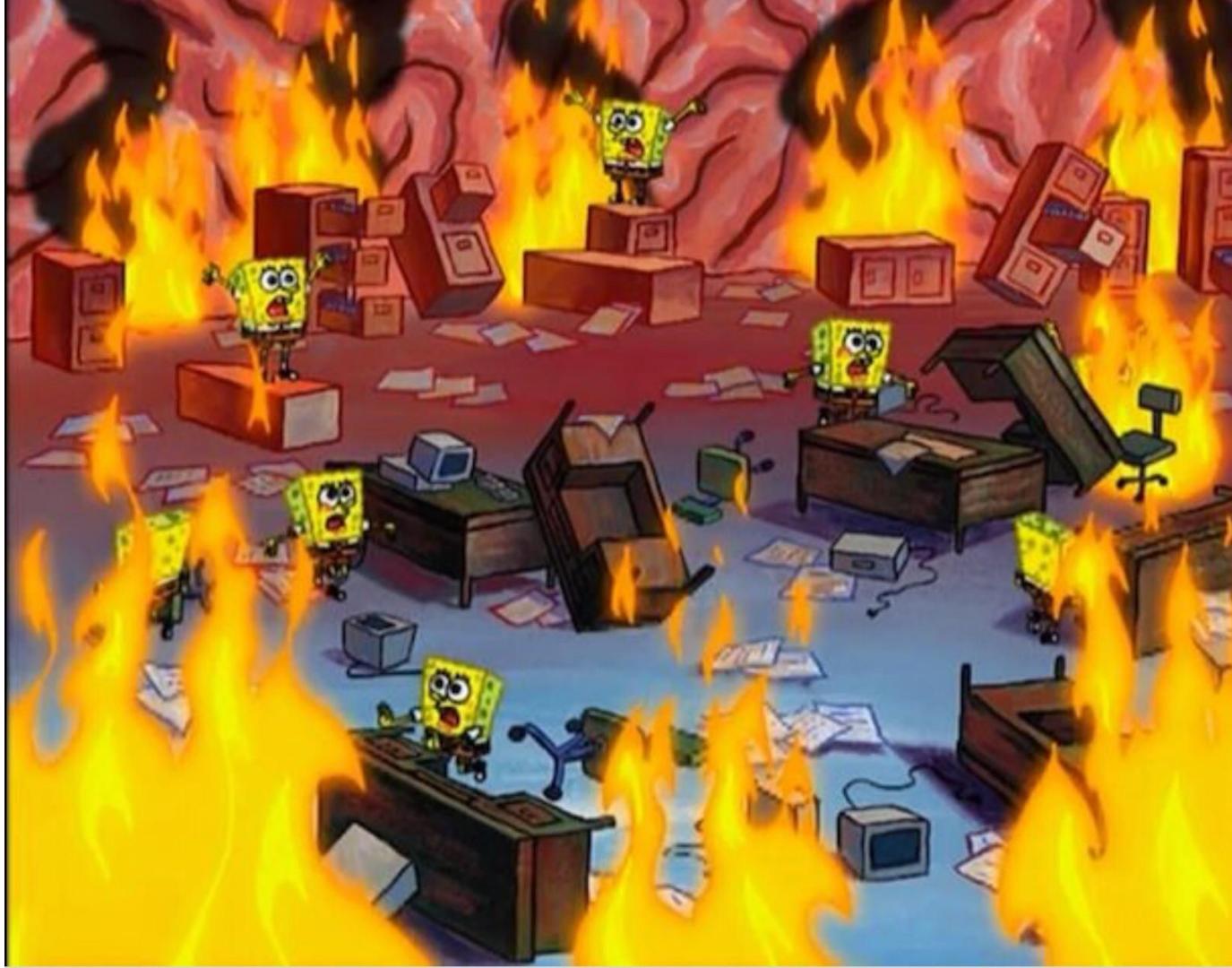


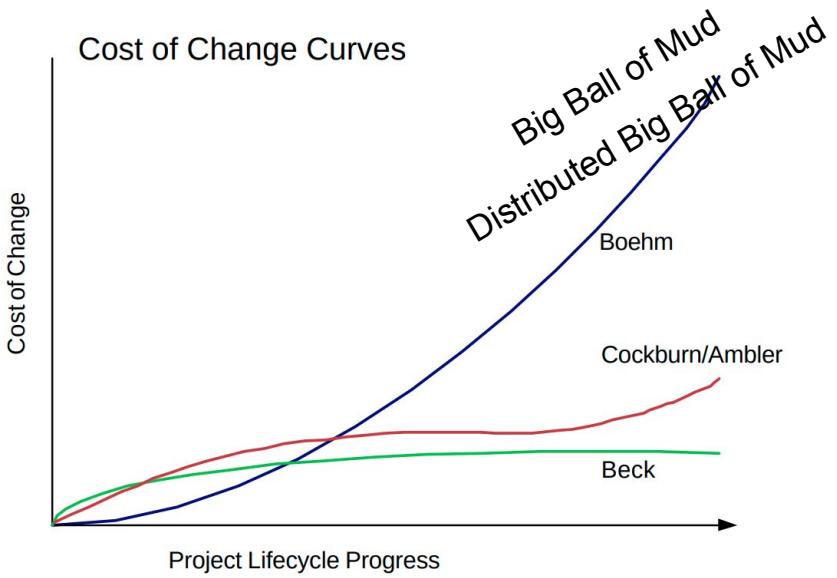
<https://softwarearchitectures.io/designing-and-validating-microservices-boundaries/>

A photograph of a person lying in a large pile of brown mud. The person is laughing heartily, with their head tilted back and mouth wide open. The mud is everywhere, covering the person's face, hair, and body. The background is a dark, textured surface of more mud.

# Big Ball of Mud

The Daily Software Anti-Pattern / Exception Not Found



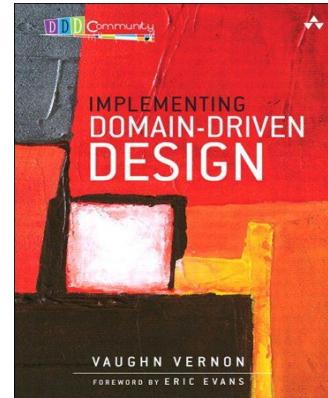
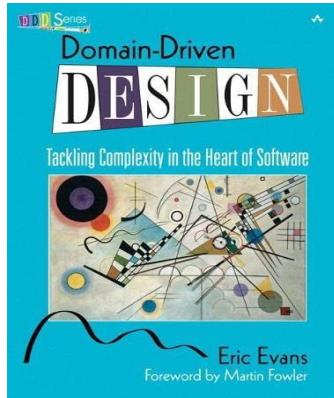


SZALVAY, Victor. An introduction to  
agile software development.  
**Danube technologies**, v. 3, 2004.



**Manifesto Ágil**

# Domínios, Contexto Delimitados & DDD





Domínio,  
o que é... eu digo?

Em DDD um **Core Domain**  
é um domínio que entrega  
um valor altamente  
estratégico



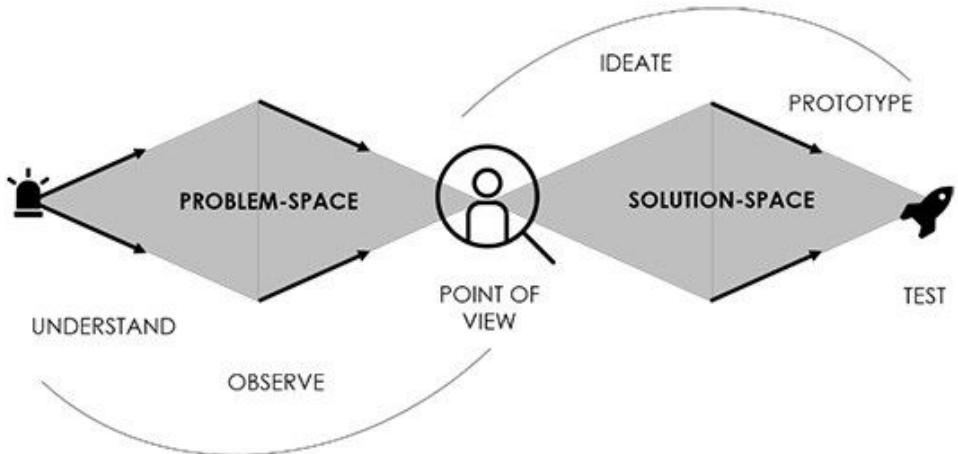
Eric Evans

**Domínio são como áreas  
onde várias equipes  
operam para agregar valor.**

Um negócio é uma coleção  
de verticais que são  
compostas por domínios



Evan Botther



# Espaço de Problema e Solução

# Design Patterns

Elements of Reusable  
Object-Oriented Software

Erich Gamma  
Richard Helm  
Ralph Johnson  
John Vlissides

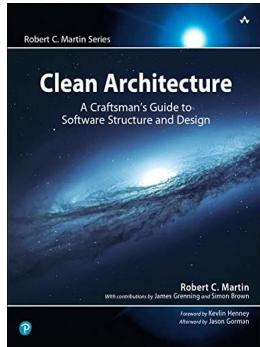


Foreword by Grady Booch

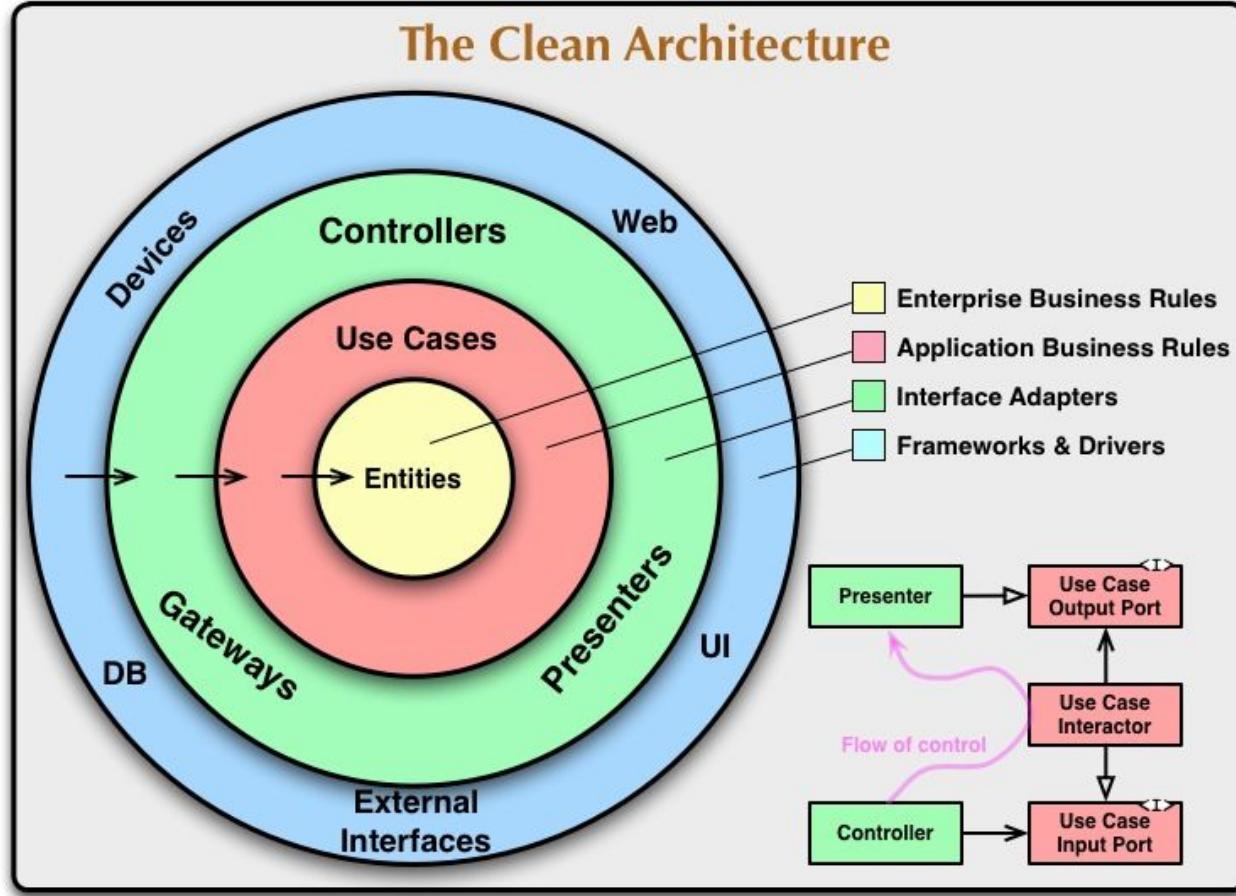
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

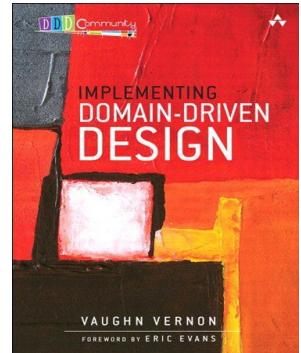
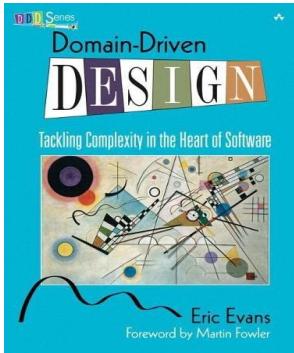
# GoF 1994

		Purpose		
		Creational	Structural	Behavioral
Scope	Class	Factory Method	Adapter	Interpreter Template Method
Object	Object	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Proxy	Chain of Responsibility Command Iterator Mediator Memento Flyweight Observer State Strategy Visitor



# Clean Arch 2013





# DDD Eric Evans 2015



## Tático =Solução



## Estratégico =Problema

## Tático =Solução



# Estratégico =Problema

# Estratégico & Problema



Perspectiva de Domínio

“Em relação ao Negócio reflete um campo na indústria que o negócio opera”

# Estratégico & Problema



Domínio  
Óleo & Gás



Domínio  
Abastecimento



Domínio  
Streaming



Domínio  
Alimentação

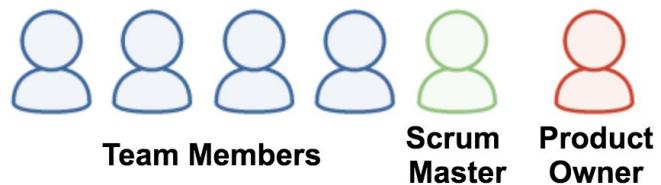
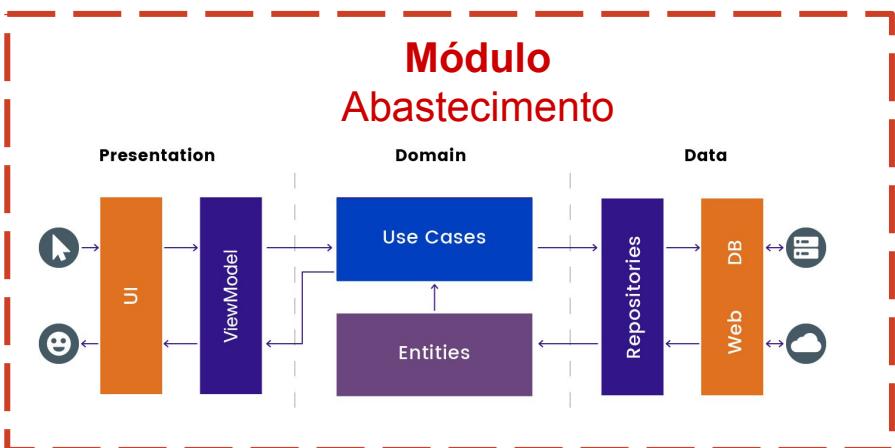


Domínio  
Transporte

Perspectiva de Domínio

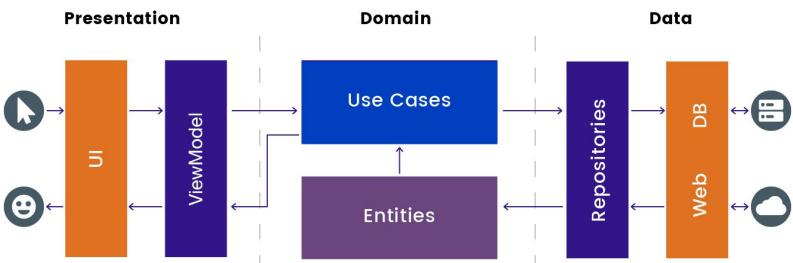
**“Em relação ao Negócio reflete um campo na indústria que o negócio opera”**

# Estratégico & Tático (DDD)



Tático

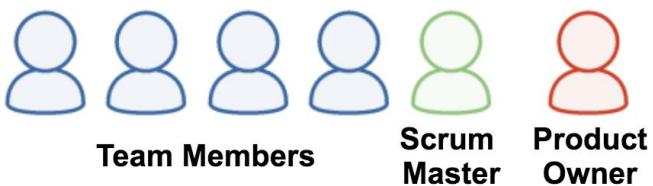
## Módulo Abastecimento



Estratégico



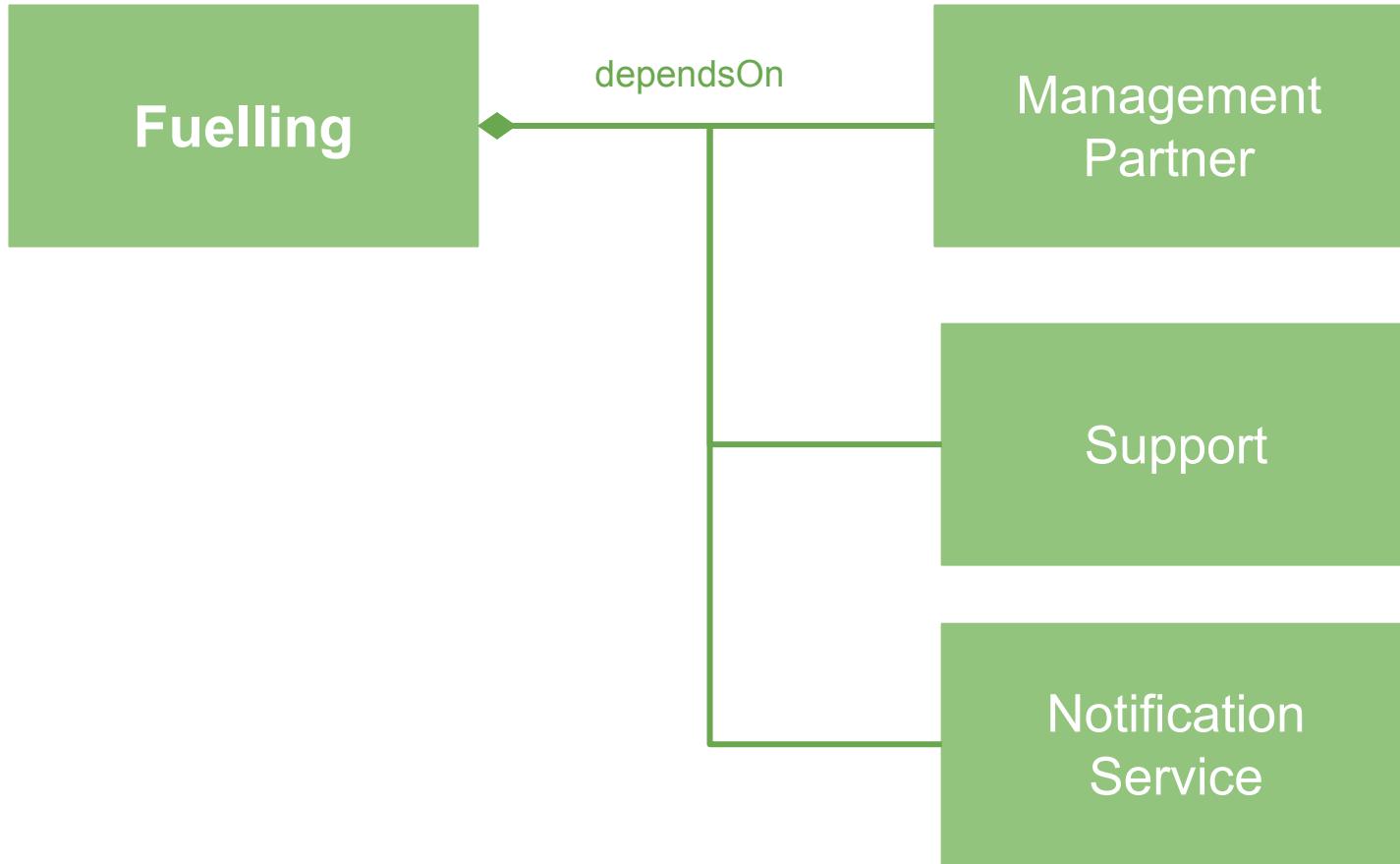
## Domínio Abastecimento

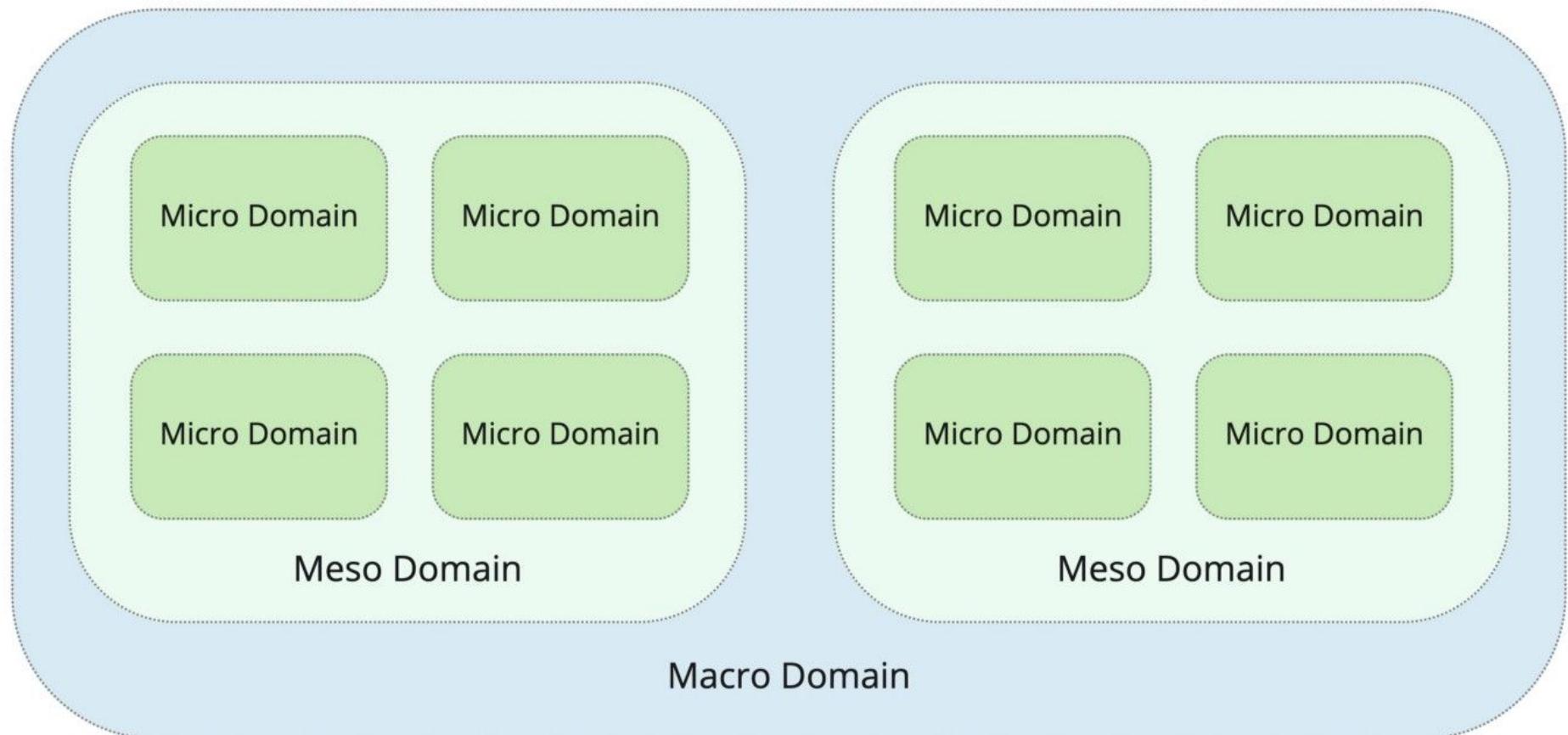


# Espaço de Problema = Estratégico

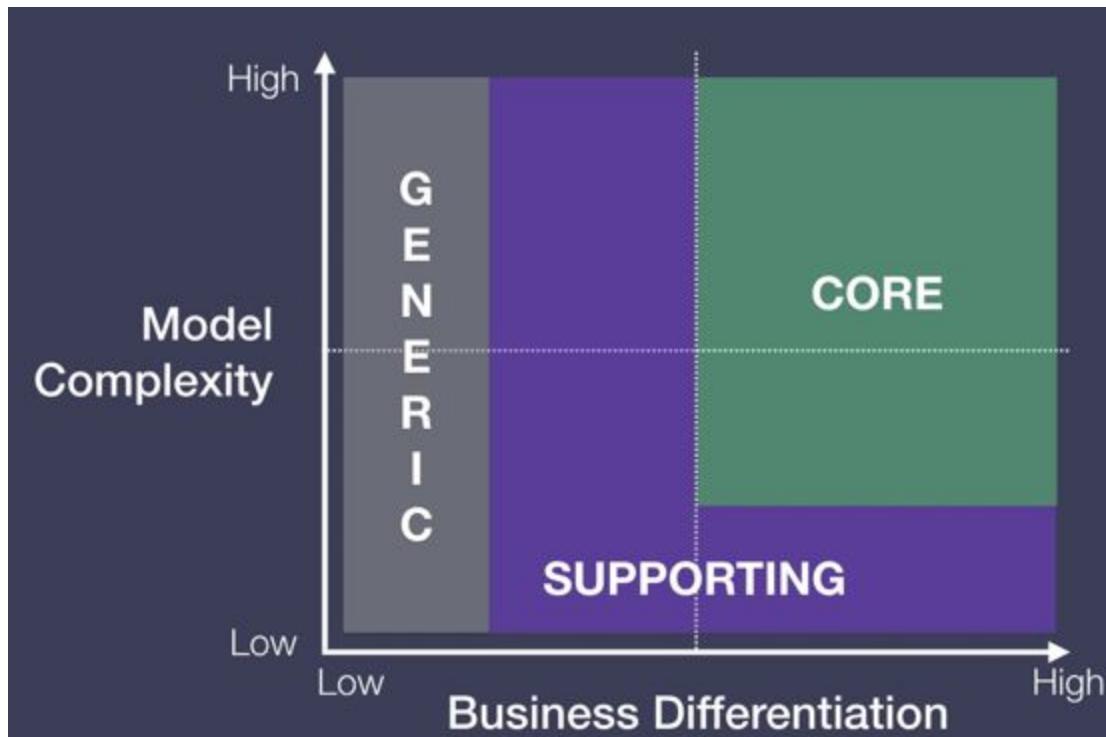
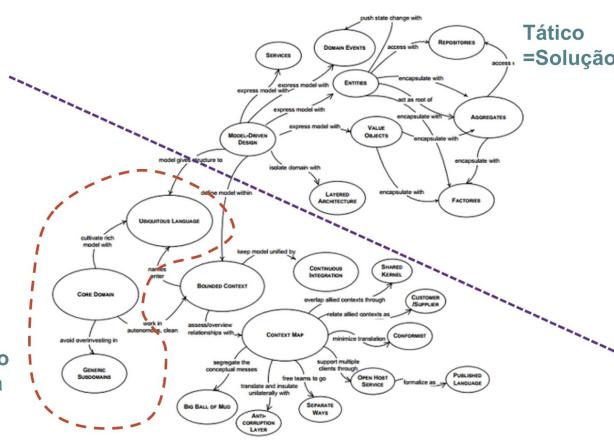
“Cada domínio pode conter N-subdomínios







Estratégico  
=Problema

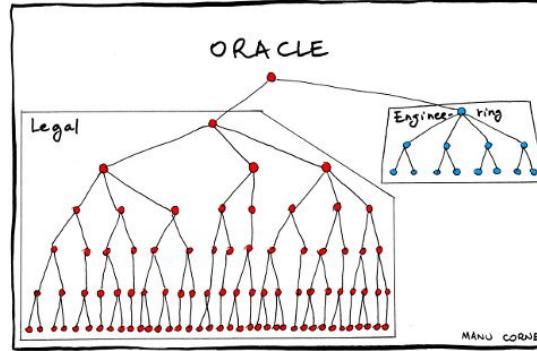
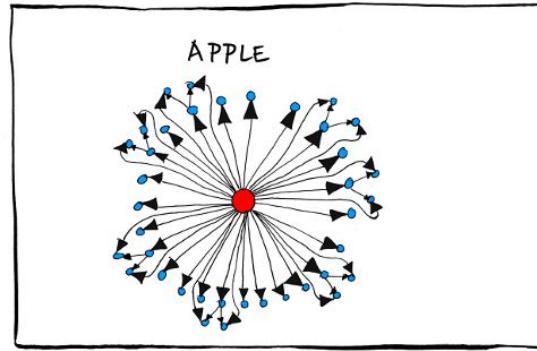
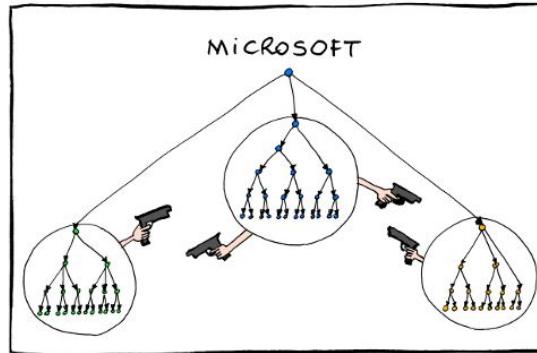
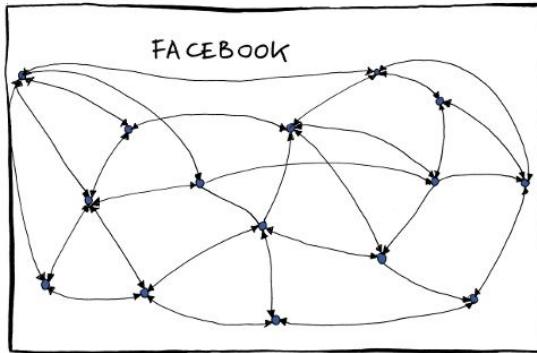
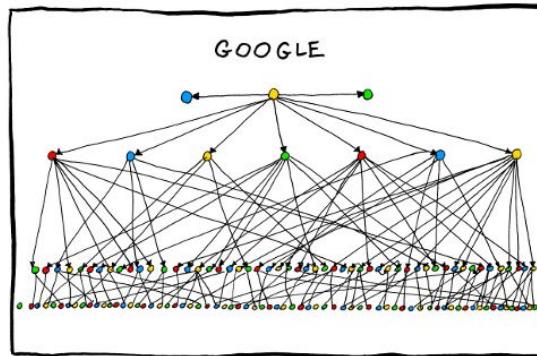
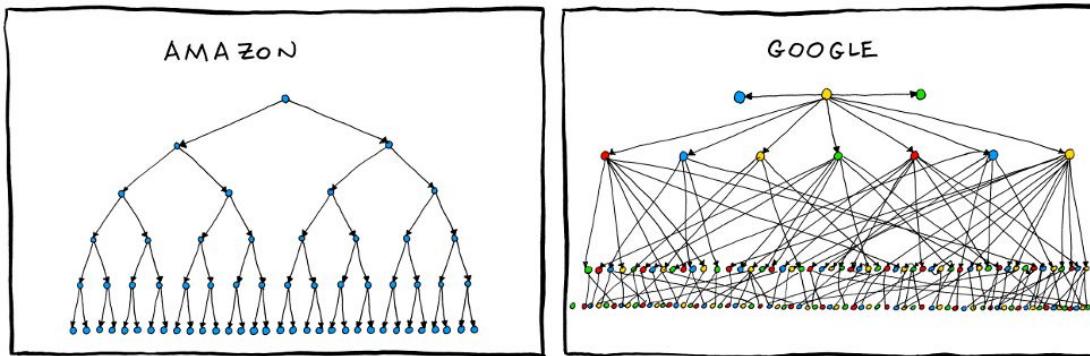


Tático  
=Solução

In 1967 I submitted a paper called "How Do Committees Invent?" to the *Harvard Business Review*. *HBR* rejected it on the grounds that I had not proved my thesis. I then submitted it to *Datamation*, the major IT magazine at that time, which published it April 1968. The text of the paper is [here](#).

Here is one form of the paper's thesis:

**Any organization that designs a system  
(defined broadly) will produce a design  
whose structure is a copy of the  
organization's communication structure.**



**Qualquer organização que projeta um sistema, produzirá um projeto cuja estrutura é uma cópia da estrutura de comunicação da organização**

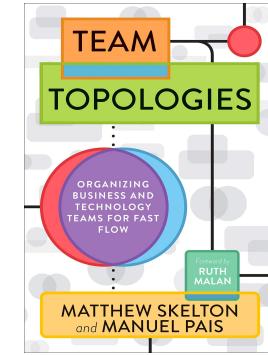
**Lei de Conway**

Organização de  
Pessoas

Comunicação



Lei de Conway



Estrutura de  
Software  
(aka Arquitetura)

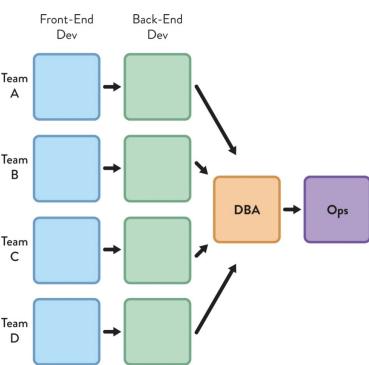
Carga Cognitiva

**Usar a estrutura da  
organização atual (Lei de  
Conway) para refatorar a  
organização**

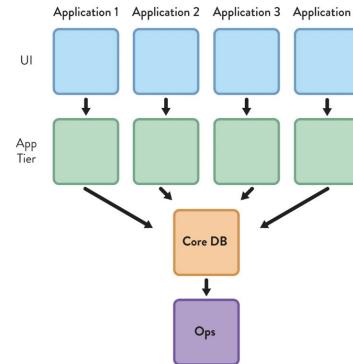
*Manouver  
Conway*

**Lei de Conway  
Inversa**

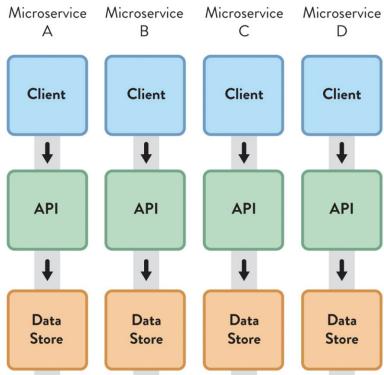
# Aplicando a Manobra de Conway Inversa



Team  
Refactory



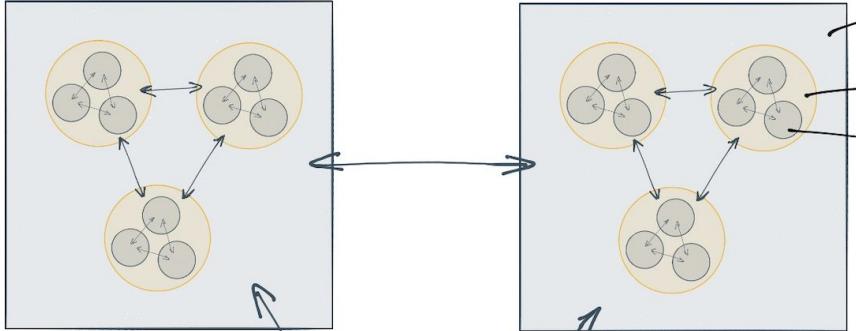
Anti-Pattern  
Fan-in



→

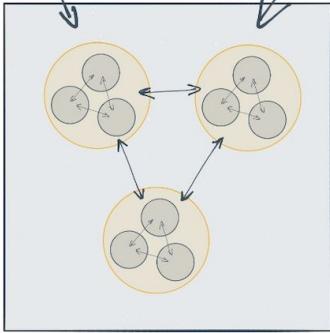
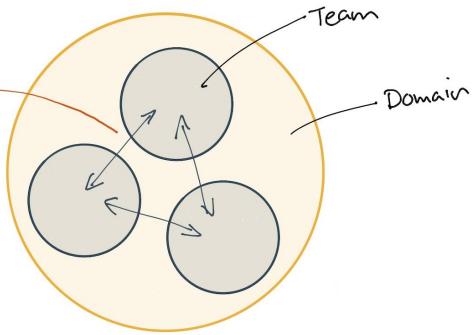


Microservice  
Pattern

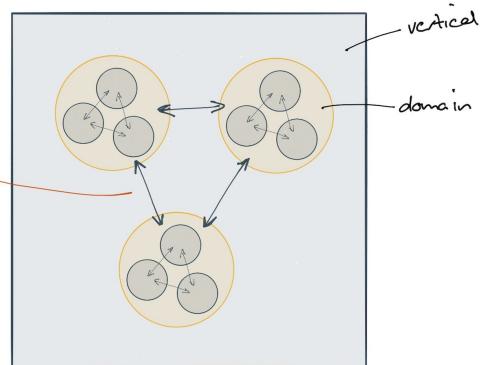


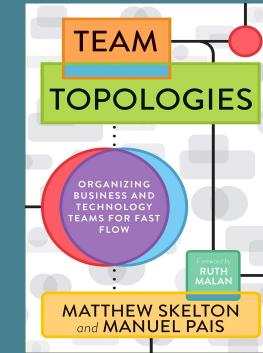
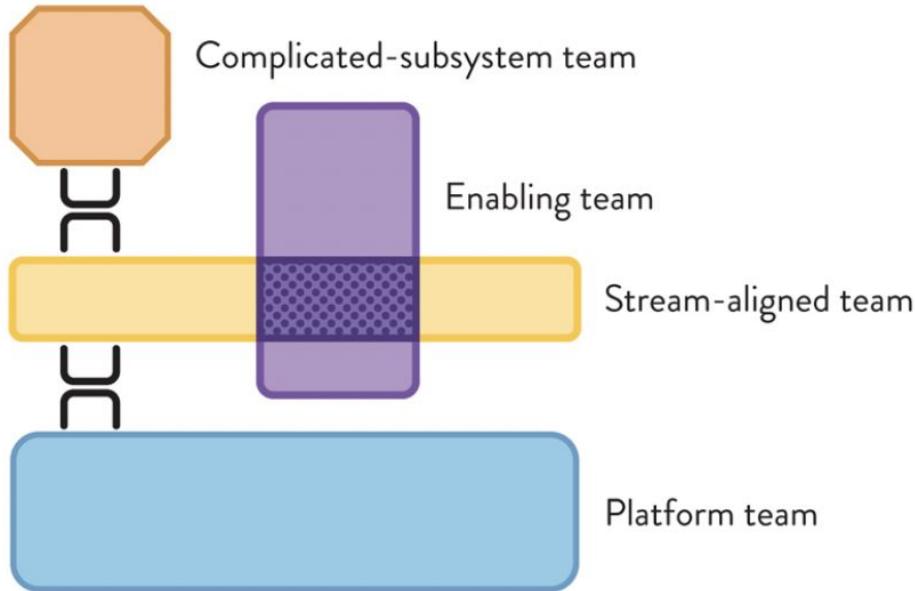
vertical  
domain  
team

Strong forces  
close alignment  
rapid change  
allows for tighter coupling

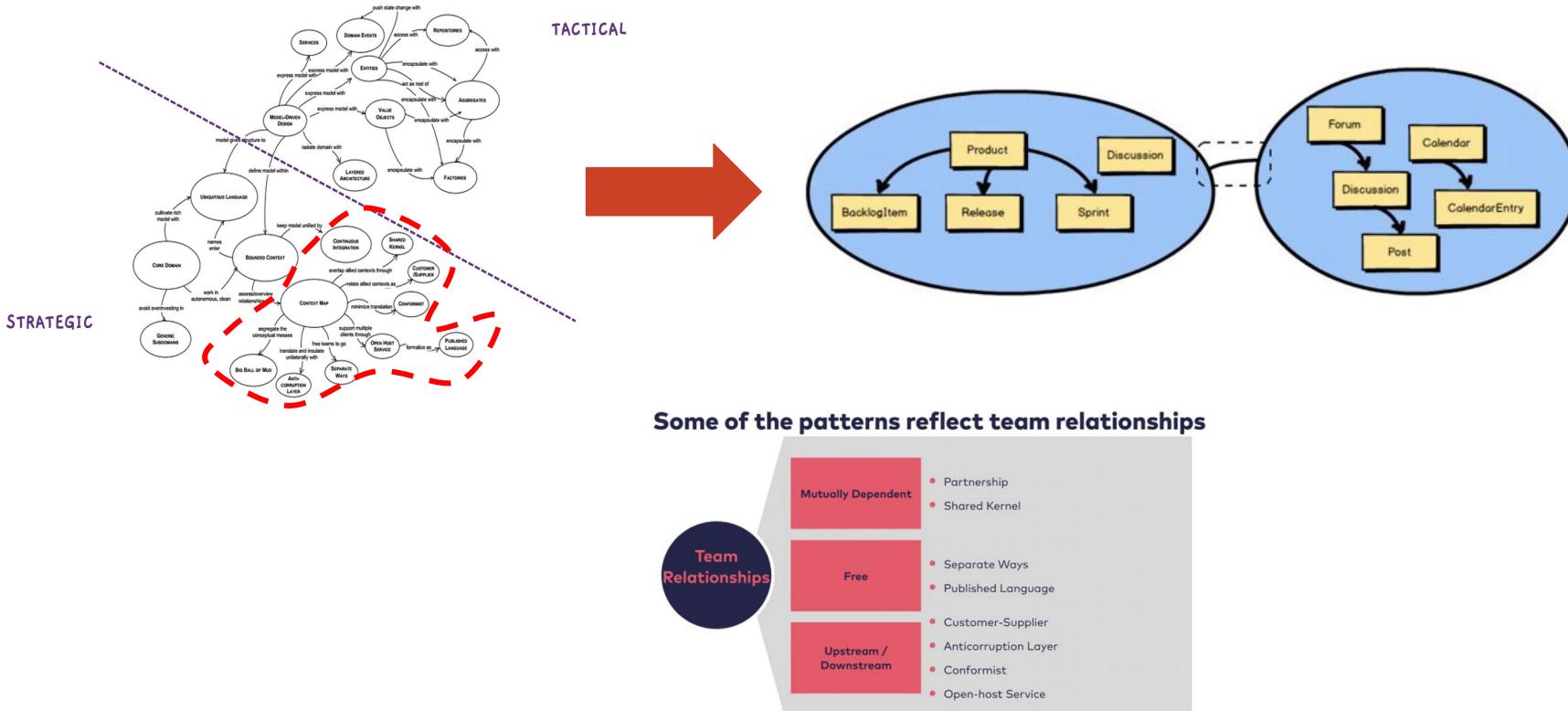


weaker forces  
less alignment  
change is slower  
need for more loose coupling

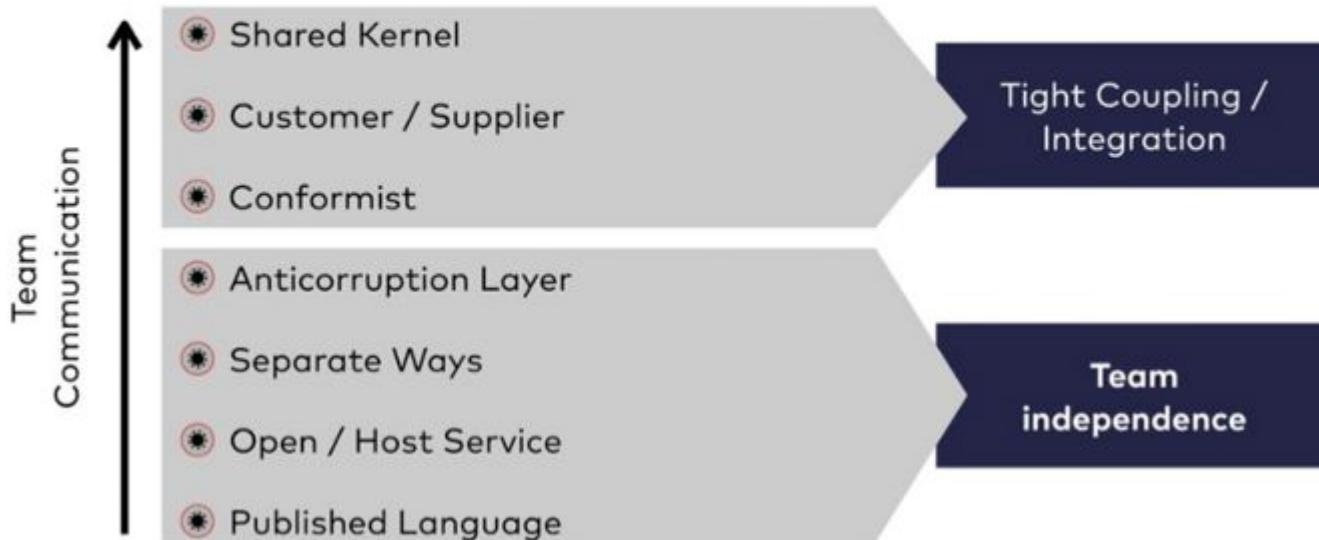




# DDD & Context Mapping



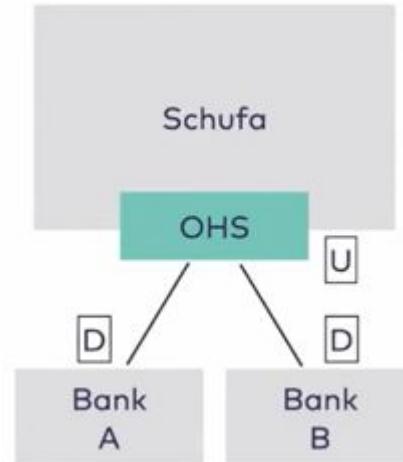
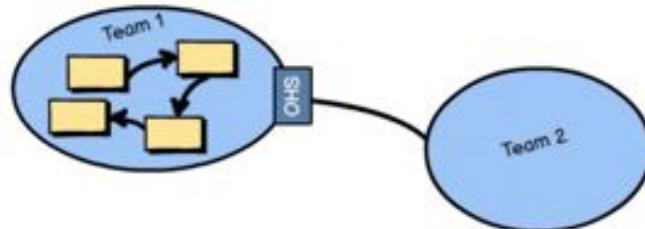
# Mind team communication



# Open-Host + Publish Language

Protocolo ou interface que dá acesso ao seu Bounded Context como um conjunto de serviços

O serviço Open-host é uma API pública





Sociotécnico,  
o que é... eu digo?

Tecnologia



Sociotécnica



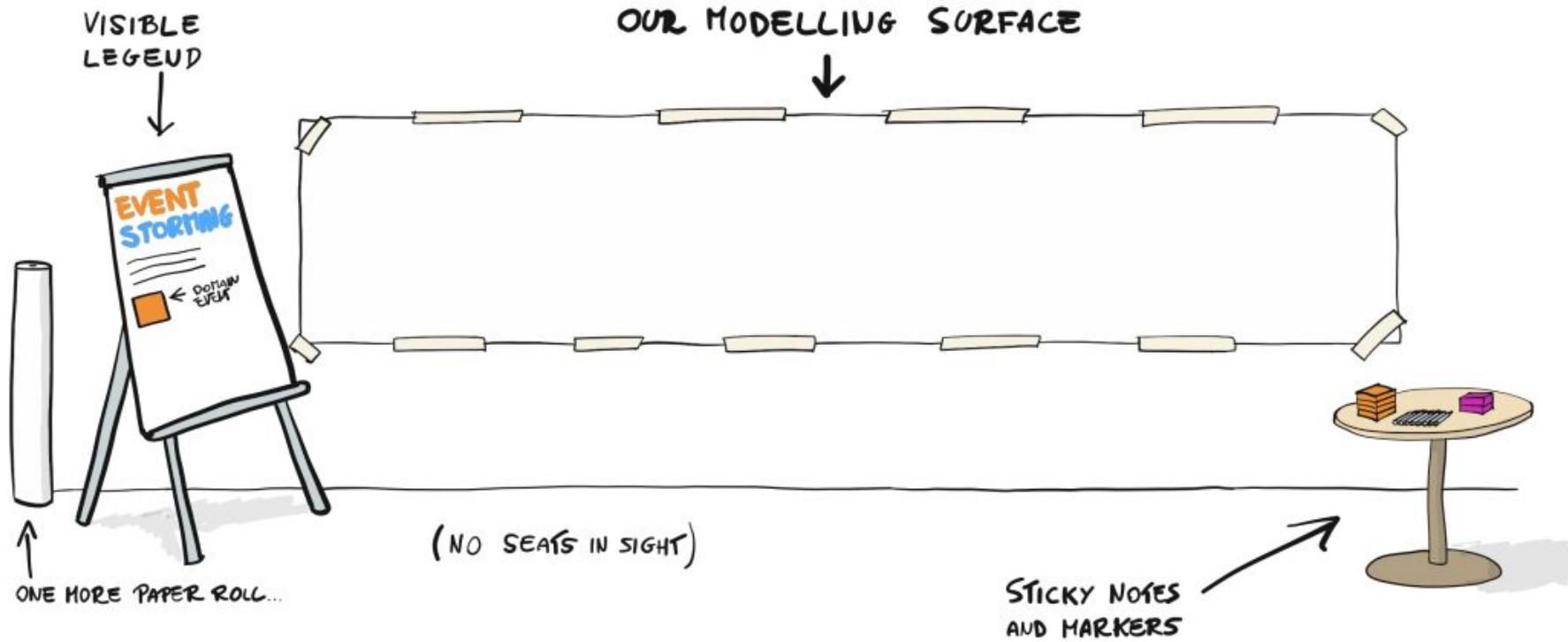
Organização

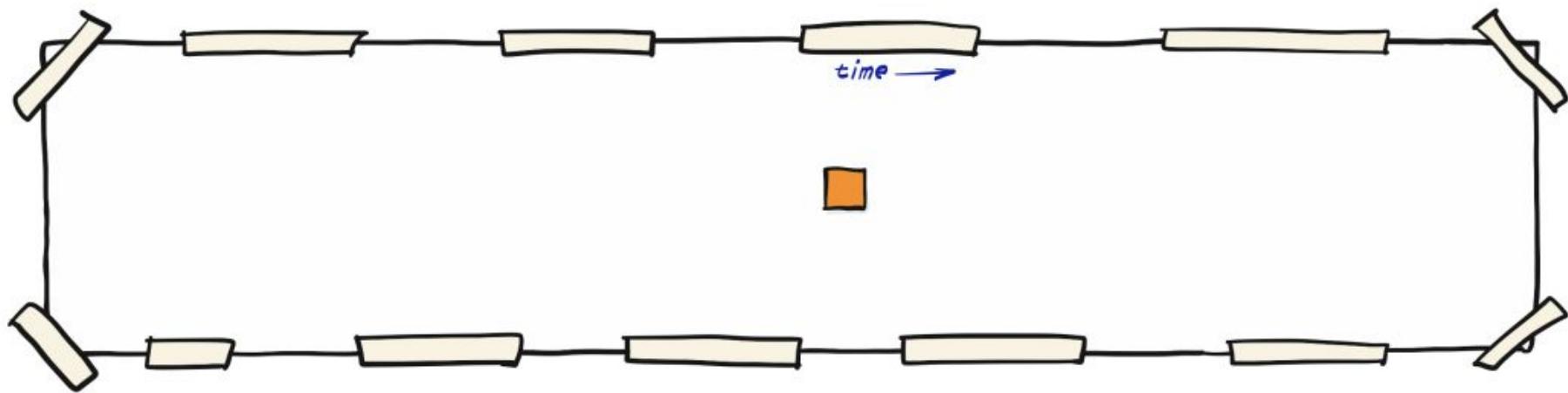


Pessoas

**Arquitetura Sociotécnica**  
envolve habilidades e  
técnicas para organizar  
pessoas e tecnologia

- EventStorming
- Team Topologies
- DDD







ITEM  
ADDED TO  
CART

THIS IS A DOMAIN EVENT

- ORANGE STICKY NOTE
- VERB AT PAST TENSE
- RELEVANT FOR DOMAIN EXPERTS



END OF  
MONTH

3

BILLABLE  
AMOUNT  
CALCULATED

NOBODY  
) REALLY  
KNOWS  
HOW

BILLABLE  
AMOUNT  
VERIFIED

INVOICE  
PREPARED

THIS IS  
WHERE  
EVERYTHING  
IS STUCK!!

END OF  
MONTH

3

BILLABLE  
AMOUNT  
CALCULATED

NOBODY  
REALLY  
KNOWS  
HOW

BILLABLE  
AMOUNT  
VERIFIED

INVOICE  
PREPARED

OMG  
IT TAKES  
AGES !!!

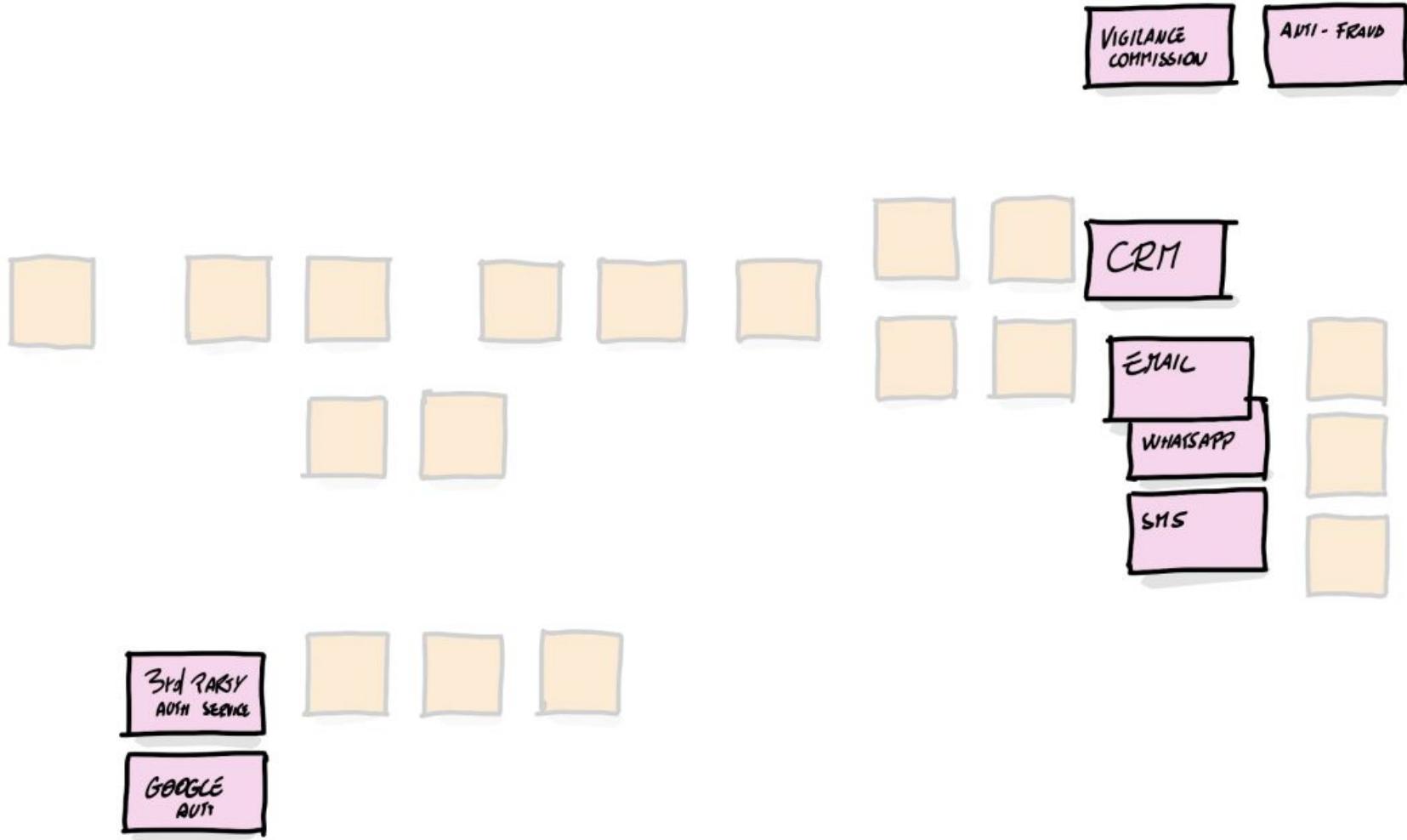
THIS IS  
WHERE  
EVERYTHING  
IS STUCK !!

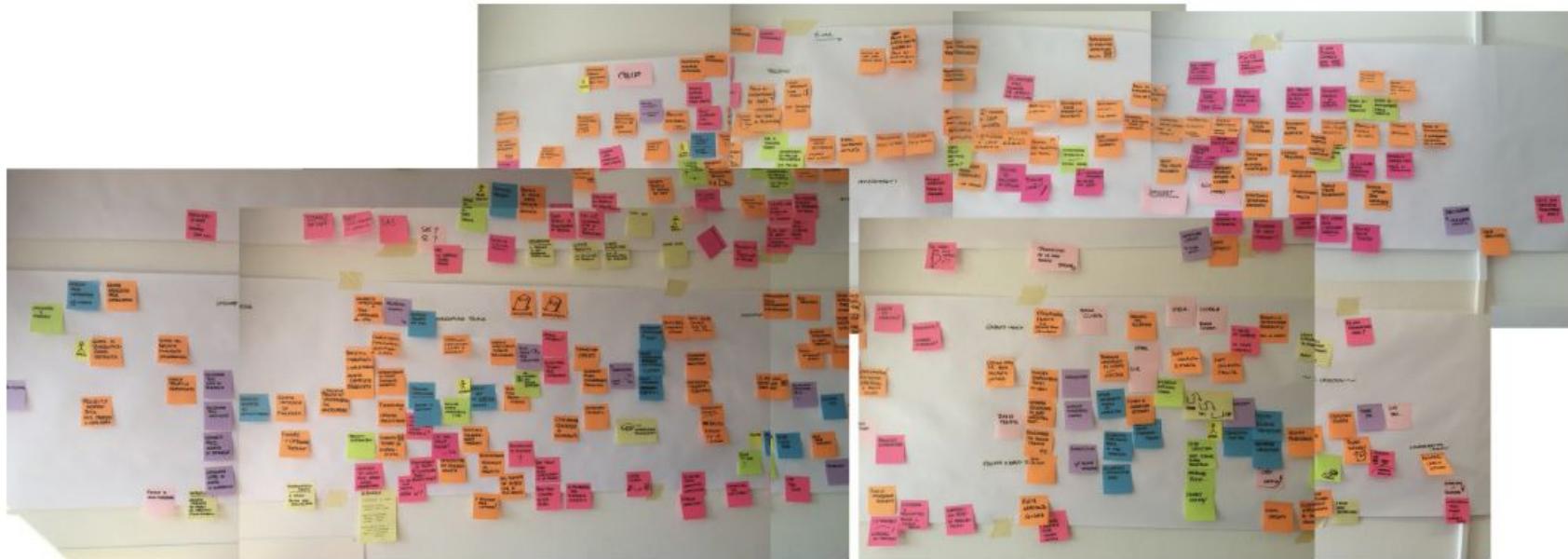
ERRORS  
ERRORS  
ERRORS  
:(

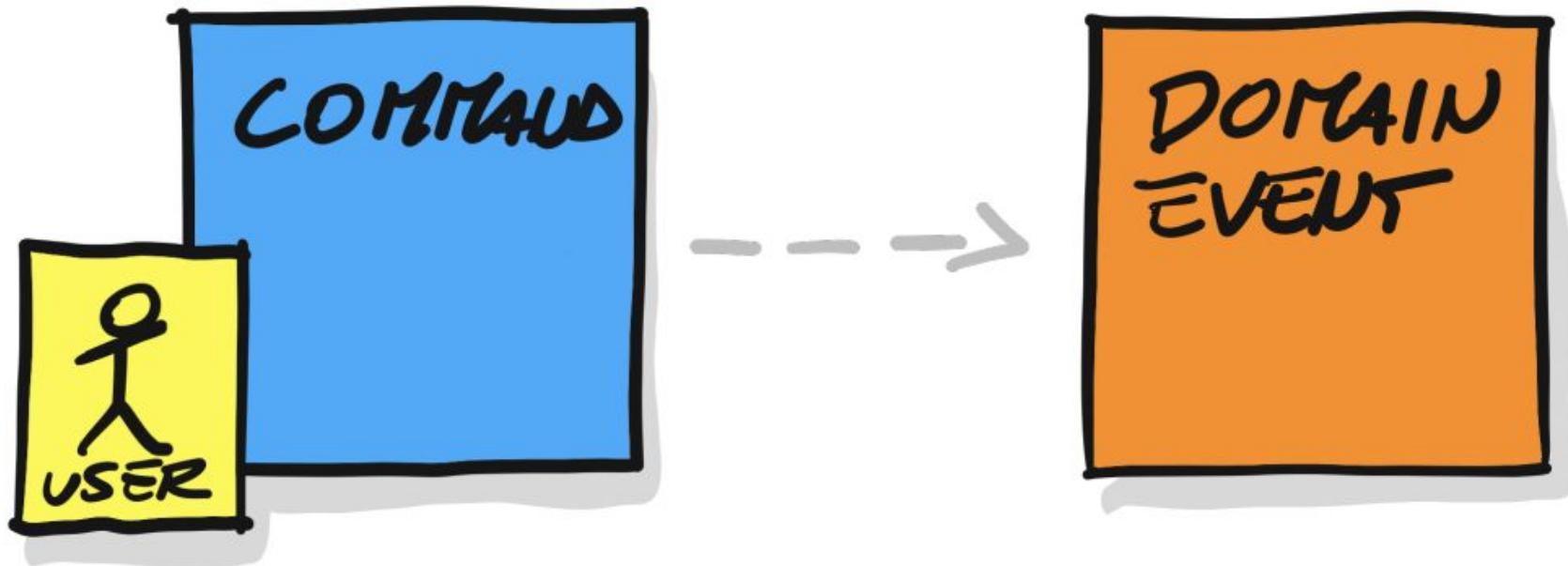
# INVESTMENT

The amount actually  
put in a specific  
Loan.

When precision emerges from the words of a domain expert, better write it somewhere...

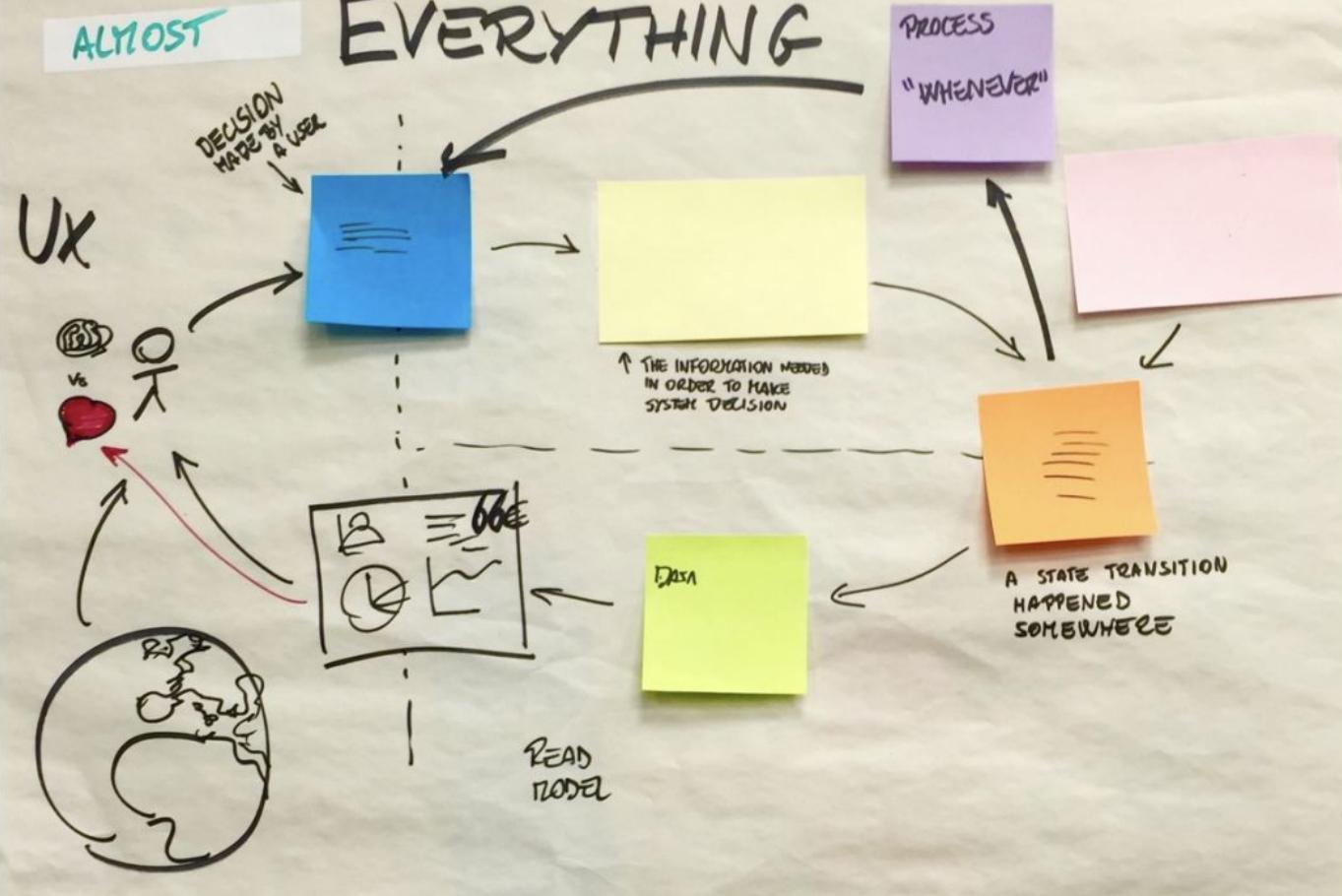


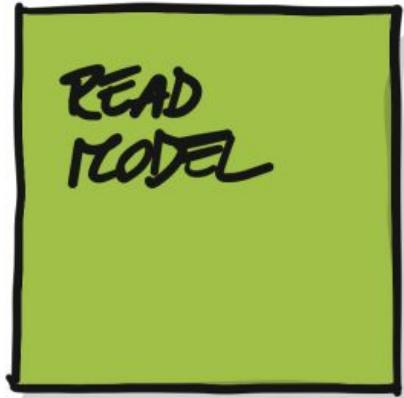
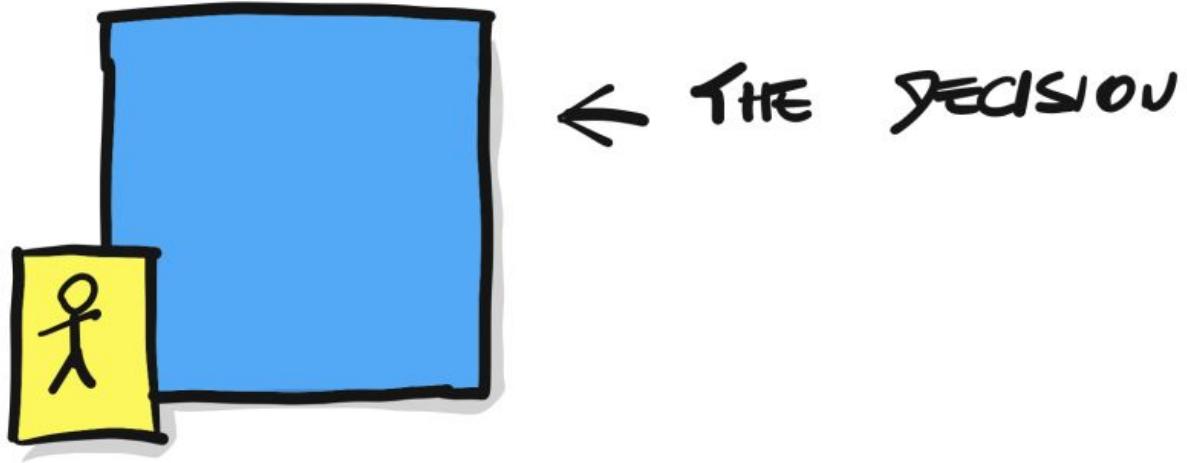




USER INITIATED ACTION

# THE PICTURE THAT EXPLAINS, EVERYTHING

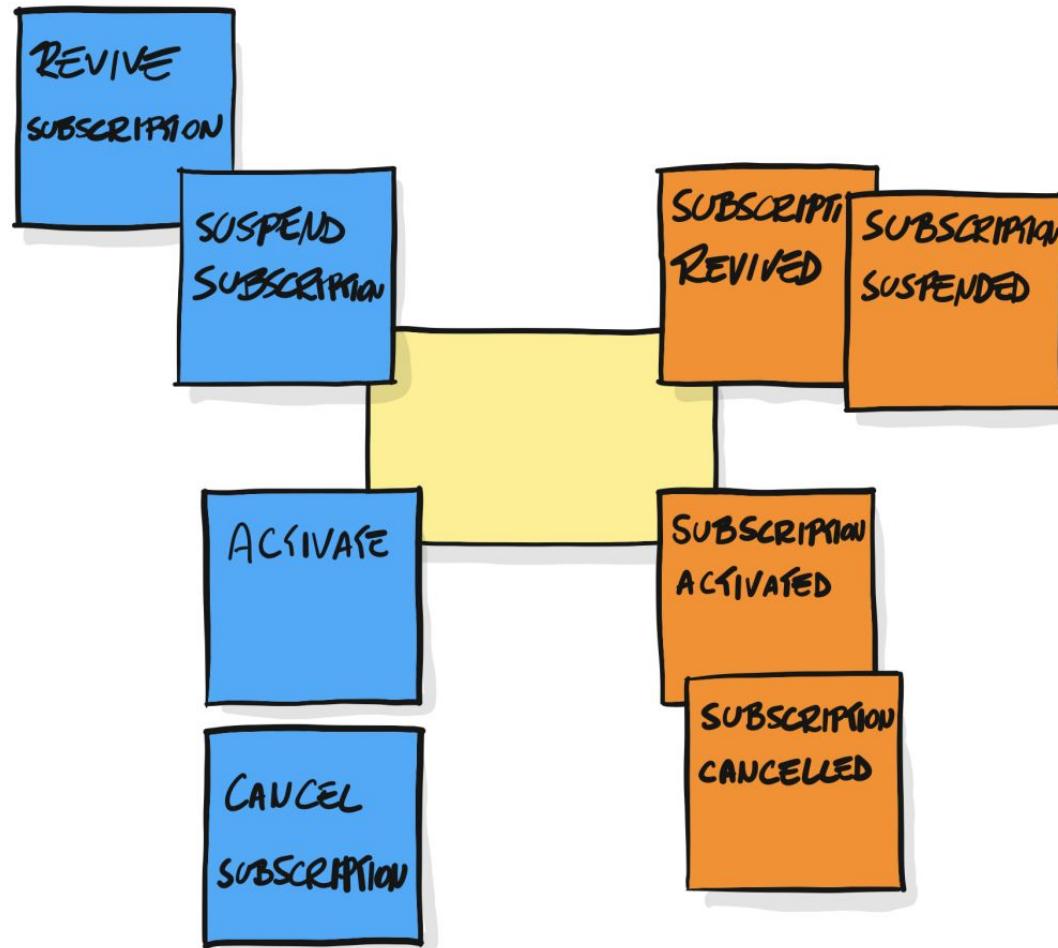




← THE DATA NEEDED  
IN ORDER TO  
MAKE THAT DECISION

"WHENEVER"

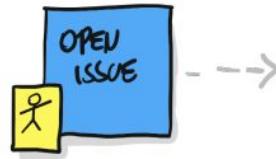
THIS IS OUR  
**POLICY**  
SOMETIMES IT'S  
AUTOMATED,  
SOMETIMES  
PEOPLE JUST  
HAVE TO REMEMBER



*A very simple aggregate, apparently*

## WHERE ARE DOMAIN EVENTS COMING FROM?

MAYBE AN **ACTION**  
STARTED BY A **USER**



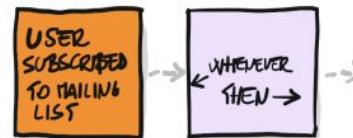
MAYBE THEY'RE COMING  
FROM AN **EXTERNAL SYSTEM**



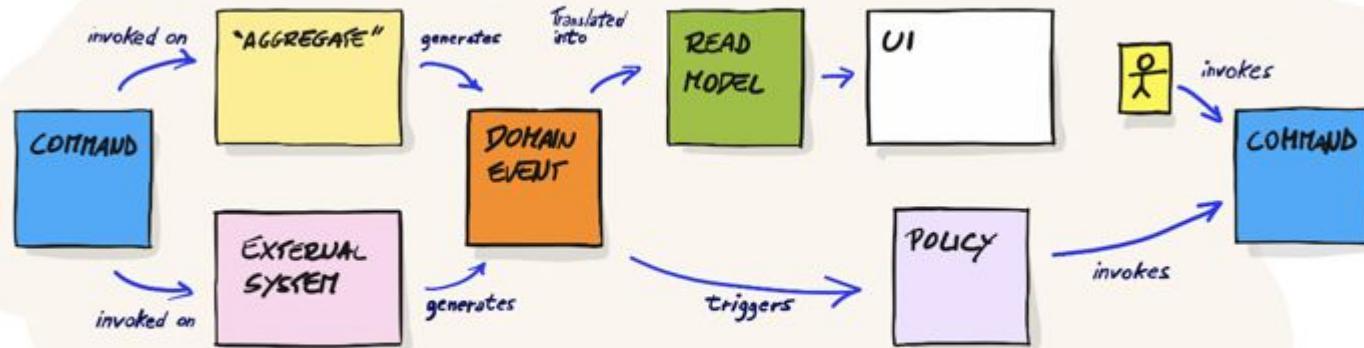
MAYBE THEY'RE JUST THE  
RESULT OF **TIME PASSING**



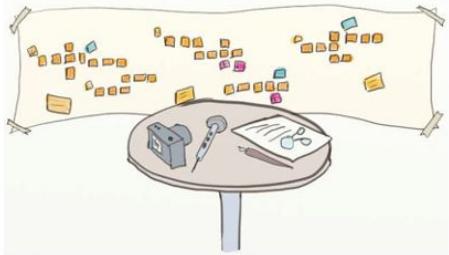
OR MAYBE, THEY'RE JUST  
THE **CONSEQUENCE**  
OF ANOTHER DOMAIN EVENT



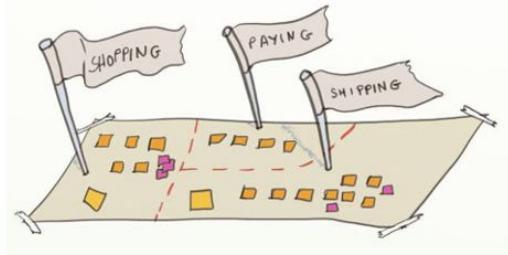
Where are domain events coming from? Four different possible answers to the question.



## Shared understanding



## Context map



## User stories



"When user X does Y, he wants to see Z, in order to ..."

## In / Out messages

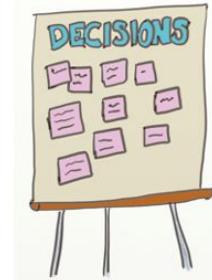


Clear contracts

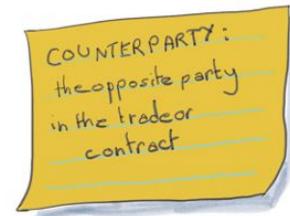
## Next Actions



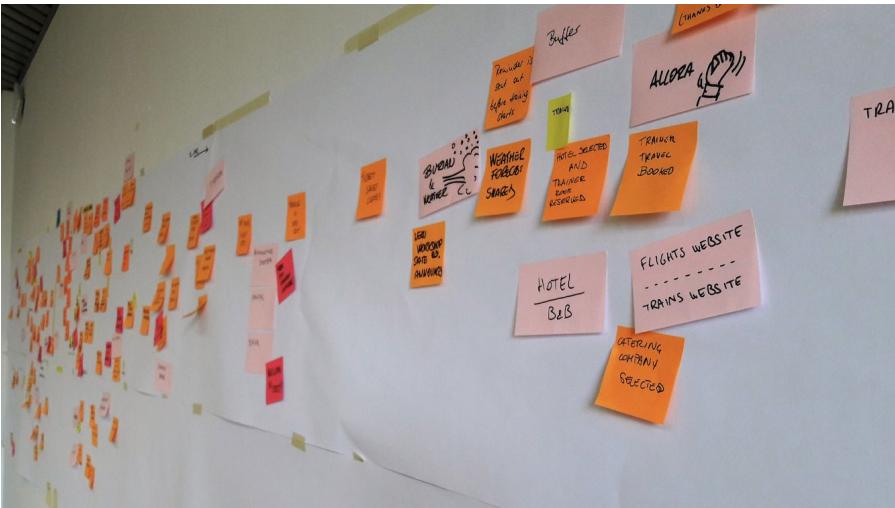
## Decisions record



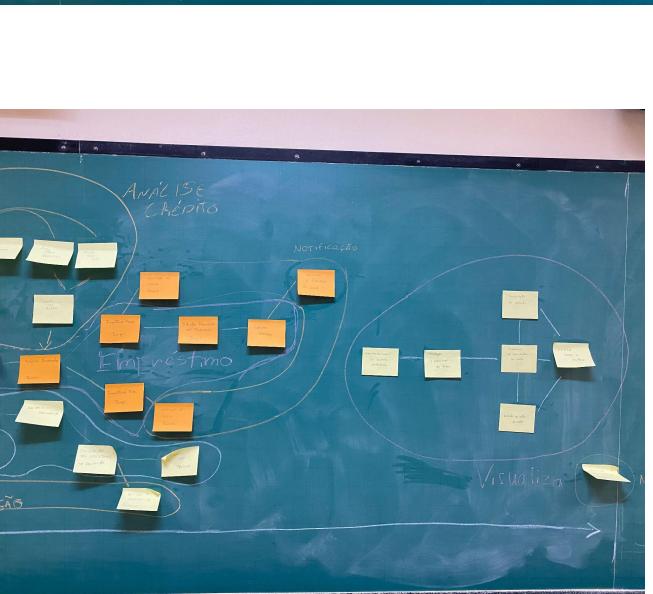
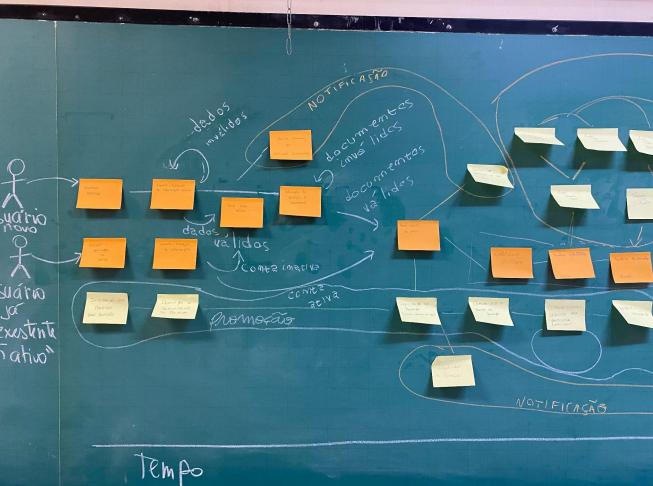
## Captured domain definition



# Hands-on - EventStorming



- 1) Dividir em Equipes (Squads)
  - 2) Usa a dinâmica de EventStorming para fazer o mapeamento de domínios de uma aplicação envolvendo os seguintes requisitos
    - a) Realizar empréstimos bancários
    - b) Ganhar e participar de promoções
    - c) Receber notificação por celular
    - d) Possibilitar a visualização de dados no celular



## Links

- <https://github.com/ddd-crew/ddd-starter-modelling-process>
- <https://www.eventstorming.com/>
- <https://medium.com/nick-tune-tech-strategy-blog/what-is-a-domain-99f658b22d7d>
- <https://martinfowler.com/articles/strong-weak-arch.html>
- <https://github.com/aceiro>
- <https://github.com/aceiro/architecture-ref-card>



Obrigado,  
... eu digo !!