Refactoring Socio-Technical Systems

Streamlining Your Structures & Flows

Dr. Franziska Horn

Umm…what?

- **Socio-Technical (ST) System**:
  An organization where people and technology interact
  → focus here: software companies
- **Refactoring**:
  In computer science: changing how code is structured while preserving its purpose and functionality

⇒ Improving *how* a company does what it does

What brings me here

- PhD in Machine Learning (ML)
- Started out as a Data Scientist & Python Developer
- Team Lead at a startup: improved internal processes 💡

Refactoring in Practice

At that startup, many things were great 🤩

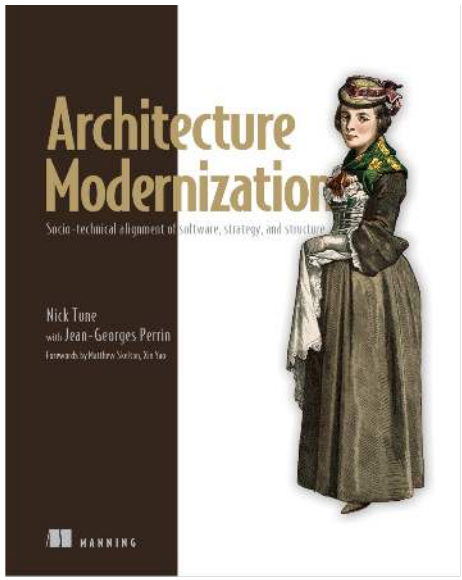
- Modular software architecture (multiple microservices)
- Deploy to production multiple times per day
- IaC and provision of new test environments via slackbot
- Weekly automated retraining of ML models on new data

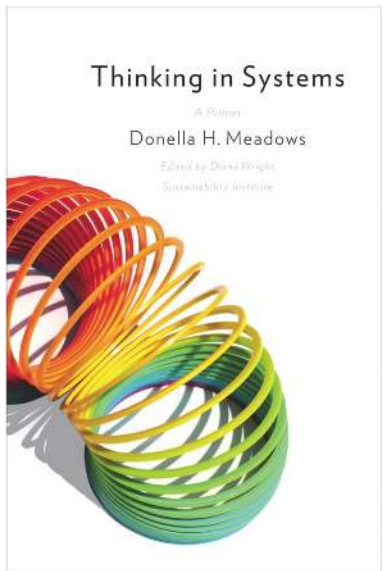
But…

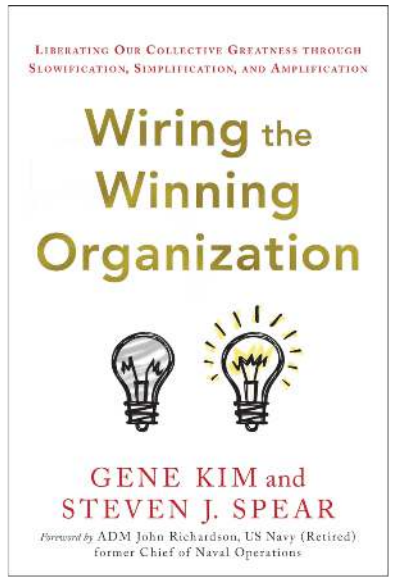
- Customer onboarding & data integration took 50 days 😱

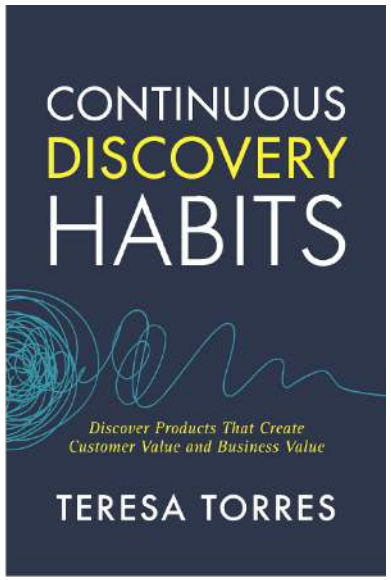
⇒ Reduced to 25 days through a series of small refactorings 🚀

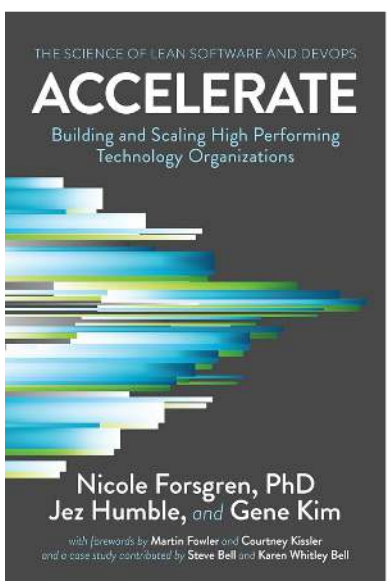
A Holistic Perspective on ST Systems

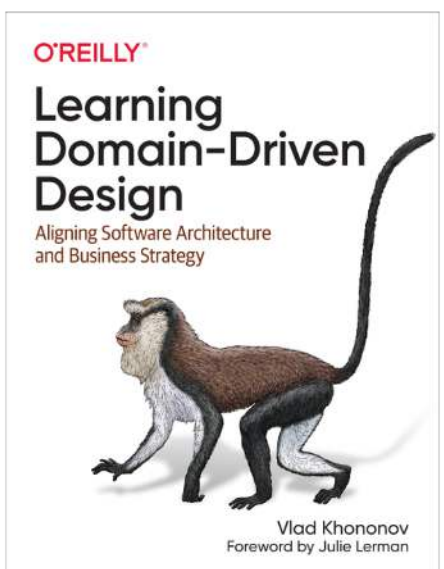


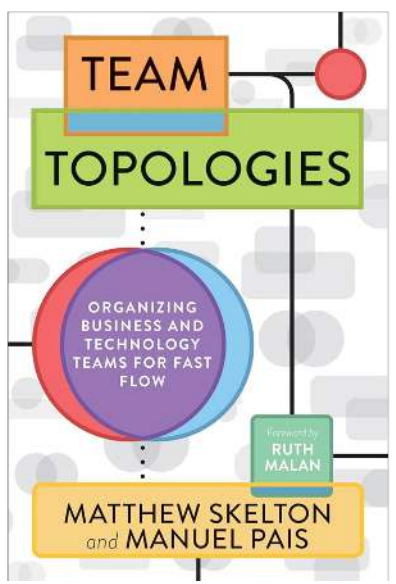


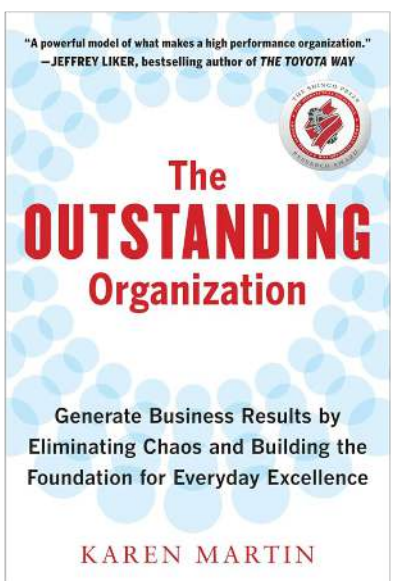


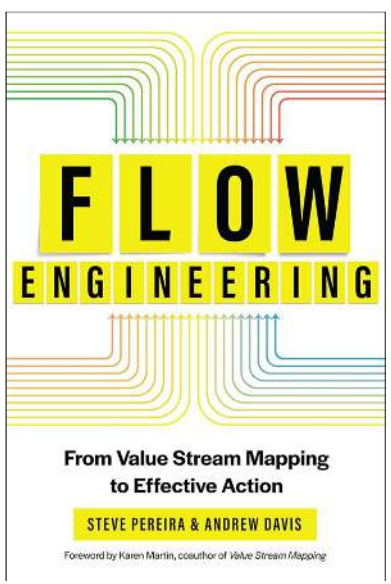




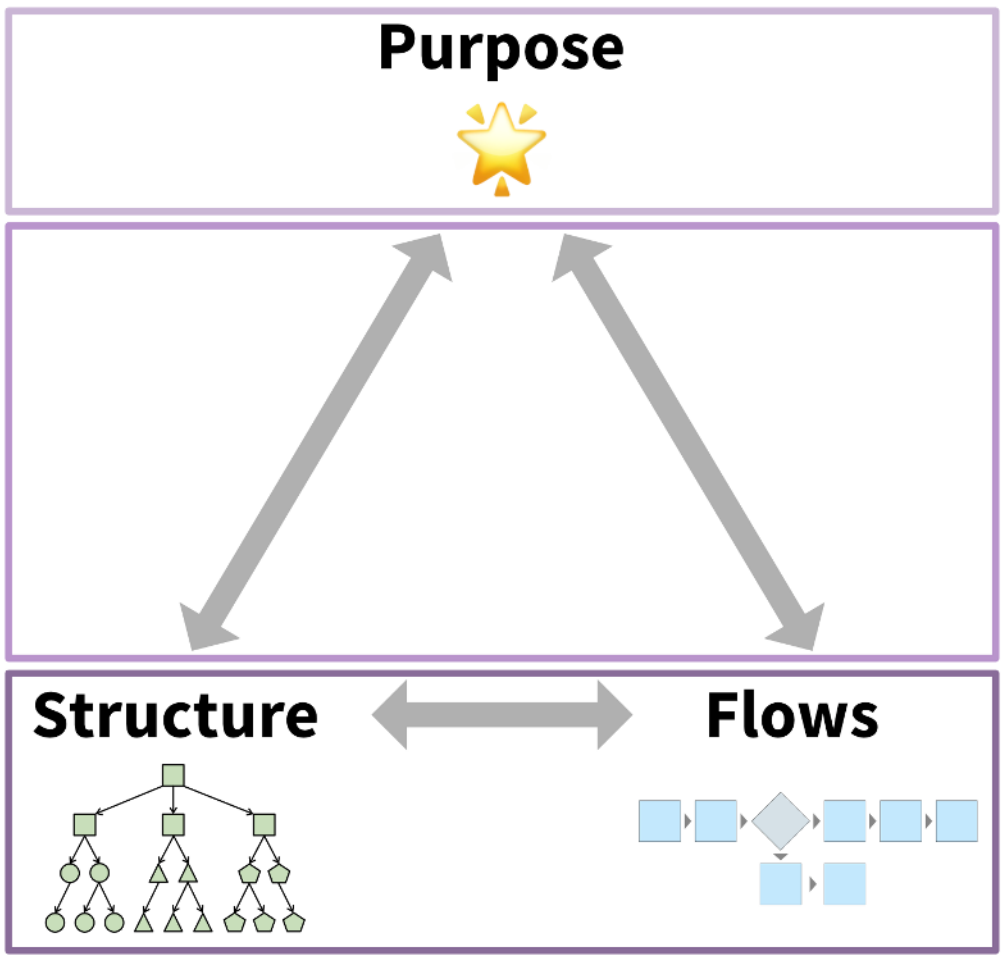








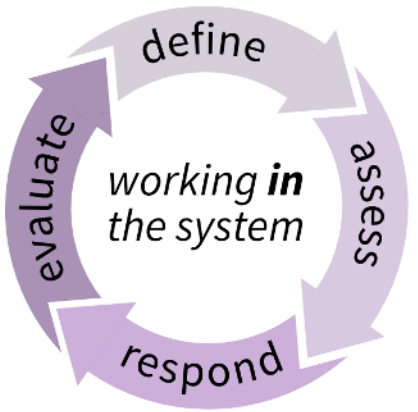
Your Company as a ST System

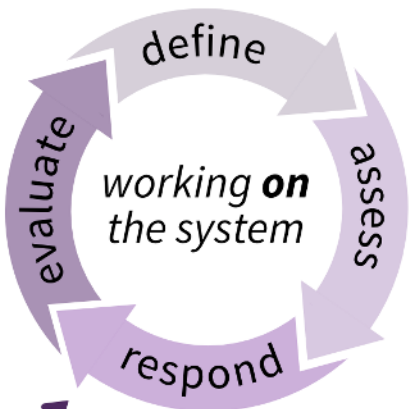


Why Outcome
Value for customers
Getting from A to B

What Output
Product / Service
Car

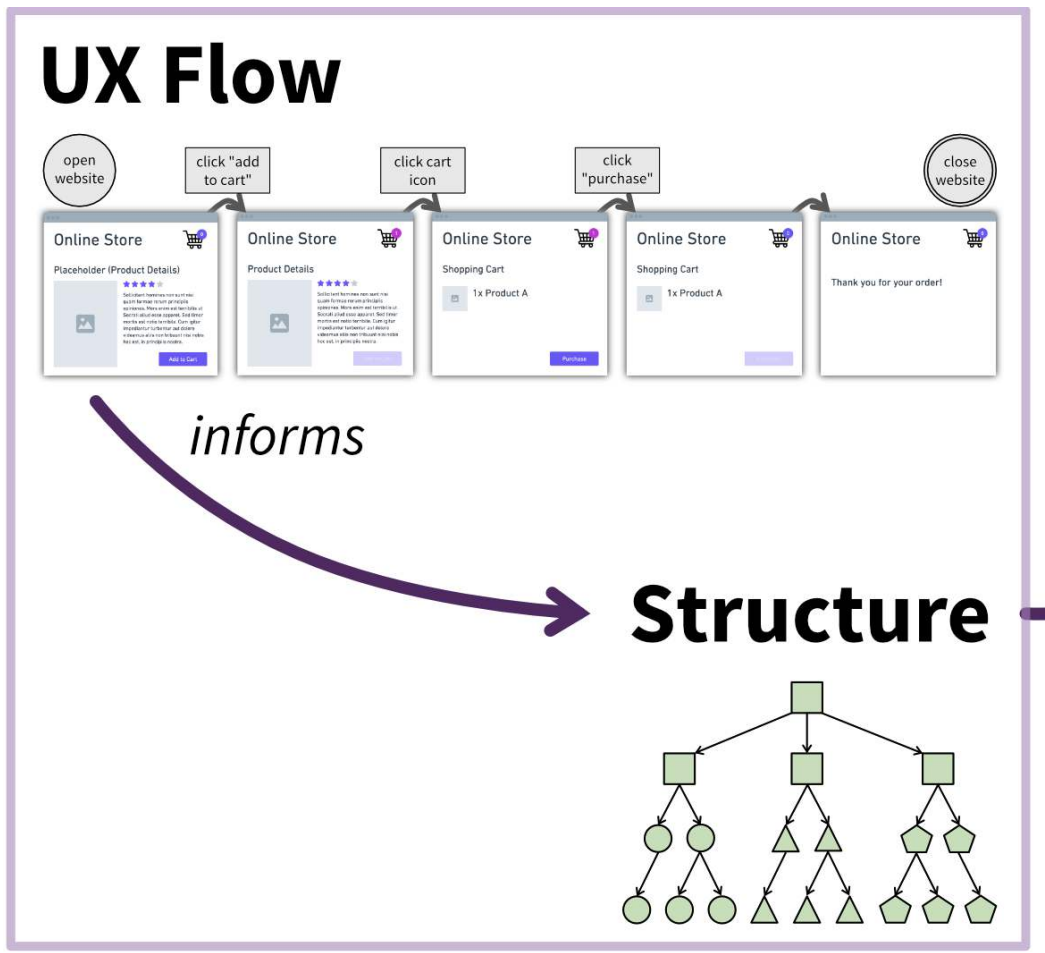
How Process
Manufacturing

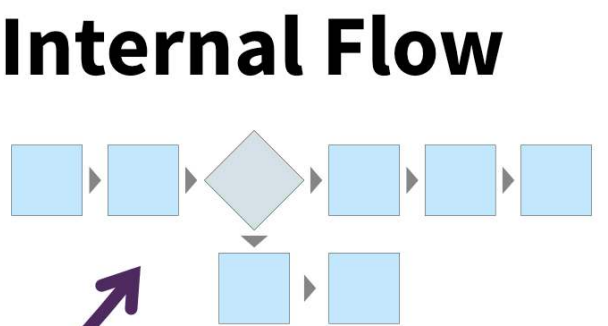




Refactoring

Refactoring ST Systems – Part 1





supports

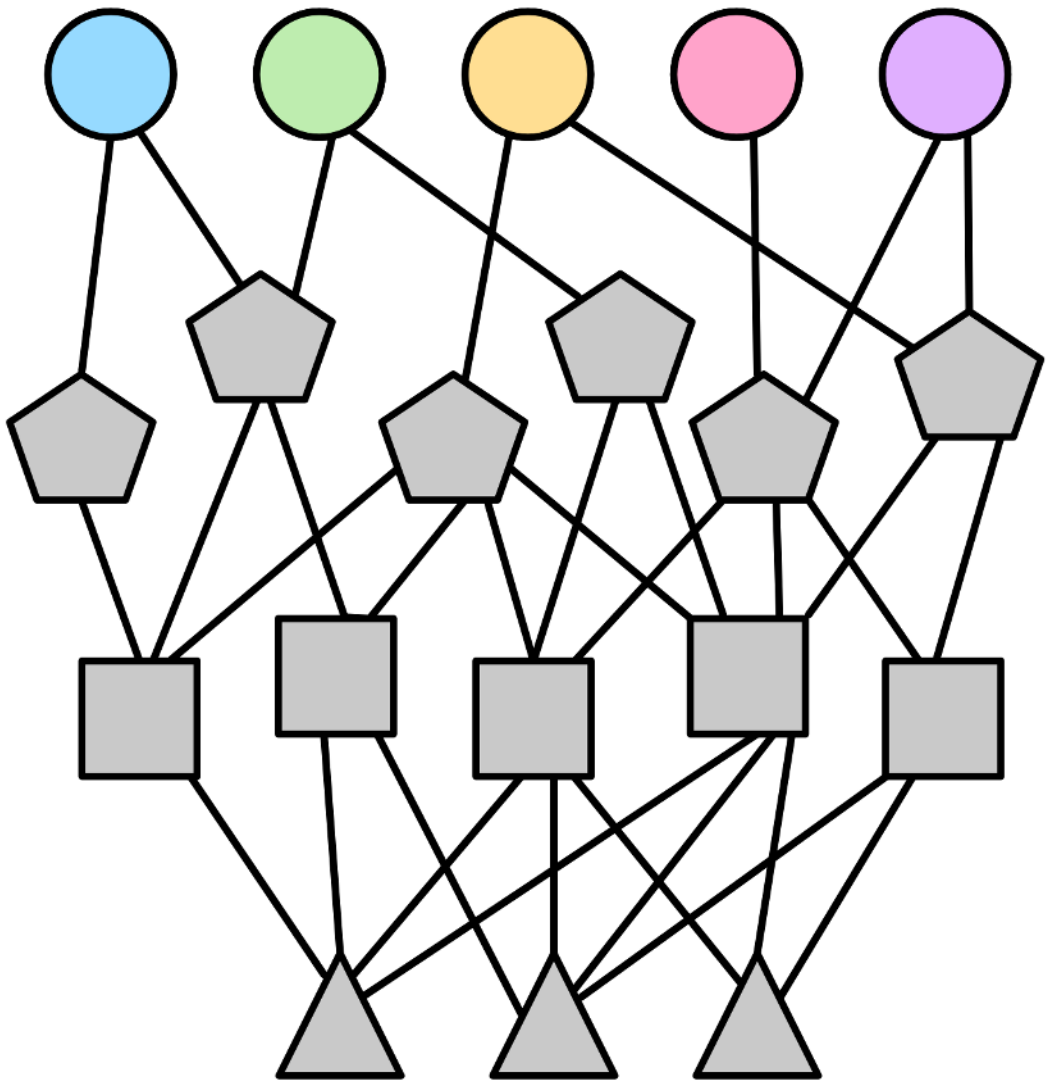
Aligning Structures with UX Flow

User
Domains / Use Cases

Product
Information Architecture

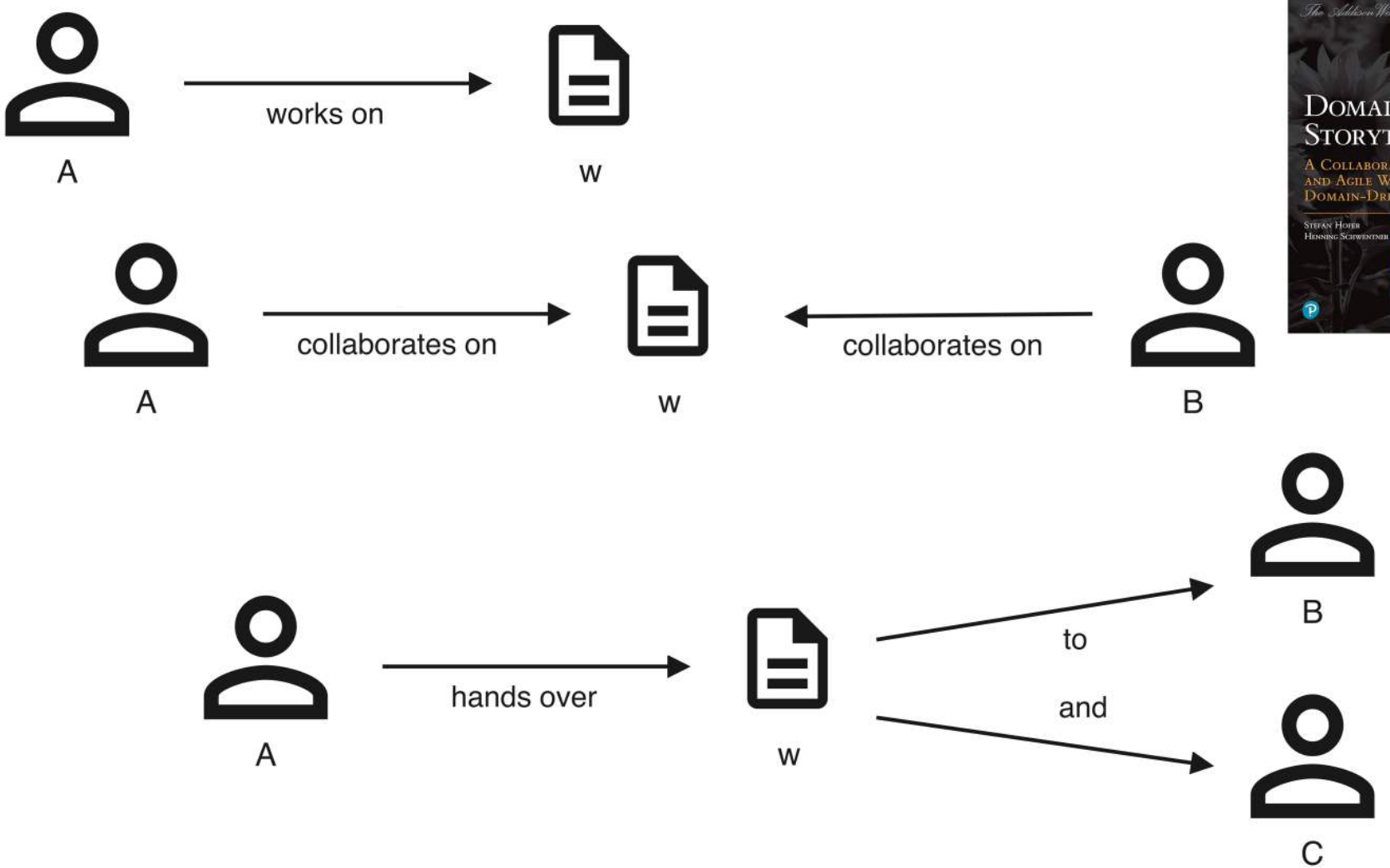
Teams
Organizational Structure

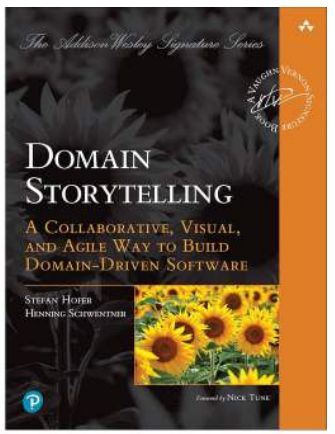
Code
Software Architecture



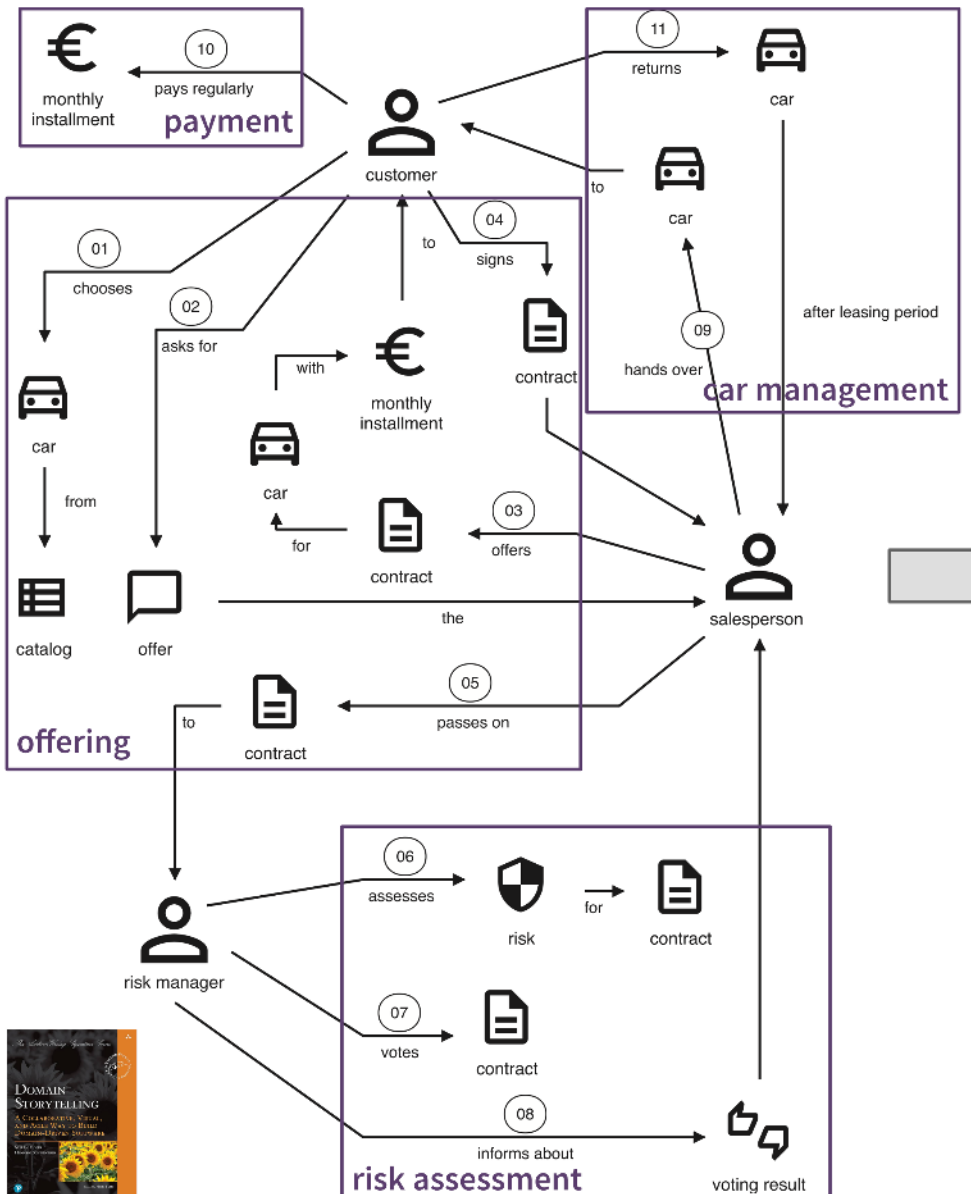
Technique: Domain Storytelling

Collaborative Modeling: Domain Experts + Developers





Domain Discovery



- Completed work object is handed off
- Different groups of domain experts work on different subprocesses
- Subprocesses have different rhythms and trigger events

Domains:

payment

offering

car management

risk assessment

Aligning Structures with UX Flow

User
Domains / Use Cases

Product
Information Architecture

Teams
Organizational Structure

Code
Software Architecture

Product UX: Information Architecture

Car Catalog

Information Architecture (IA) aligned with use cases:

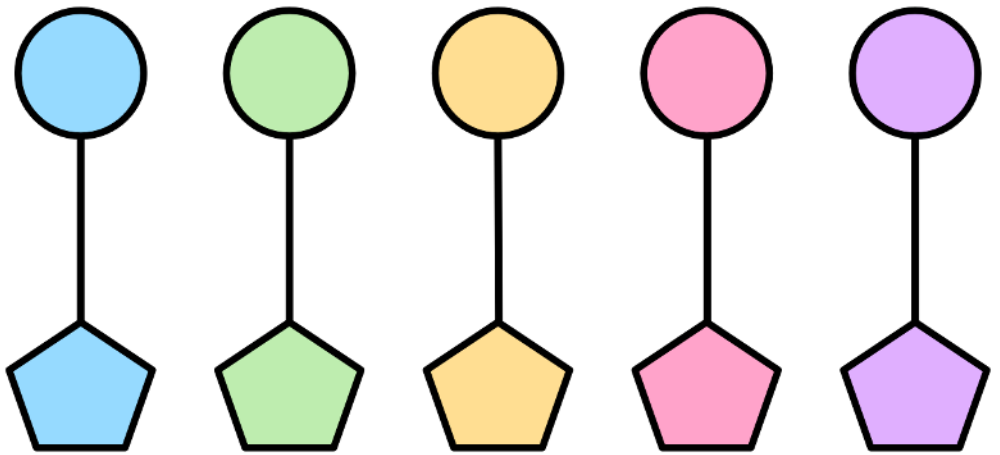
Minivan	City Car	Sports Car	Pickup
Model A	Model D	Model G	Model K
Model B	Model E	Model H	
Model C	Model F		

or

Audi	BMW	VW	Toyota
Model A	Model B	Model C	Model E
Model D	Model H	Model F	Model K
Model G			

Aligning Structures with UX Flow

User
Domains / Use Cases

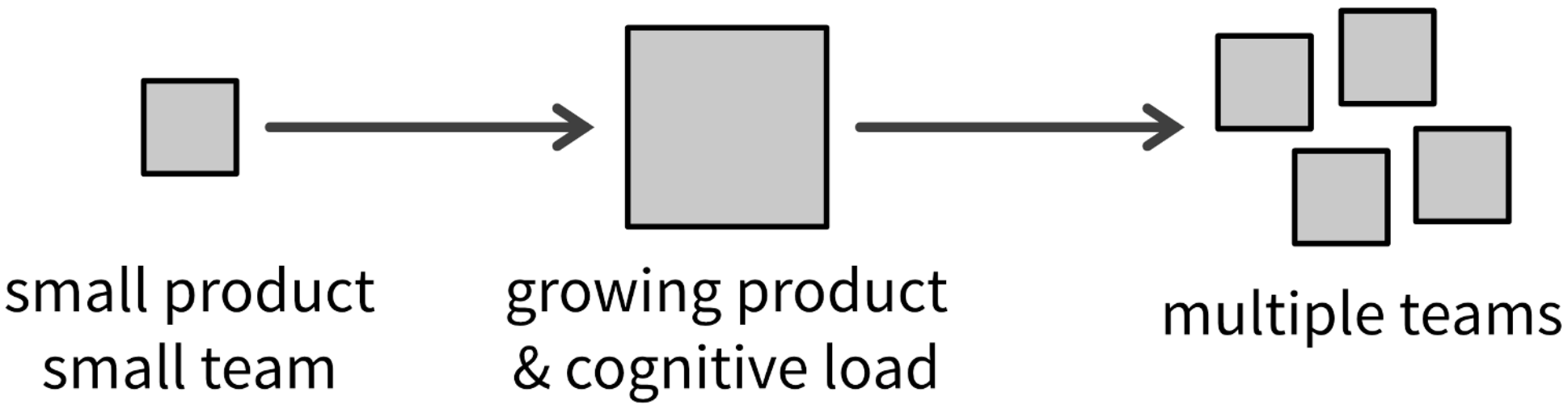


Product
Information Architecture

Teams
Organizational Structure

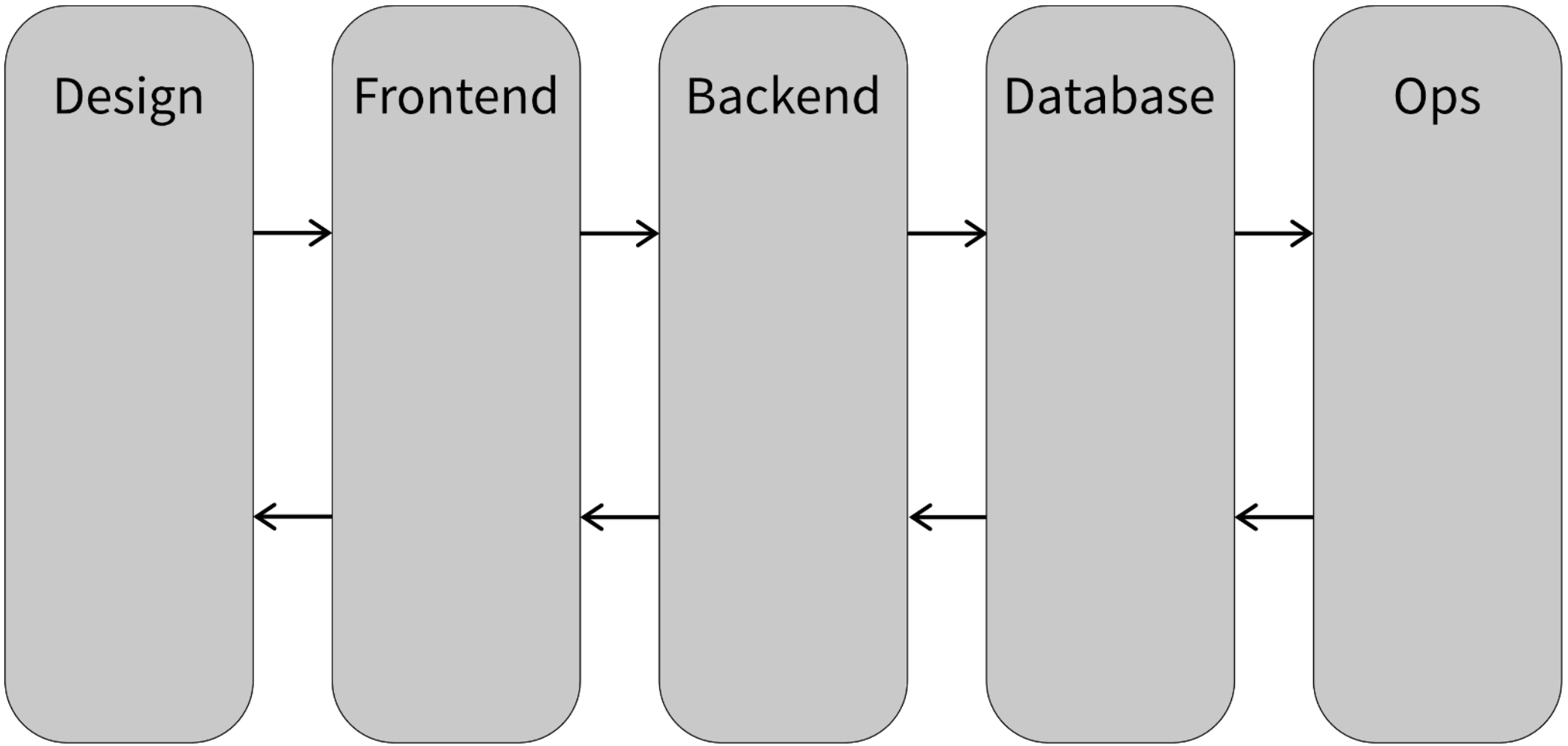
Code
Software Architecture

Growth & Increased Cognitive Load

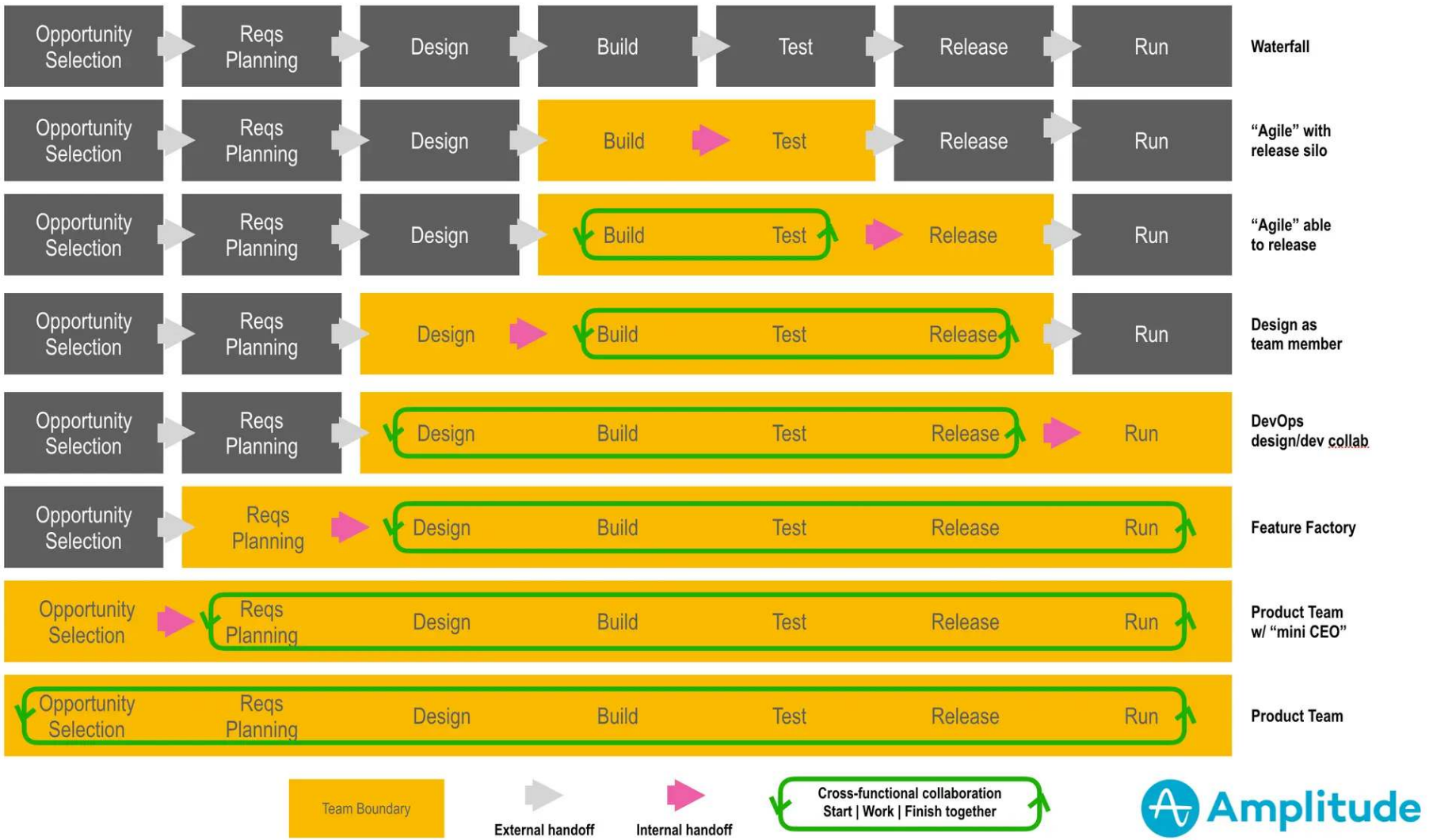


⇒ **Goal:** minimize hand-offs & dependencies between teams

Functional Teams: Dependencies

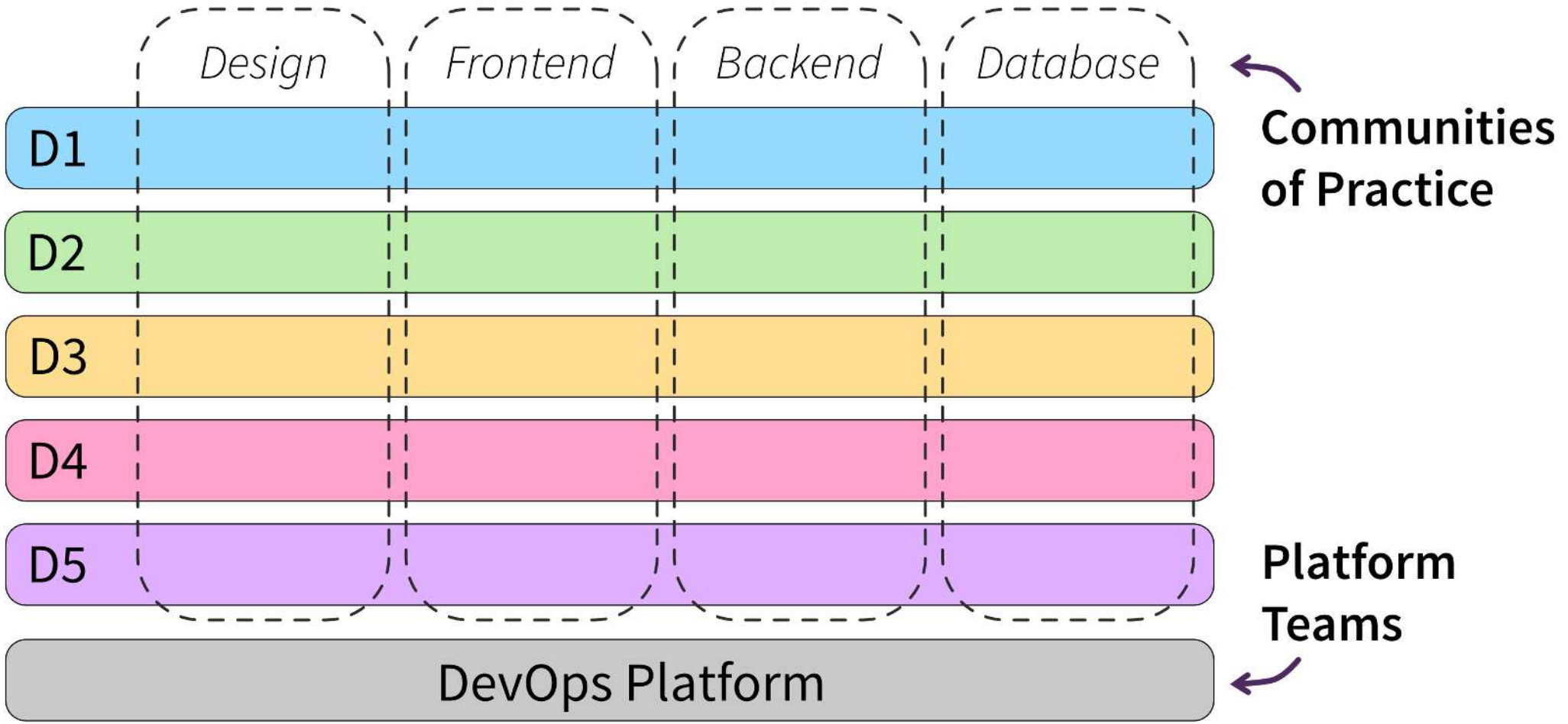


Close the Feedback Loop!

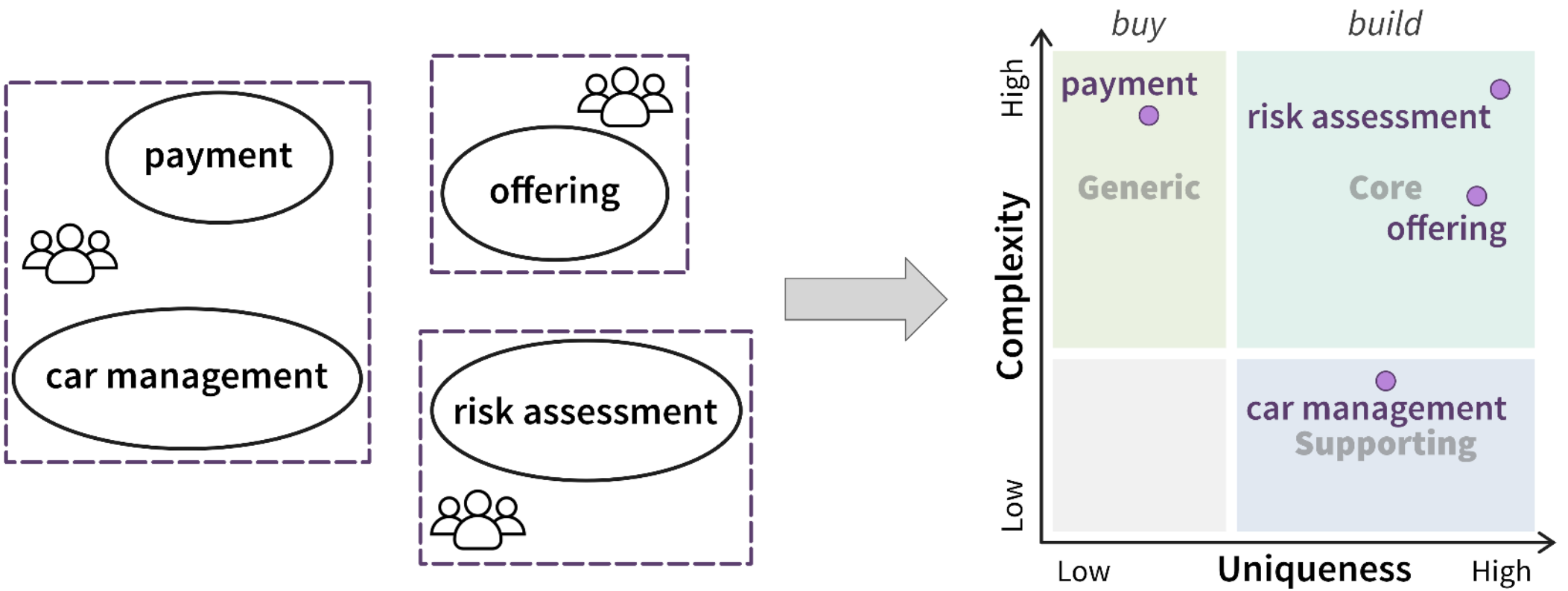


Source: https://amplitude.com/blog/journey-to-product-teams-infographic

Cross-Functional Teams (Matrix Org)



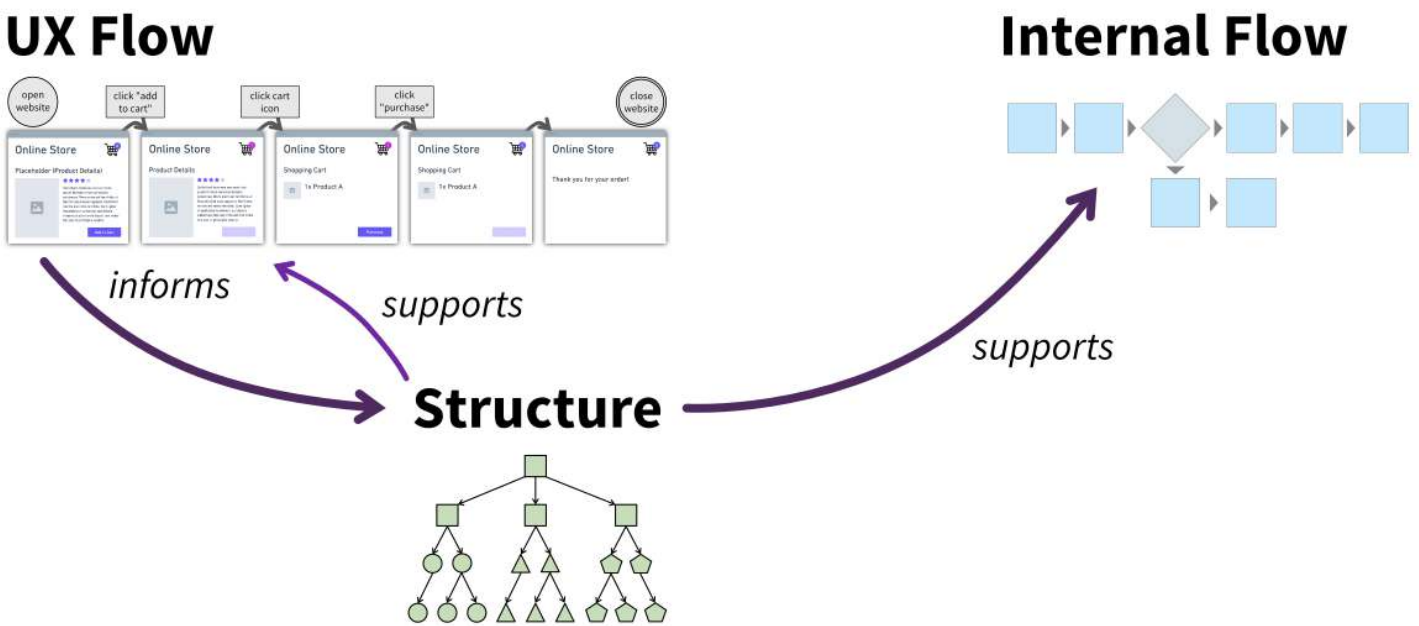
But How? Categorize Domains!



Inverse Conway Maneuver

Conway's Law:

> Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations.
>
> — Melvin E. Conway, How Do Committees Invent? (1968)



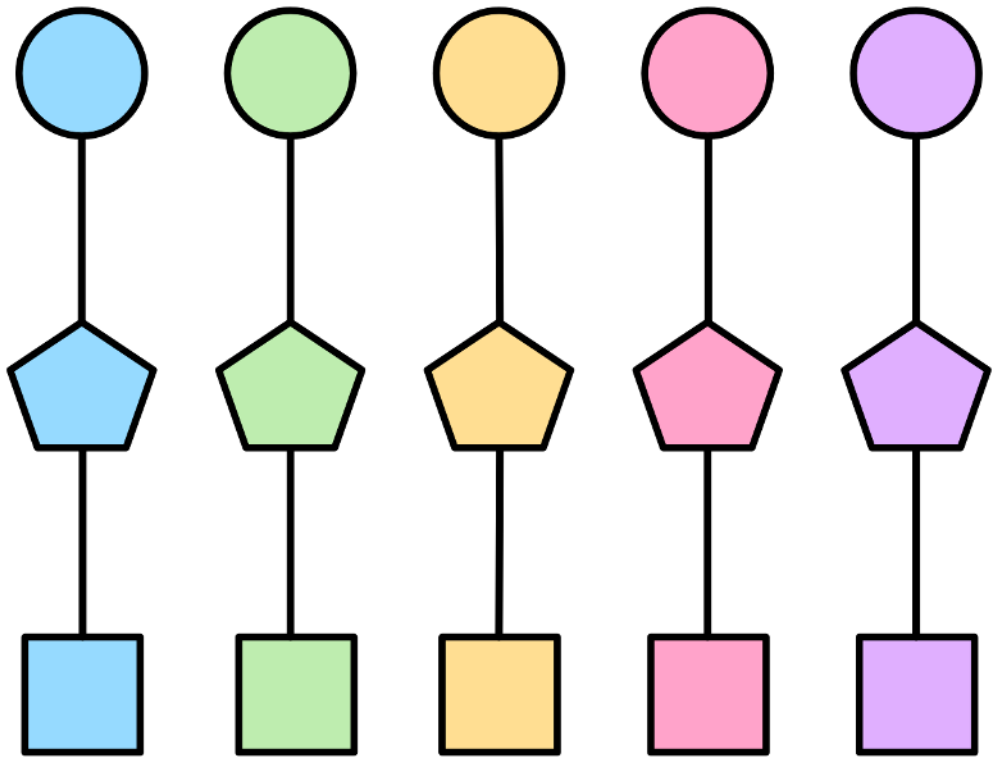
Aligning Structures with UX Flow

User
Domains / Use Cases

Product
Information Architecture

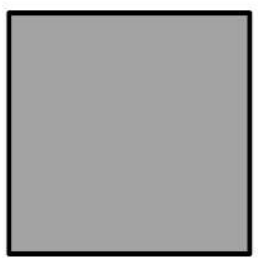
Teams
Organizational Structure

Code
Software Architecture



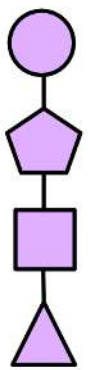
Balance Coupling

High Local Complexity



Monolith

Just right



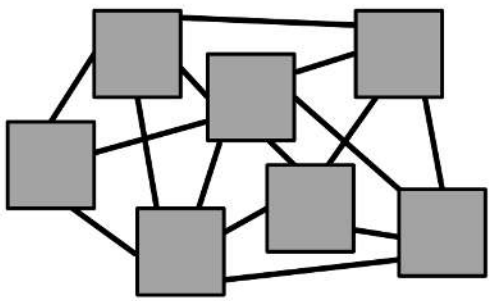
1 Domain

1 Service

1 Team

1 Repo & DB

High Global Complexity



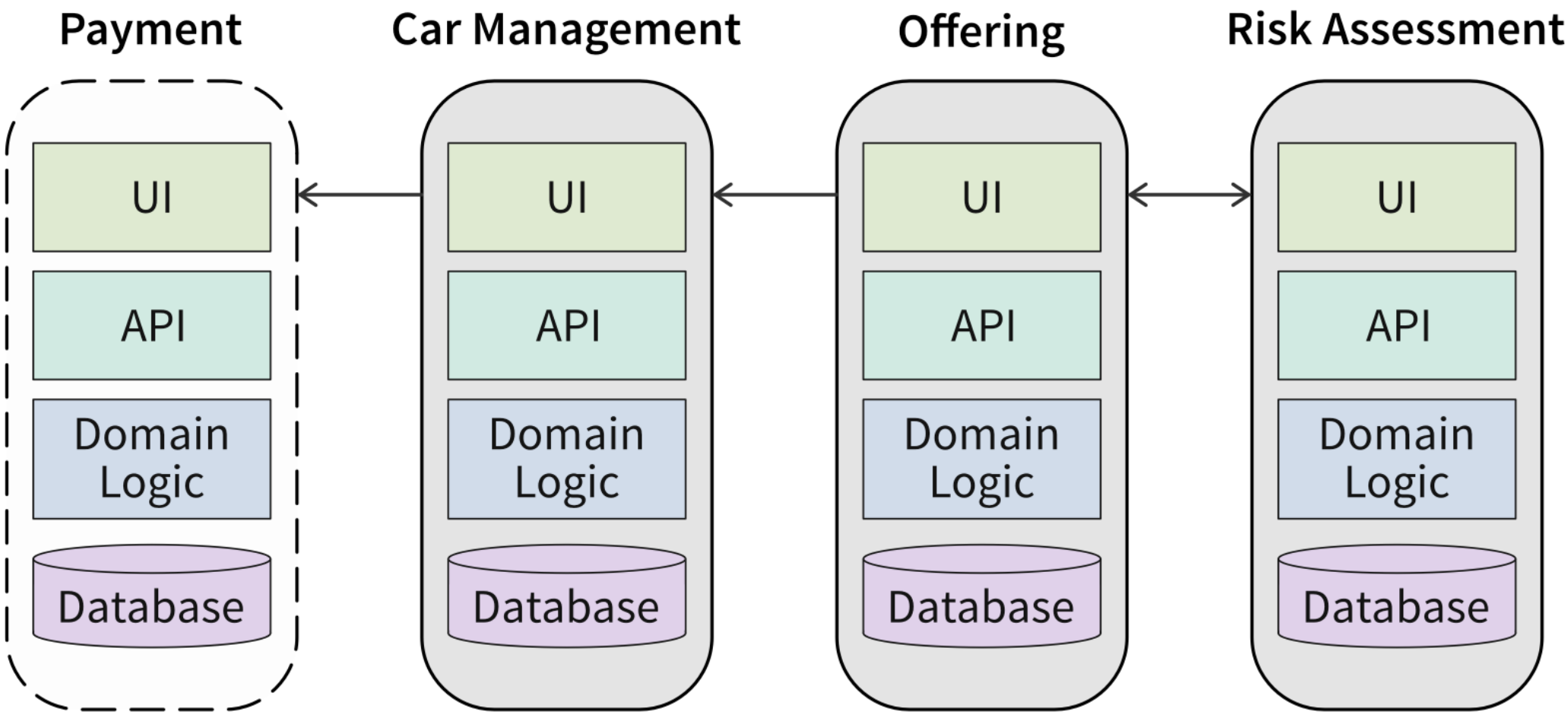
Coupled Microservices

keep it modular & needs good CI/CD

manageable cognitive load

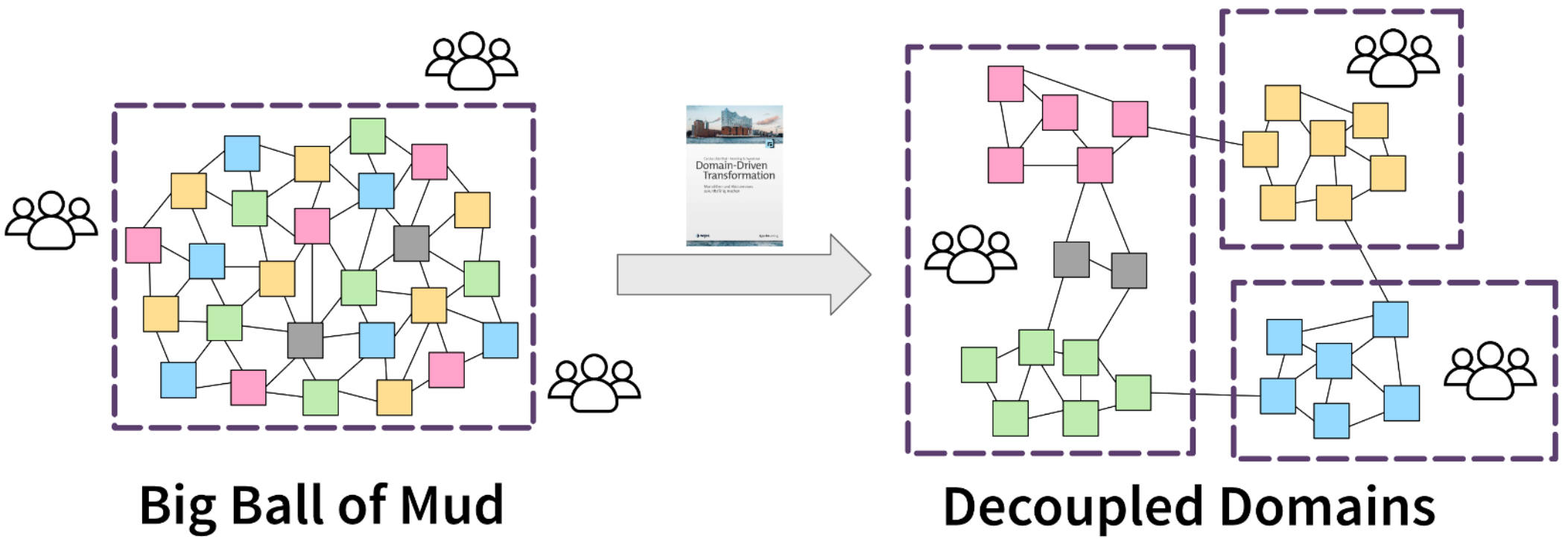
beware of coupling at DB

One Microservice per Domain



But our Legacy Software…

Coupled Code ⇒ Dependent Teams 😭



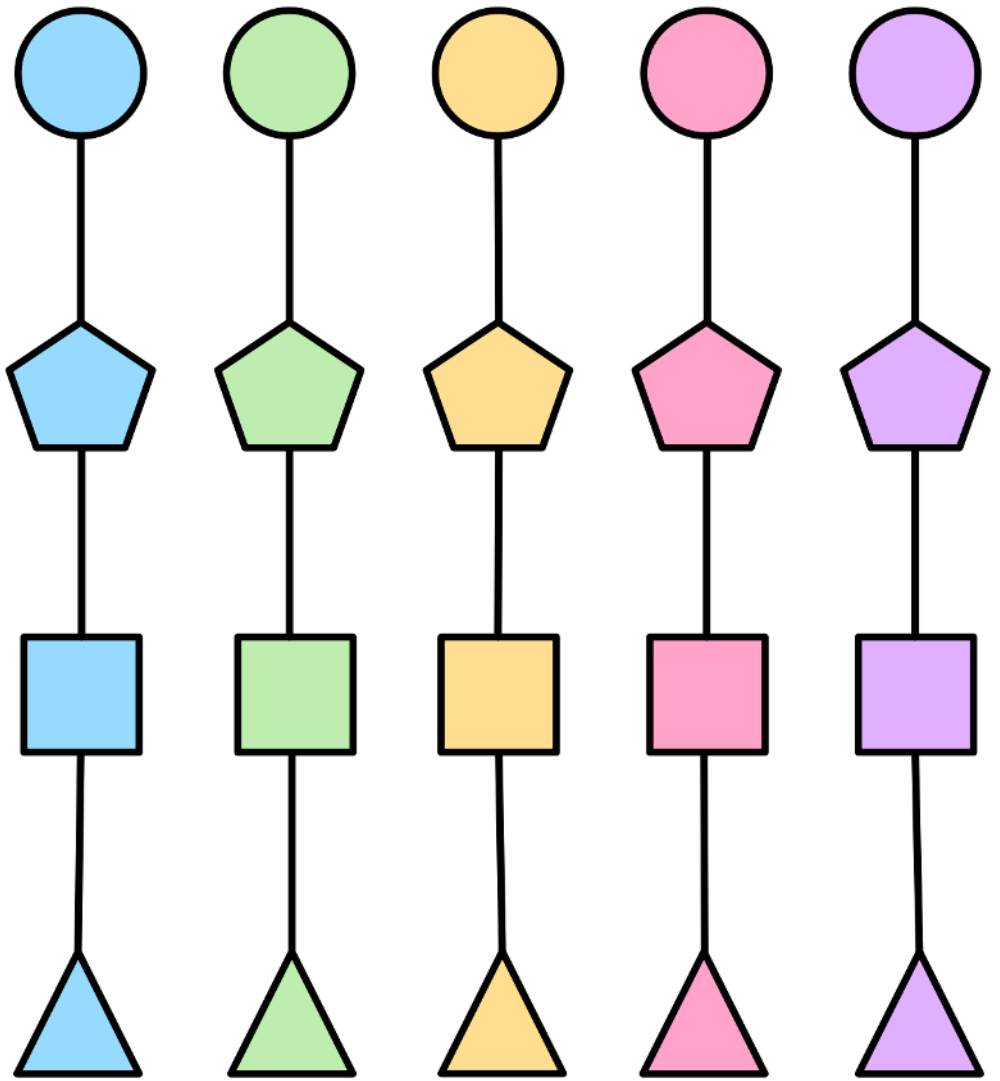
Aligning Structures with UX Flow ✅

User
Domains / Use Cases

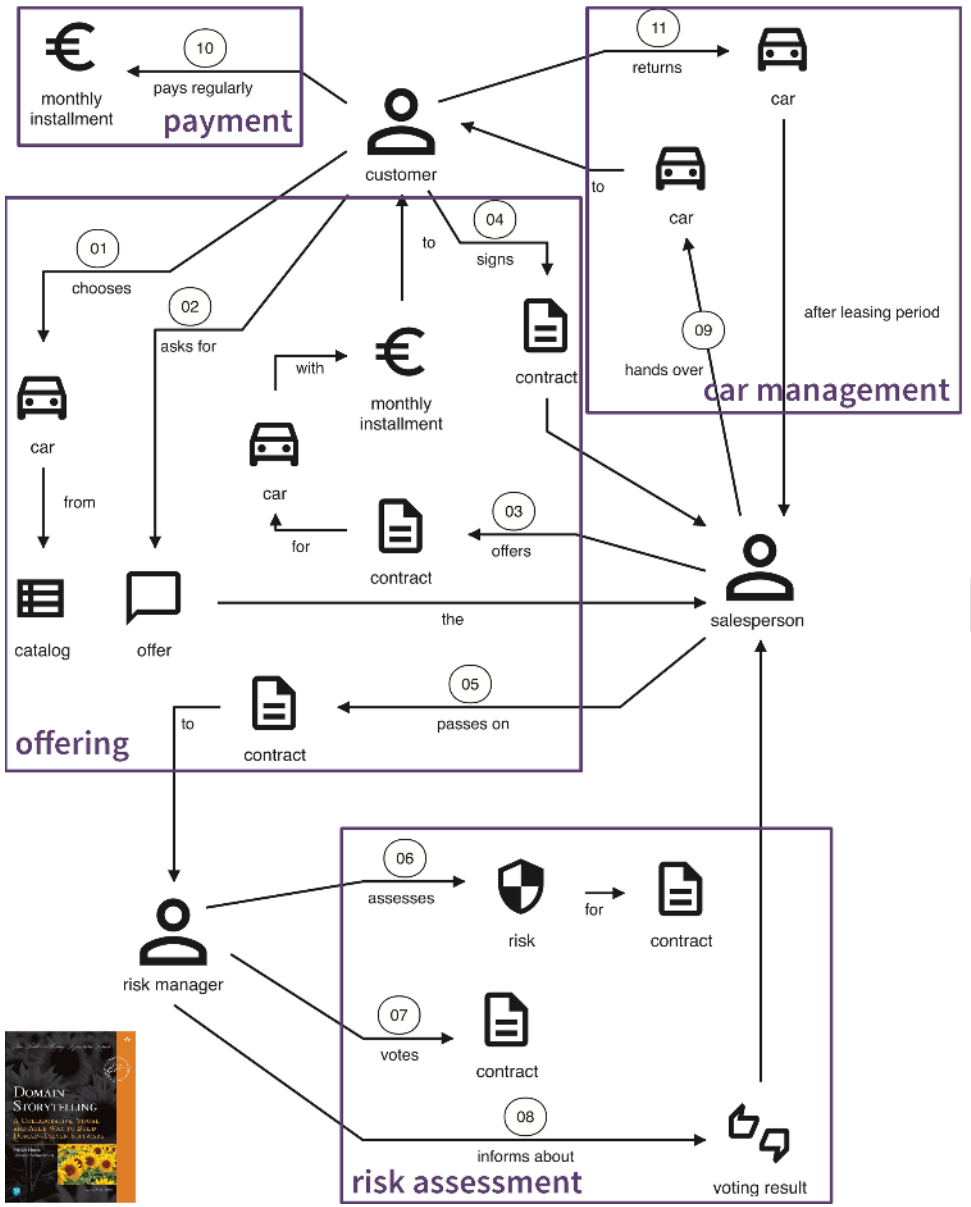
Product
Information Architecture

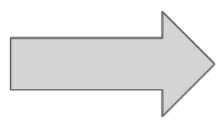
Teams
Organizational Structure

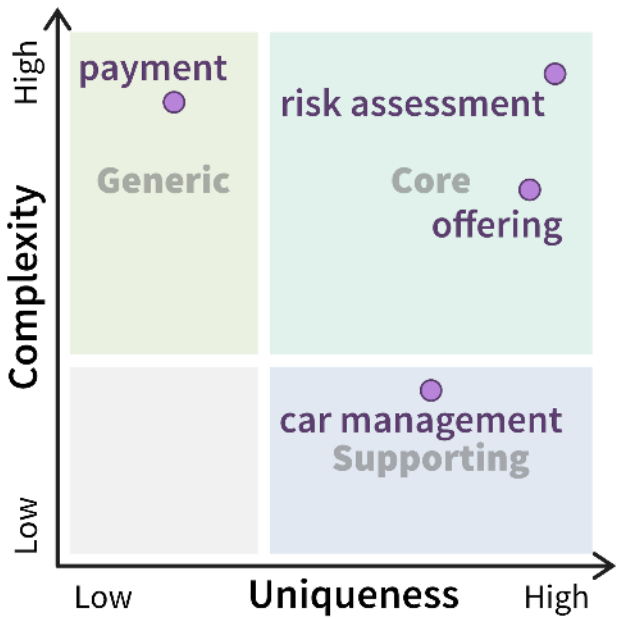
Code
Software Architecture

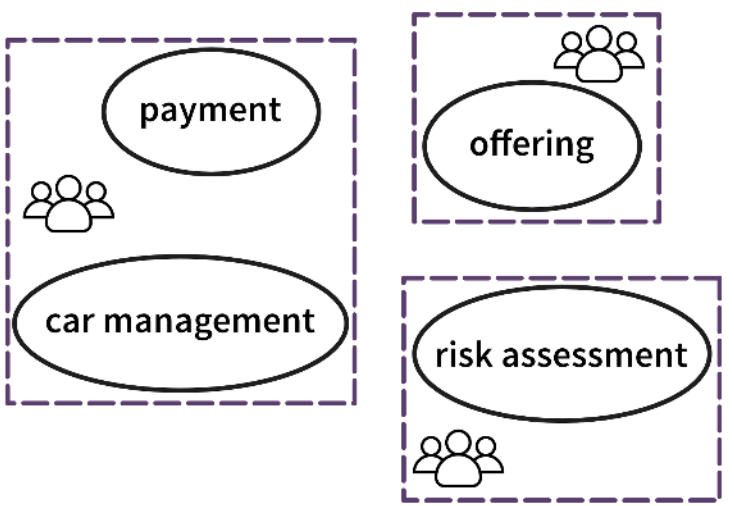


Now it's your turn!

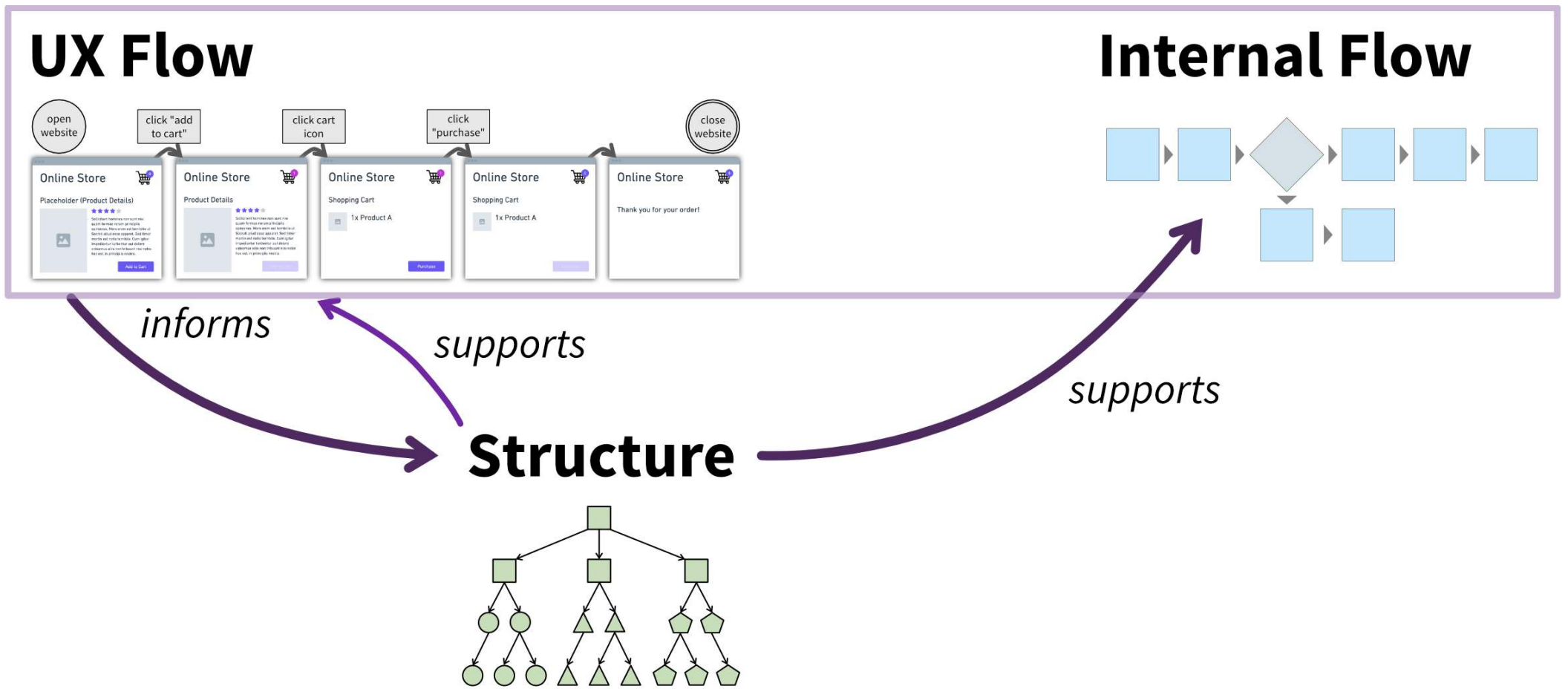








Refactoring ST Systems – Pt. 2: Flows



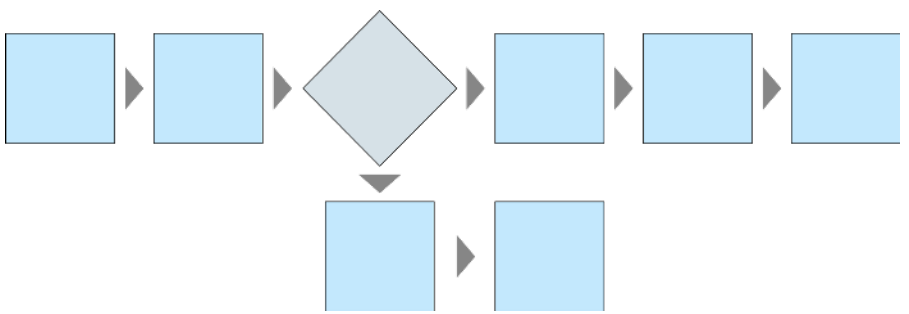
Flows worth looking at

- UX Flows:
  - Account creation & onboarding … and deletion
  - Purchase / subscription renewal & billing
- Internal Flows:
  - Onboarding of new developers
  - Deployment process (incl. feature flags, A/B tests)
  - Updating content for marketing campaigns
  - Database migrations
  - Rollback & disaster recovery

“Fix the Flow” Workshop

Preparation: Sketch the flow

- No need for formalities (like BPMN or UML)



- Talk to different people to get diverse perspectives

“Fix the Flow” Workshop – Step 1/6

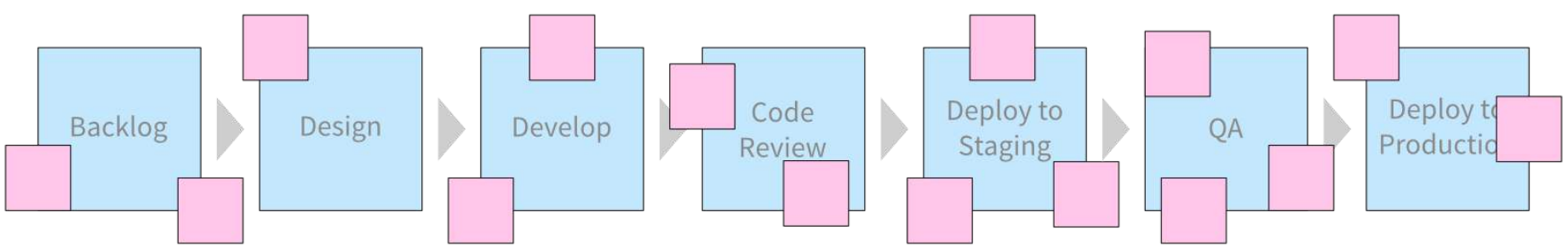
Align on the flow & target outcome:

- *How much time does this process take now? What’s the goal?*

“Fix the Flow” Workshop – Step 2/6

Brainstorm pain points:

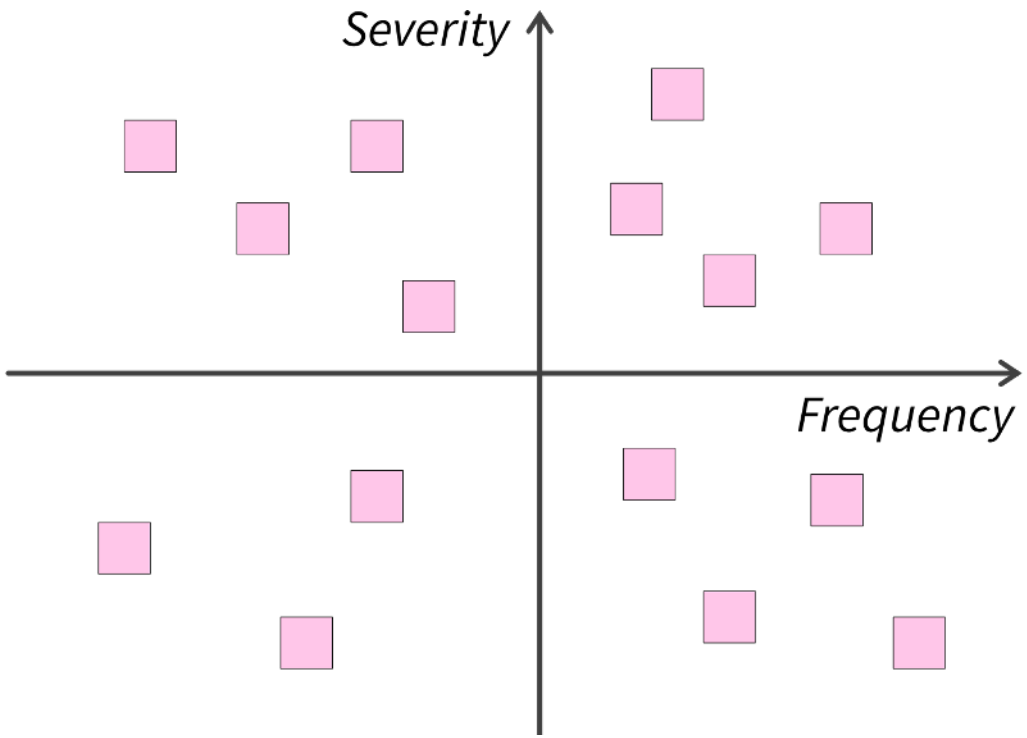
- *What is annoying?*
  (e.g., manual steps, dependencies, wait times, complexity)



“Fix the Flow” Workshop – Step 3/6

Prioritize pain points:

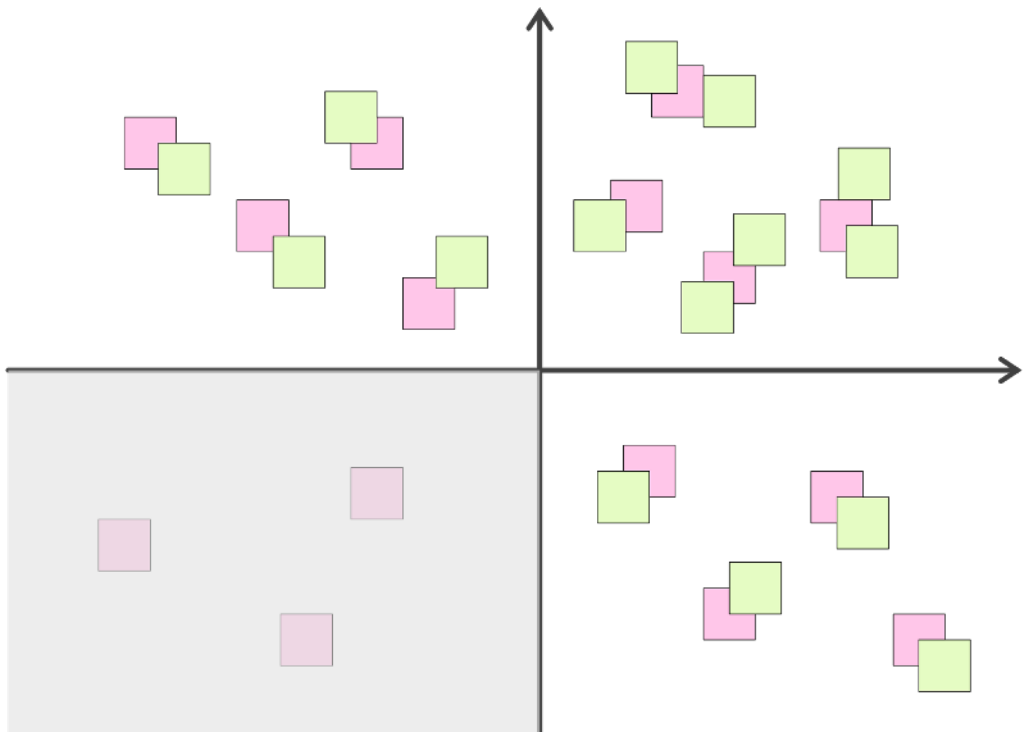
- *How long does it take?*
- *How often does it happen?*



“Fix the Flow” Workshop – Step 4/6

Brainstorm solutions:

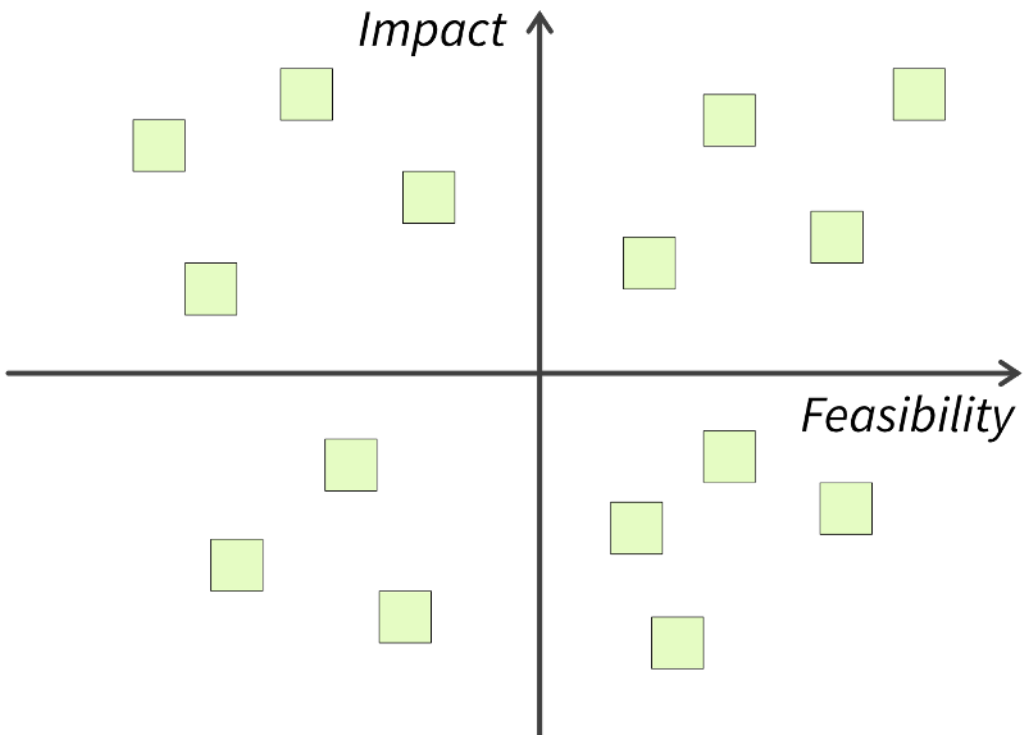
- *What can be improved?*
  - *simplify*
  - *automate*
  - *create self-service options*



“Fix the Flow” Workshop – Step 5/6

Prioritize solutions:

- *What’s the impact on our target outcome (e.g., reduced time)?*
- *How easy would it be to implement?*



“Fix the Flow” Workshop – Step 6/6

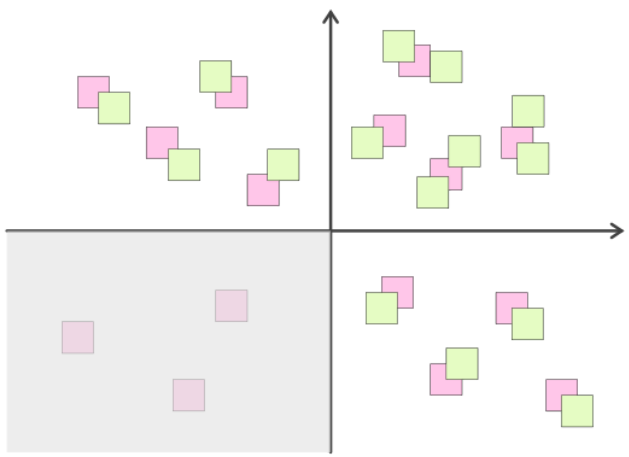
Make a plan:

Assign owners & deadlines (e.g., in Jira)

1. Owner (by Date)
2. Owner (by Date)
3. Owner (by Date)
4. Owner (by Date)

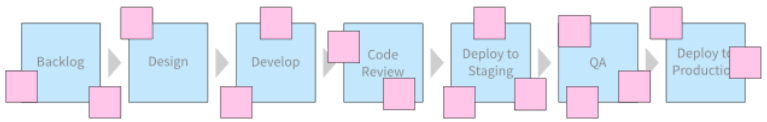
“Fix the Flow” Workshop – Summary

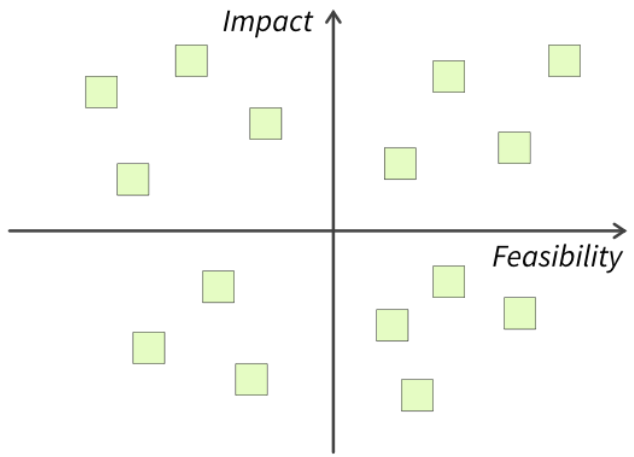
Step 1: Align on Flow & Target Outcome



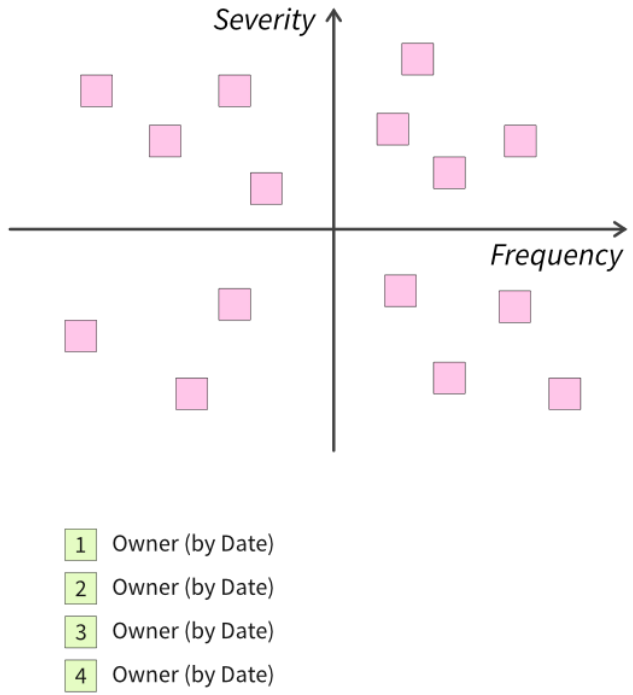
Step 2: Brainstorm Pain Points

(e.g., manual steps, dependencies, wait times, complexity)





Step 3: Prioritize Pain Points



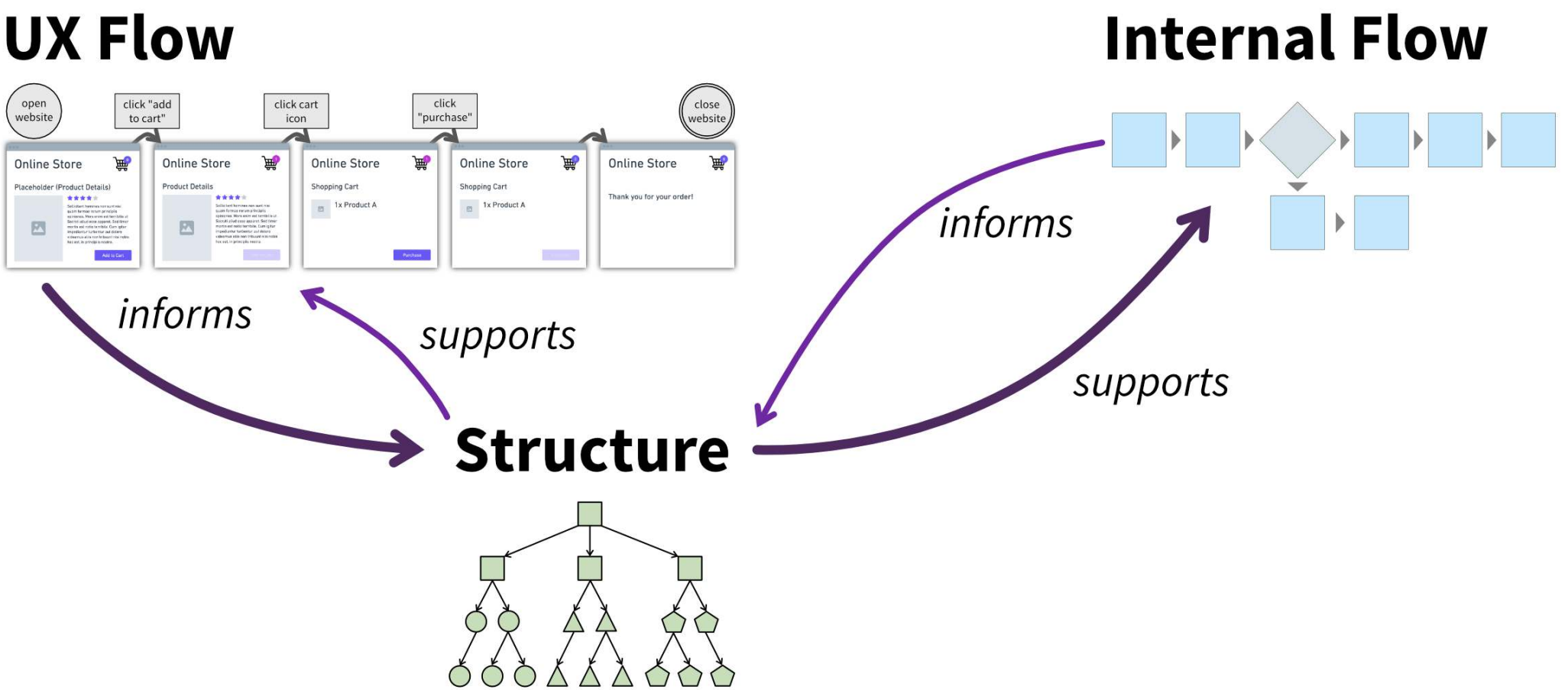
Step 4: Brainstorm Solutions

(e.g., simplify, automate, create self-service options)

Step 5: Prioritize Solutions

Step 6: Make a Plan

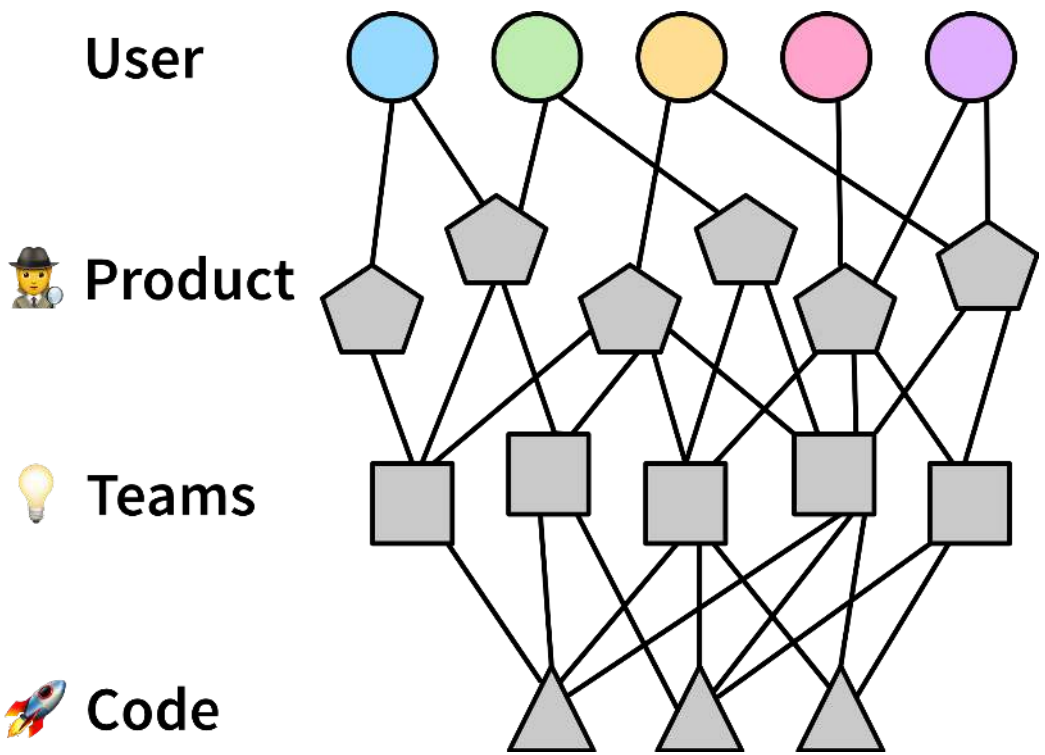
Refactoring ST Systems – Closed Loop



Need help? Hire me 🤓

Freelance Consulting Services:

1. **System Audit**
   Identify opportunities
2. **Help with refactoring**
   Hands-on, part of your team



Thank you for your attention!

Slides:

franziskahorn.de/resources.html

Connect:

linkedin.com/in/franziska-horn

Any questions?