

People vs Processes:

How to Survive, Influence, and Improve Quality

Alexander Ptakhin, Berlin

**A dark presentation
With slighter ending**

Agenda

Cases for studying

- Two teams: frontend & backend
- One QA between two cross-functional teams
- Communications chaos
- Pair programming

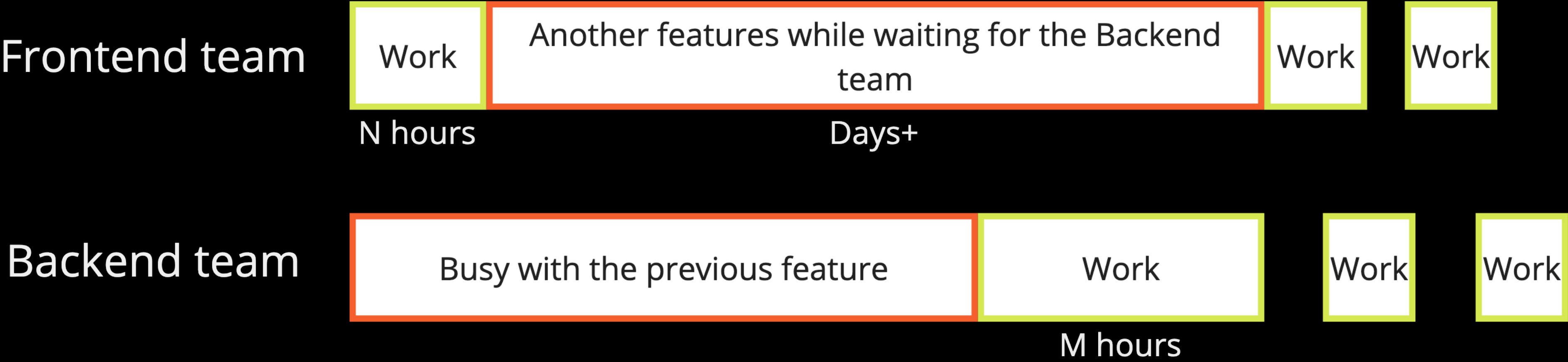
Two teams: frontend & backend

Two teams: frontend & backend

Bad advices

- At the beginning all looks good
- Both teams are in progress
- Frontend made things in advance
- Backend team is “slow”
- Changes are often deployed from both sides
- Client complains it's not working

Two teams: frontend & backend



Two teams: frontend & backend

Bad advices

- Unfinished work is a waste
- Unfinished from which perspective? The frontend card was planned, merged, and completed.
- From the user's perspective: "I can't use it." Does it mean it wasn't done?
- When our cards are finished, what exactly is happening?

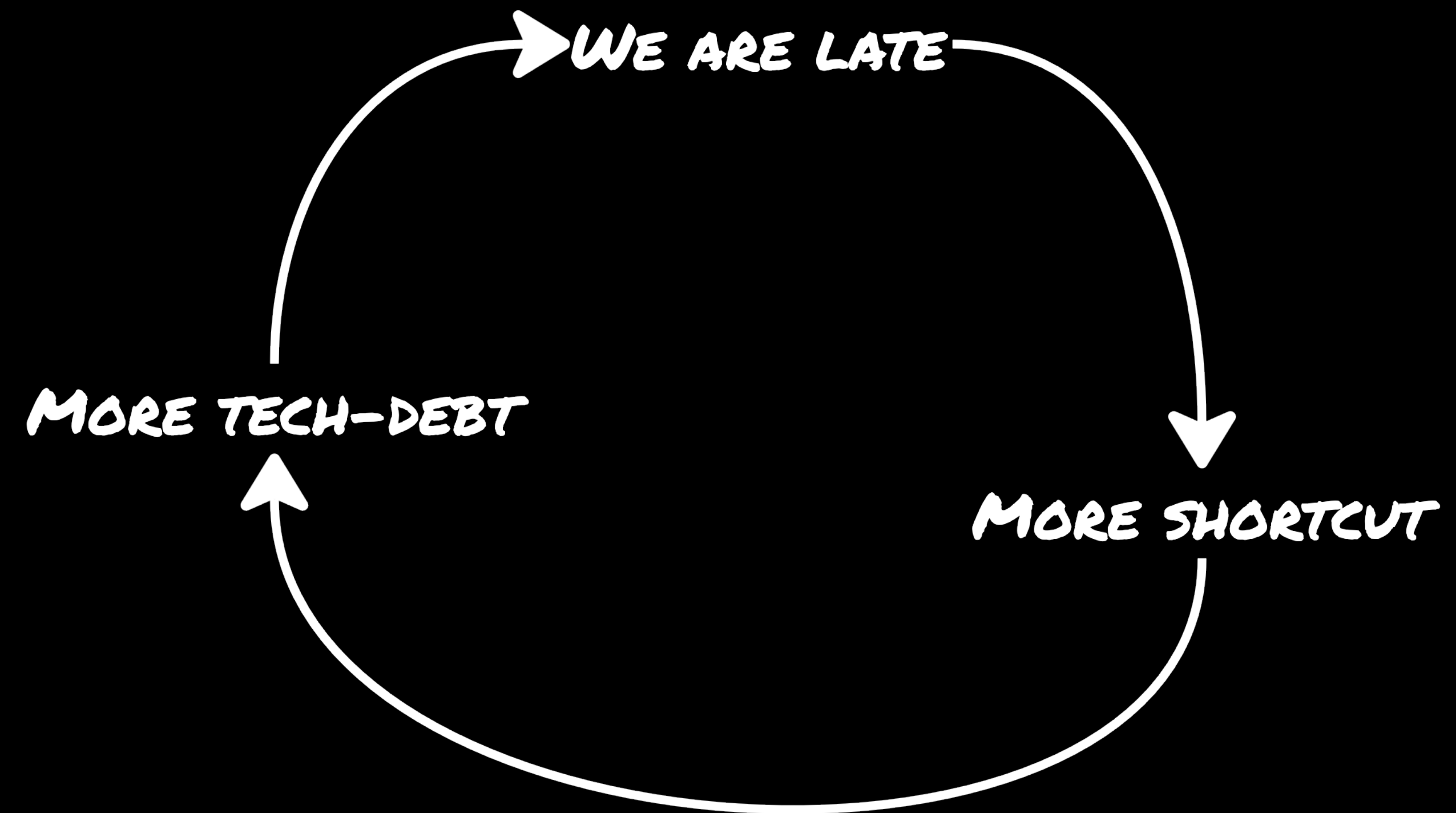
Two teams: frontend & backend

Bad consequences

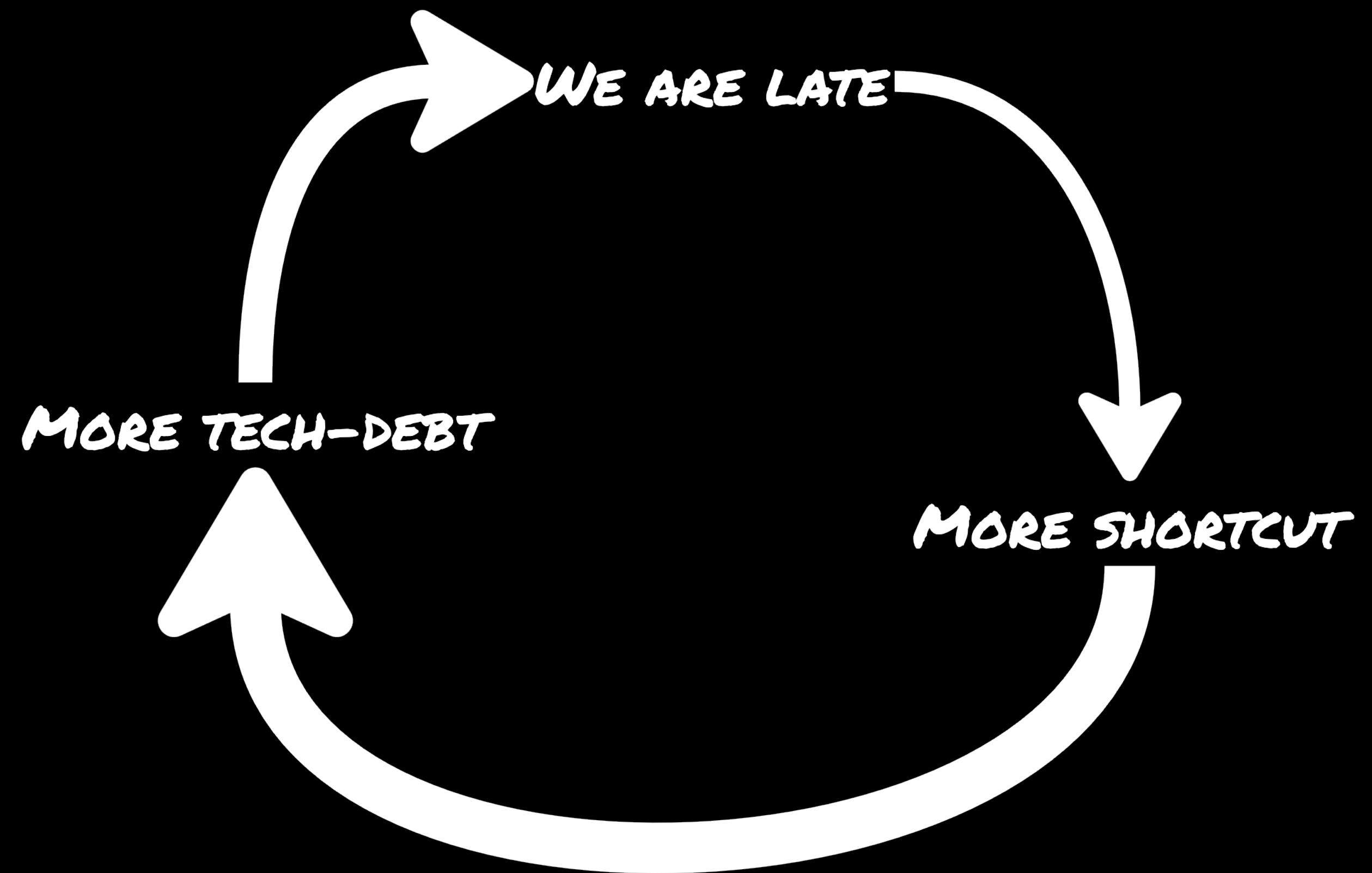
- The most challenging part wasn't adequately tested
- Come on! We are already late compared to the front end
- Let's do it fast now, then think
- It was tested. But wrong
- It wasn't working. Yes

Bad consequences

- The consequence of solving the issue was much bigger tech debt



Bad consequences



Two teams: frontend & backend

How to do better

- Sit together frontend/backend
- Just-in-time
- More sync
- Kanban, Queue theory

**One QA between two cross-
functional teams**

One QA between two cross-functional teams

Context

- Write e2e tests
- Resources are limited
- No one loved to get the task back
- There are no happy testers with big queues to test

One QA between two cross-functional teams

How to do better

- For teamleads: watch for returns count

One QA between two cross-functional teams

Team 1

Task A	Task B	Task C	Edit Task A	Release A&B
--------	--------	--------	-------------	-------------

Team 2

Task X	Task Y	Task Z	Task W	Edit Task X	Edit Task Y	Release X&Y
--------	--------	--------	--------	-------------	-------------	-------------

QA

Test Task A	Test Task B	Test Task X	Test Task Y	Test Task Z	Test Task W
-------------	-------------	-------------	-------------	-------------	-------------

Communications chaos

Communications chaos

Bad advices

- No one knows who is responsible for what
- A random person involved writes to a random person: agrees on new things

Communications chaos

How to do better

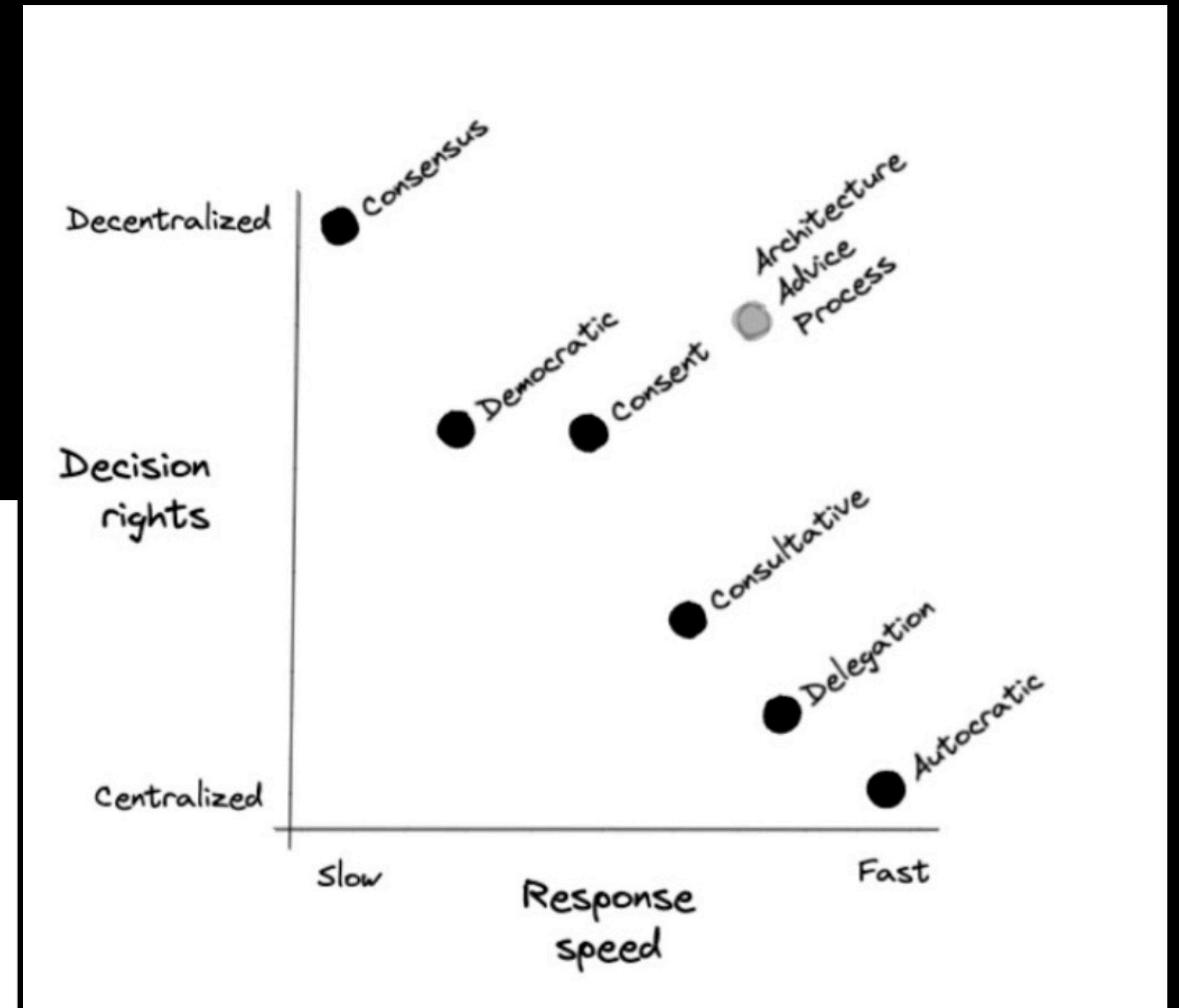
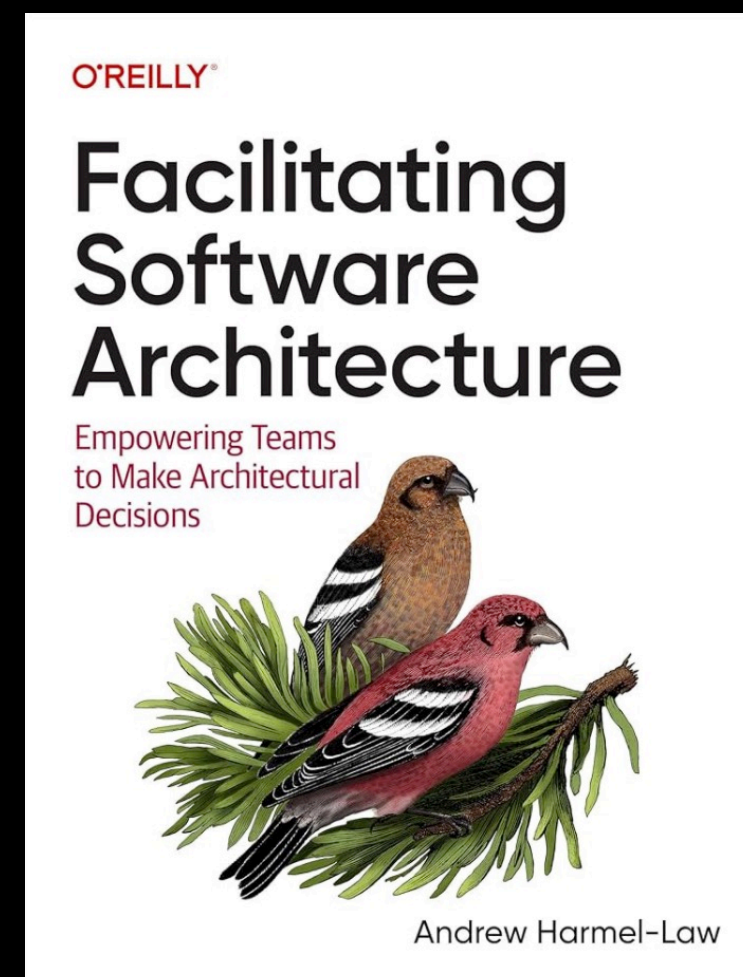
- For projects kick-off meetings
- Open channels to join to tag people if any problems
- Too many requests. Why requests are happening? Let's solve the reason before
- Someone on duty, who is leading the issue resolving
- Assignee is the person leading the story, not the only implementor

Pair programming

Pair programming

Bad advices

- Developers love shiny technologies
- Leads love shiny processes
- So we push processes



Pair programming

How to do better

- Start with the problem: lack of quality, learning, software performance, long development time

Recommendations

- Don't drag practices
- Start with problems and guide

Main outcome

- If we are on linear position qa or developer, we can bring quality locally without asking or at least awareness
- If we are team leads, we can think of processes slowing us down

Thank you for your attention

Questions?

- Slides link
- Kanban guide
- Dragan Stepanović - Async Code Reviews Are Choking Your Company's Throughput

