

Platform
Engineering Day
NORTH AMERICA

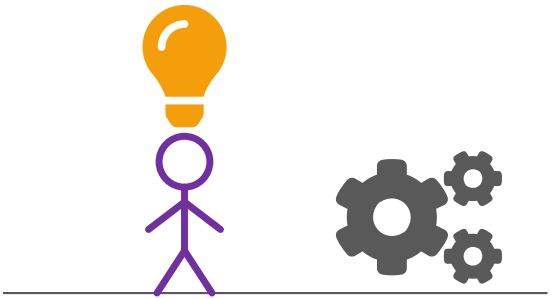
The Easiest Path Wins

Earning Platform Adoption
with DDD and Product Thinking

Fabrizio Lazzaretti, Patrick Müller

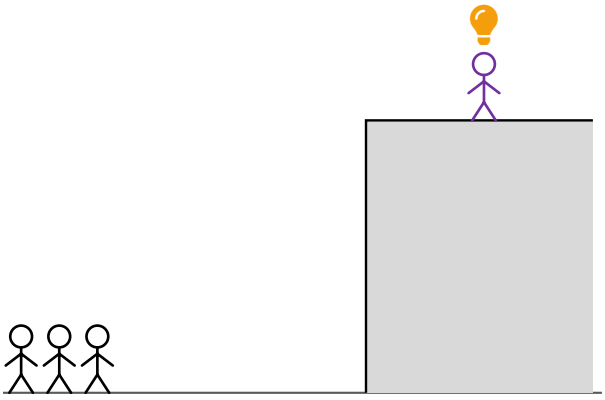
The idea

- You *think* you have a great idea
- You build something that makes others' work easier, as you see it



The cliff

- Golden paths nobody walks
- The learning curve is too high to climb
- Mandated tools get routed around
- No one gets or likes what you provide



Who we are

Platform Engineering Day NORTH AMERICA





Fabrizio Lazzaretti
Senior Managing Consultant, Wavestone
CNCF Ambassador
Co-author of Crafting Great APIs with DDD

Patrick Müller
Make IT Work, Make IT Fly
patrick-m-sc.de

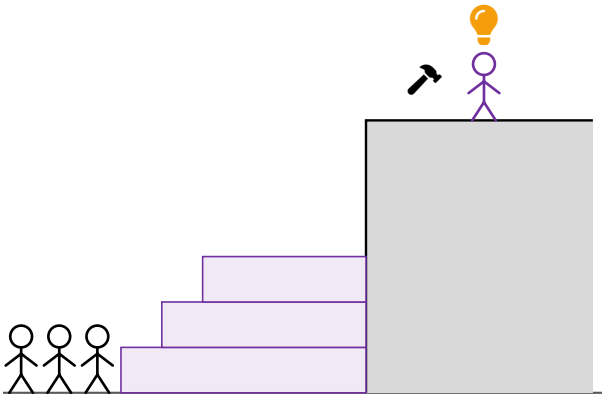
We've seen lots of these cliffs, internal and external, helped to solve them and to set up platforms properly

Why platforms fail

- **Mandate, not merit:** mandates are good for guardrails, not as a substitute for quality
- **Everything for everybody:** so nothing is perfect for anybody
- **Build once, improve never:** a platform is a living product, not a delivery
- **The wrong language:** not how cool your tech is, but the use case and its benefit
- **Wrong KPIs:** they kill even the best platform: onboarded teams, not satisfied teams

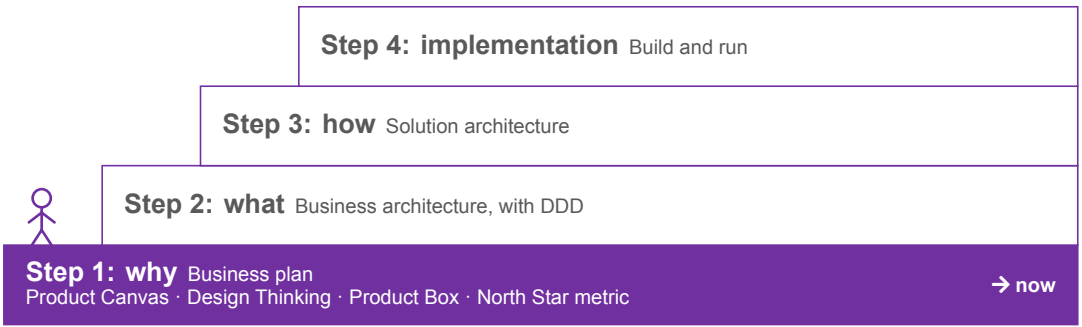
Build the stairs





How do we lower the entrance barrier and improve the customer experience, and thus be finally successful?

From value to build



Product Canvas

1 Vision

The why, how, what pitch



2 Goals

What do we want to do?

3 Key metrics

How do we measure, and measure again?

4 Customers

Who? Not everybody!

5 Unique value

Why do we need to build it?

6 Customer need

Their current pain points

7 Key resources

What is needed?

8 Channels

Marketing: how do we reach them?

9 Revenue streams

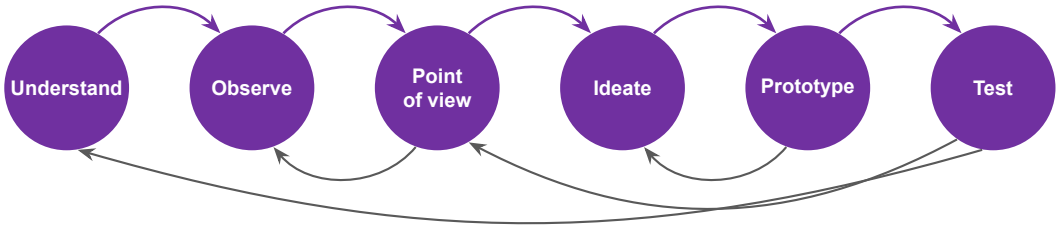
How do we get money for the product?

10 Cost structure

The costs to run the platform

Design Thinking

- What does the process you simplify really need?
- Find the side benefits and quick wins, pull them into the canvas

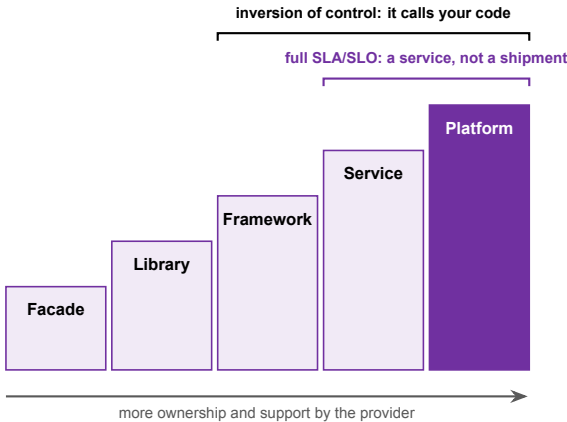


Widen your viewpoint: look at the greater context with the eyes of your customers, and test it

Do we need a platform?

“We need a platform.”

Why? Everyone’s building one.
Solving what? Unclear.
At which cost? (Return on investment)



From value to build



Platform Engineering Day NORTH AMERICA

Step 4: implementation Build and run

Step 3: how Solution architecture

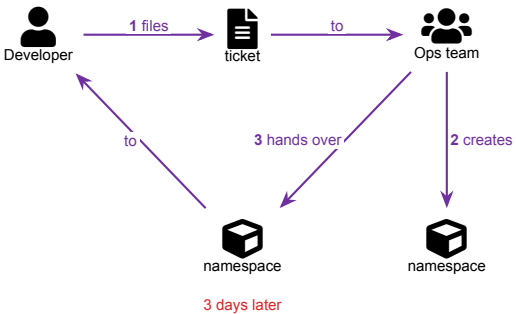
Step 2: what Business architecture, with DDD
Domain Storytelling · EventStorming · Bounded contexts · Requirements management
→ now

Step 1: why Business plan
Product Canvas · Design Thinking · Product Box · North Star metric
✔ done

Domain Storytelling

Domain-Driven Design

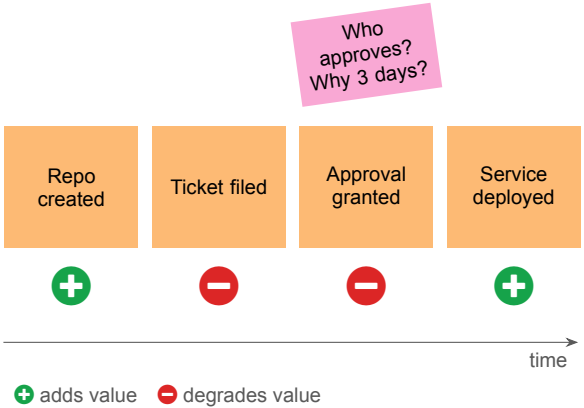
- The real workflow, not the one management knows
- Include the “other” stakeholders: consultants, externals
- Use it to specify and detail your use cases, and get a domain language



Do we all have the same understanding?

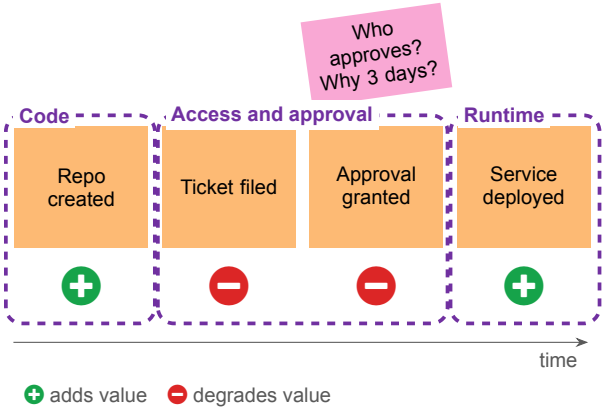
EventStorming

- What happens exactly?
- What can go wrong?
- Does each event add or degrade value?



Bounded contexts

- Where the language changes, draw the line
- One context, one team, one API
- Inside a context, let AI build: spec-driven development



Requirements management



Quality requirements: needed for the SLA/SLO

Change management: how and when do we change, and when is something removed?

Prioritization: what comes first

→ **This keeps the platform alive for long: no feature creep, the quality the customer needs**

→ **The base for test management**

From value to build



Platform Engineering Day NORTH AMERICA



Step 4: implementation Build and run

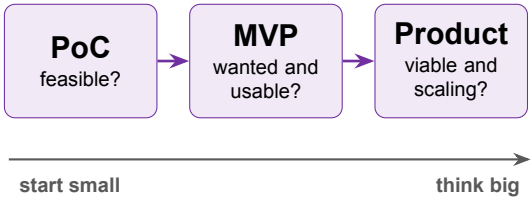
Step 3: how Solution architecture
Think big, start small · Beta users · Kotter's 8 Steps
→ now

Step 2: what Business architecture, with DDD
Domain Storytelling · EventStorming · Bounded contexts · Requirements management
✔ done

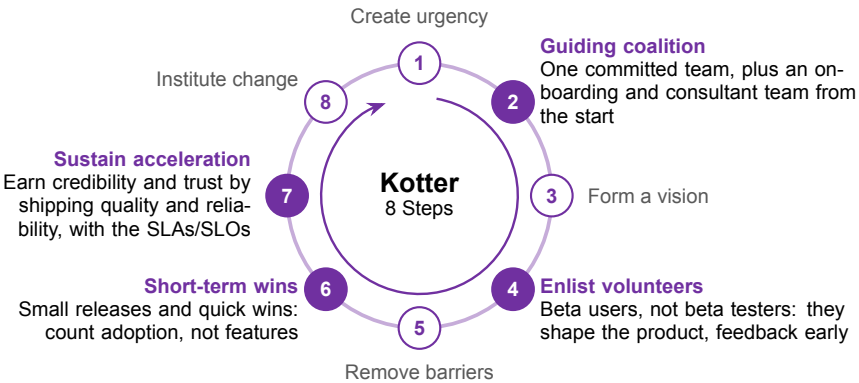
Step 1: why Business plan
Product Canvas · Design Thinking · Product Box · North Star metric
✔ done

Think big, start small

- How do we build a roadmap?
- Start with an MVP of the target vision
  - Helps you keep track of your purpose: stops feature creep
  - Spans the vision, the goals and your use cases, down to the implementation



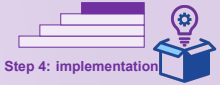
The team, beta users and the change



Have a customer and stakeholder management in place and running

Talk about the product: internal platforms need marketing too!

From value to build



Platform Engineering Day NORTH AMERICA



Step 4: implementation Build and run
API first, GUI on top · Wasm, open-closed · Lifecycle, FinOps
→ now

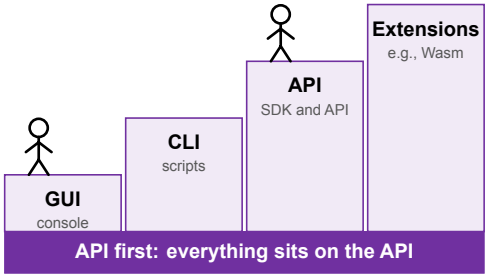
Step 3: how Solution architecture
Think big, start small · Beta users · Kotter's 8 Steps
✔ done

Step 2: what Business architecture, with DDD
Domain Storytelling · EventStorming · Bounded contexts · Requirements management
✔ done

Step 1: why Business plan
Product Canvas · Design Thinking · Product Box · North Star metric
✔ done

A step for every level

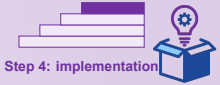
- API first: one API per bounded context, in the ubiquitous language
- GUI and CLI on top, as the beginner steps, like AWS*
- It scales, keeps people on the platform, and they can extend it, e.g., with Wasm
- Keep it simple to start, like Lego: child's play!
- The demo needs to be simple!



* The concept: console, CLI and SDKs call the same service APIs. Not every console feature uses a public API.

But who cares? We have AI!

And AI?



- AI churns code faster: APIs must be stable and well scoped
  → **cut along the bounded contexts**
- An agent monolith is still a monolith, MCP is still an API
- Let AI help your customers
  - AI-written extensions, safe in the Wasm sandbox
  - Good APIs: their agents get it right

What else is needed

- **Product Lifecycle Management:** introduce, grow and retire features
- **FinOps:** resource optimization and cost management
- **ITIL, IT4IT, EA:** next to them, not instead
- **Prove your value to management:** become a tactical or strategic asset
- **Self-financing:** so the platform doesn't die or stop improving

What you take home: frameworks and methods



Platform Engineering Day NORTH AMERICA



Step 4: implementation Build and run
API first, GUI on top · Wasm, open-closed · Lifecycle, FinOps

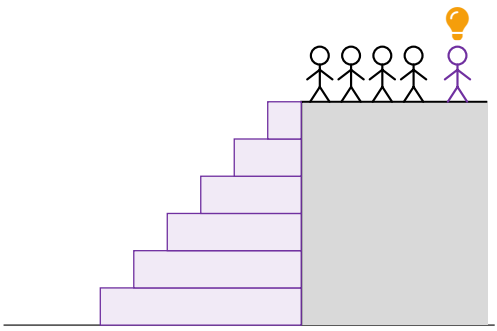
Step 3: how Solution architecture
Think big, start small · Beta users · Kotter's 8 Steps

Step 2: what Business architecture, with DDD
Domain Storytelling · EventStorming · Bounded contexts · Requirements management

Step 1: why Business plan
Product Canvas · Design Thinking · Product Box · North Star metric

The easiest path wins

- The platform takes work off its users, so their effort drops. *Not yours!*
- Know the struggles of your customers by heart, not the things you like to change
- Treat your customers like kings: be trustworthy and reliable



That makes it the easiest path, and the easiest path wins

Thank you!

Platform Engineering Day NORTH AMERICA

The Easiest Path Wins: Earning Platform Adoption with DDD and Product Thinking

https://lazzaretti.me/peday26



PRESENTER

Fabrizio Lazzaretti

Senior Managing Consultant
Wavestone · CNCF Ambassador



PRESENTER

Patrick Müller

Make IT Work, Make IT Fly
patrick-m-sc.de



Platform Engineering Day NORTH AMERICA