

Digital Transformation: Why well-planned multi-year programs still fail!

And what to do instead!

Anu Pandey

Venki Arunachalam



VMware Tanzu Labs

Tanzu Labs (formerly Pivotal Labs) is an autonomous digital consultancy in VMware.

Specialise in transformation through developing successful (profitable) custom software products.

Let's start with a trivia

What's common with these companies?

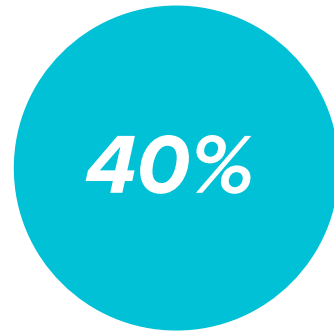


Building **successful digital solutions** is hard, so many organisations look at “**Digital Transformations**” as a means to become a modern **digital first** organisation.

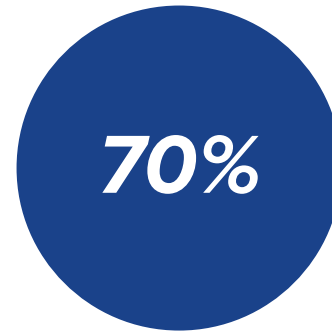
Some interesting facts about “Digital Transformations”



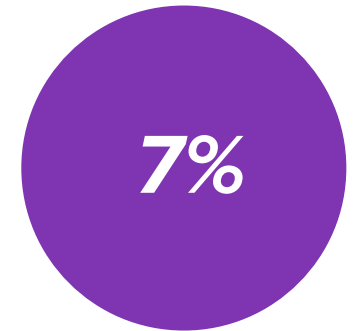
Organisations
have a digital
first business
strategy



Technology
spend on digital
transformation



Digital
transformations
fail



Organisations
claim to have
fully
implemented
their
transformation

Source: Forbes

A digital transformation story: case study

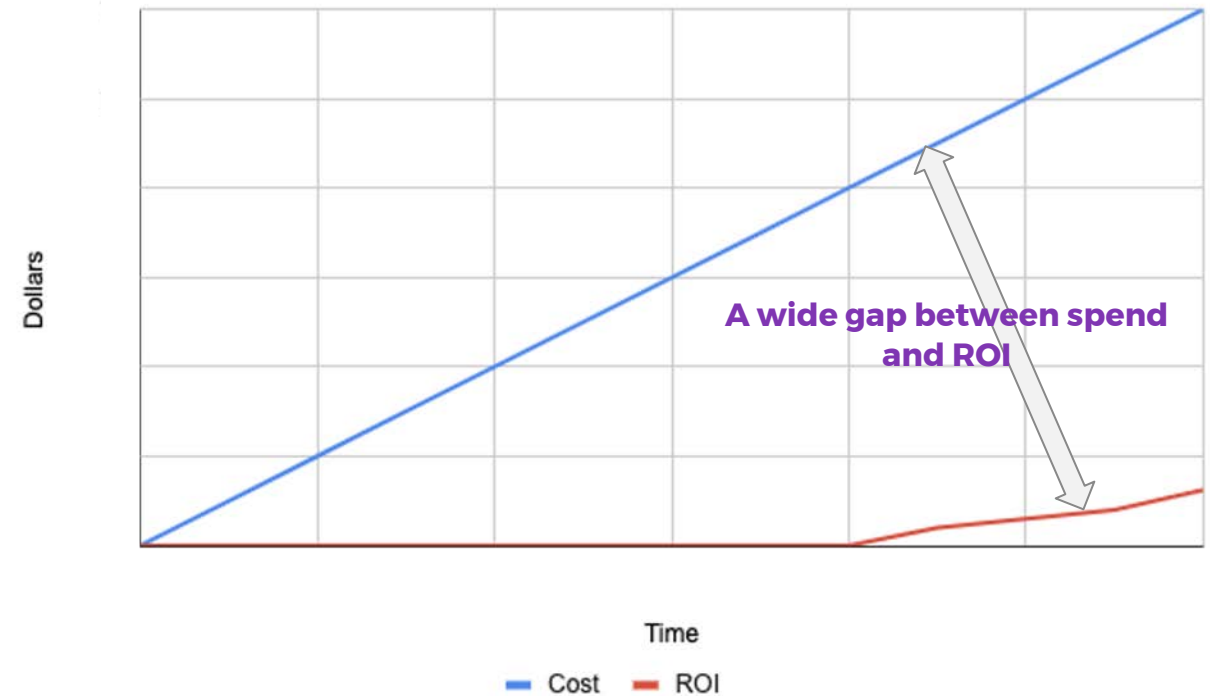
Become a modern digital organisation

By **modernising** and **consolidating** multiple legacy disparate platforms performing the same function to **reduce operation cost, drive efficiencies** and **gain competitive advantage**



Proverbial transformation checklist

- ✓ Program should have a right objective
- ✓ Executive support for the goals
- ✓ Approved Funding
- ✓ Strong Plan



3 key failure points in a Digital Transformation

Digital Transformation turns out to be an implementation of

“Process” (SAFe, Scrum, LeSS, DSDM)

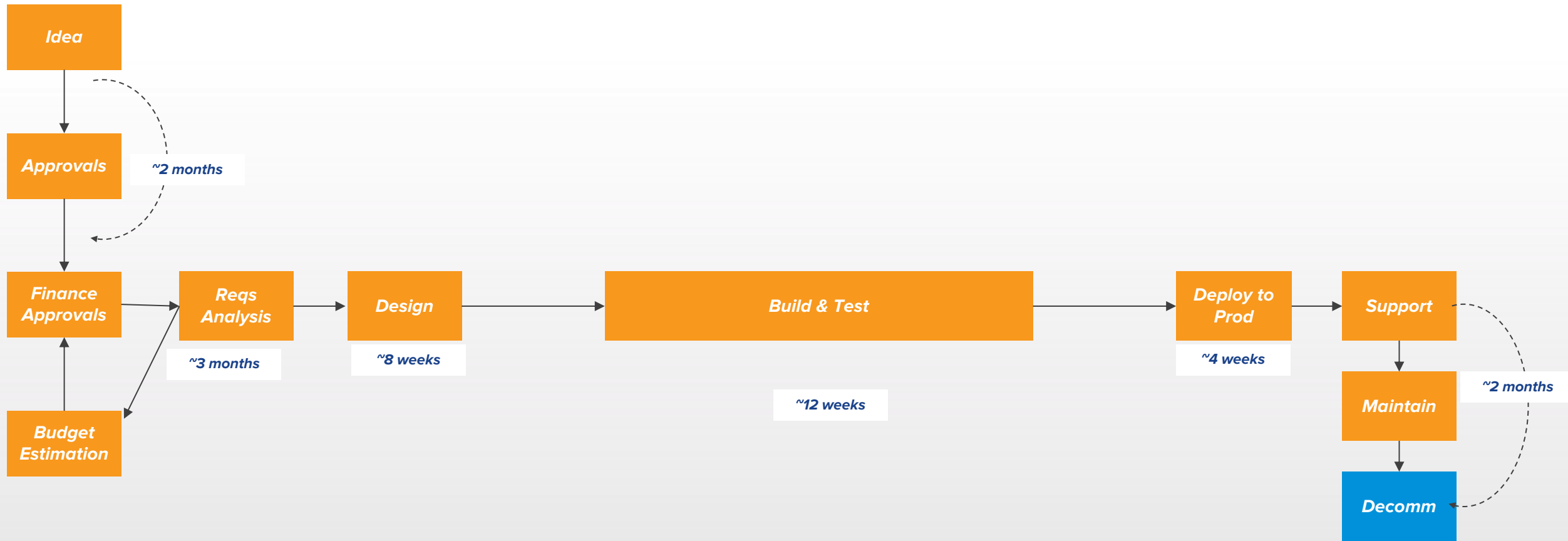
or

“Technology” (Cloud Migration, SAP, Salesforce, AEM)

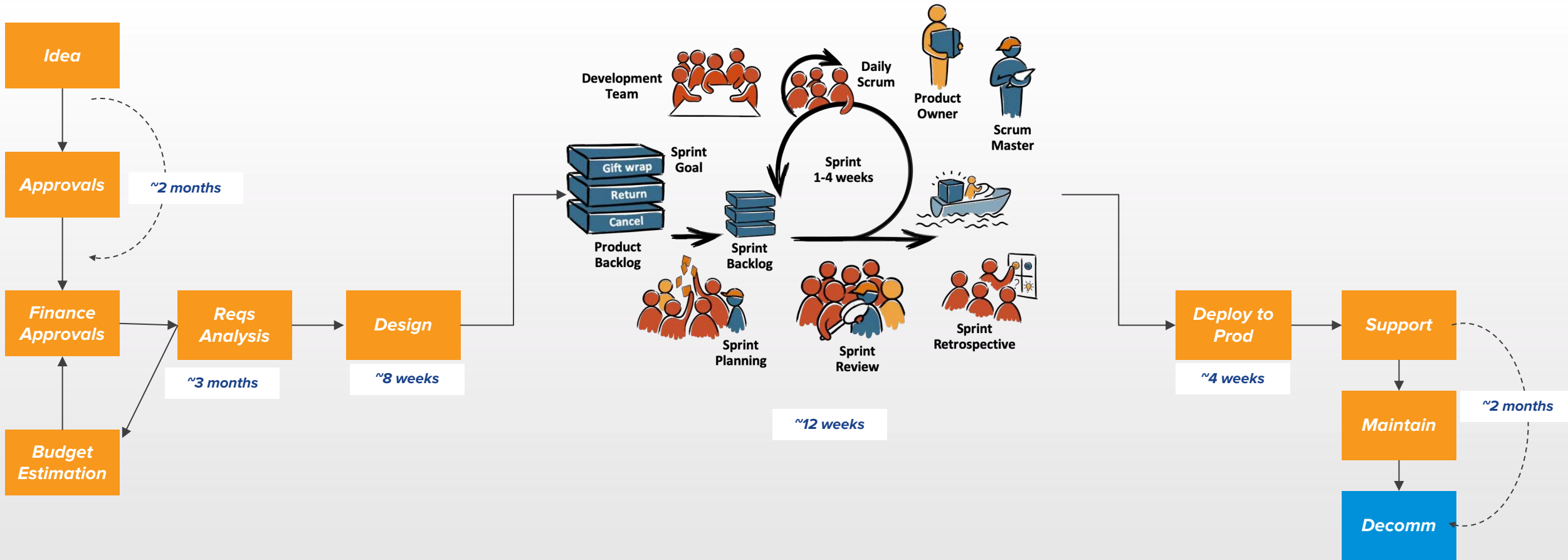


#1

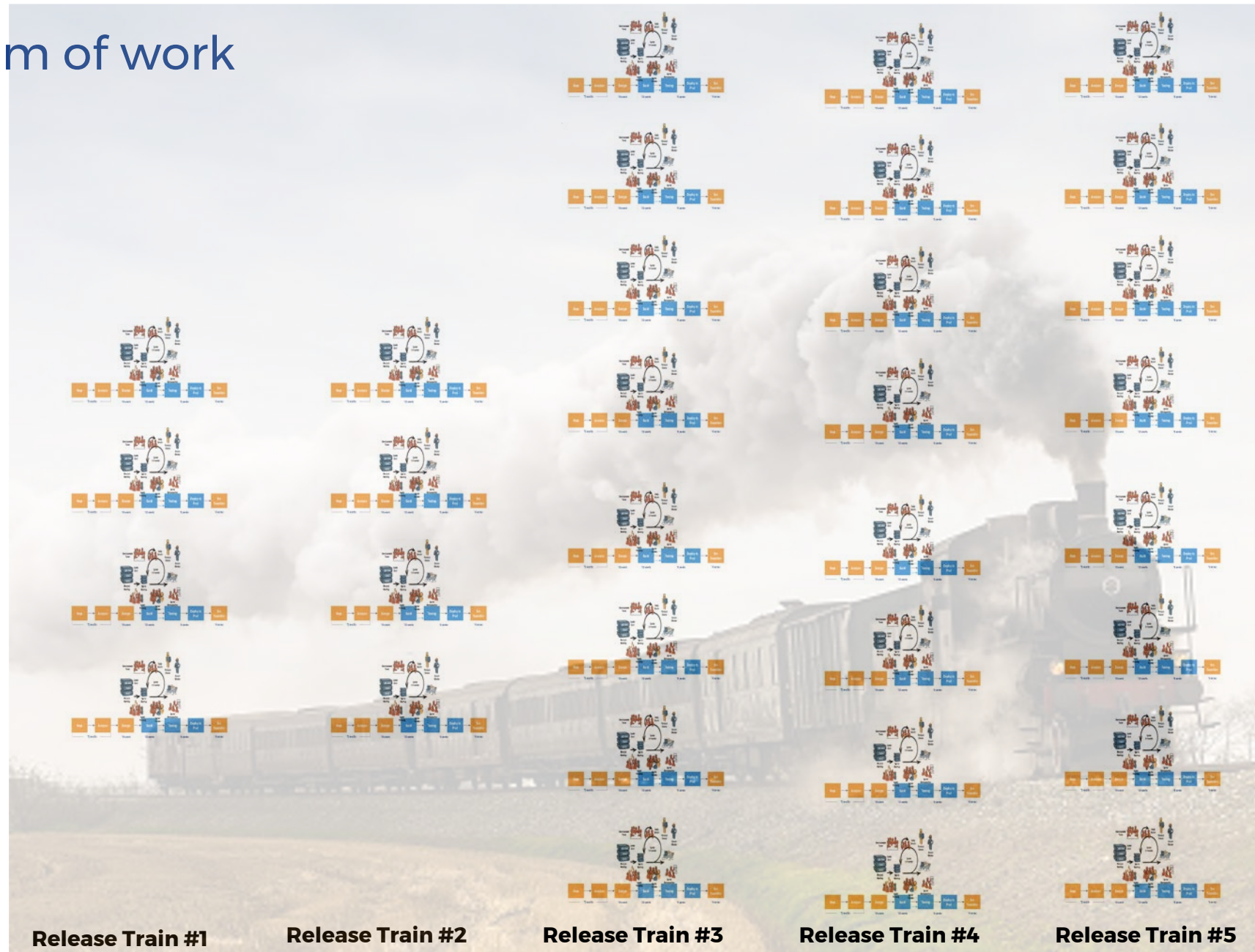
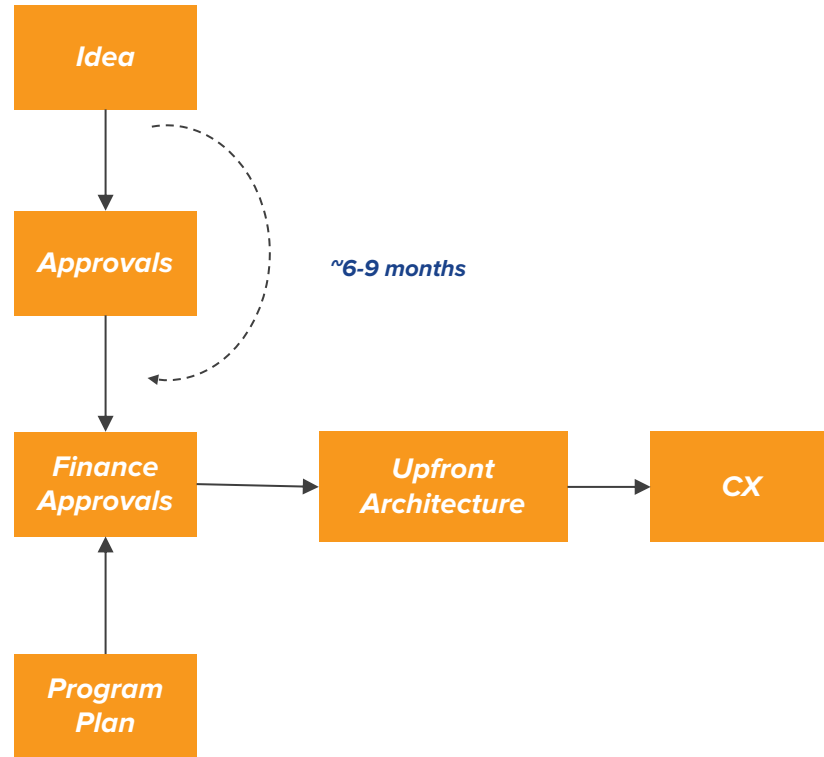
The development process was changed from



To the classic 'Water-Scrum-Fall'



And fitted into a program of work

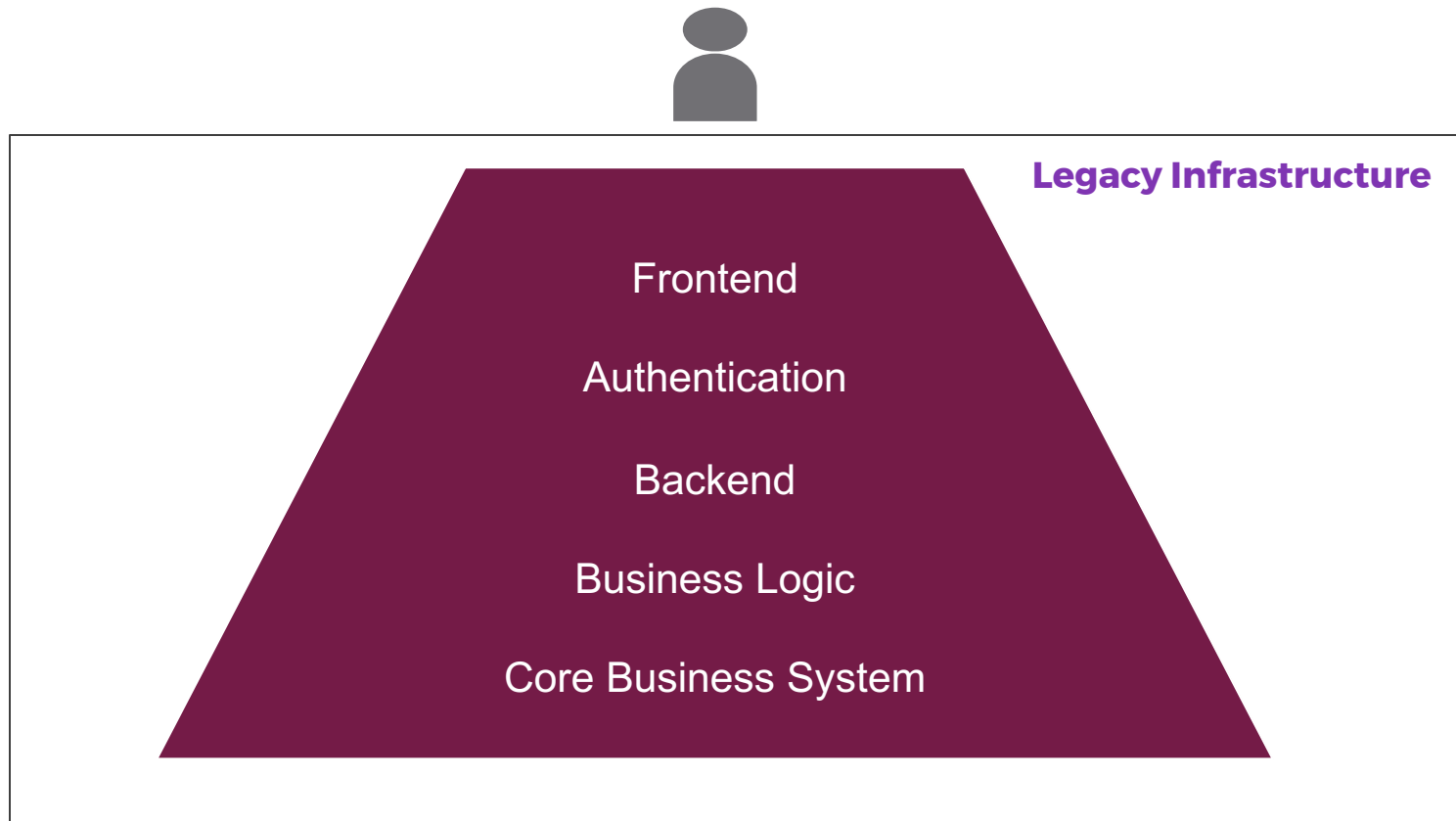


Architecture in Digital transformations
becomes a block diagram of
“reusable components”

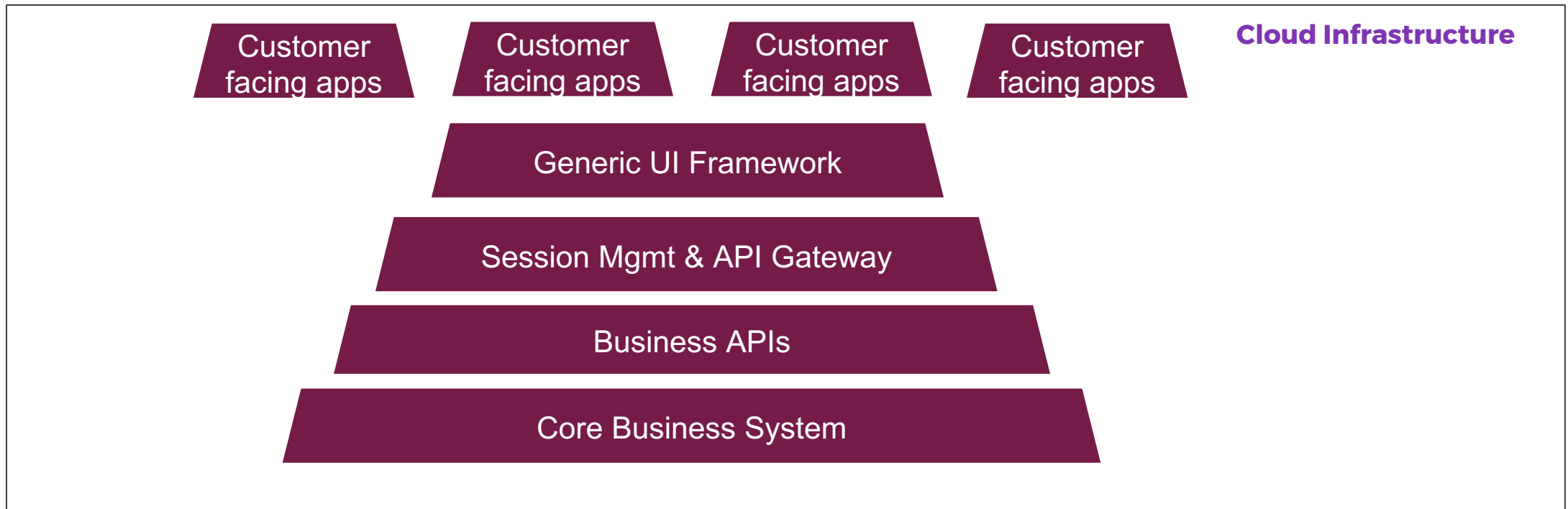


#2

The enterprise architecture changed from



To the “must have” microservices architecture

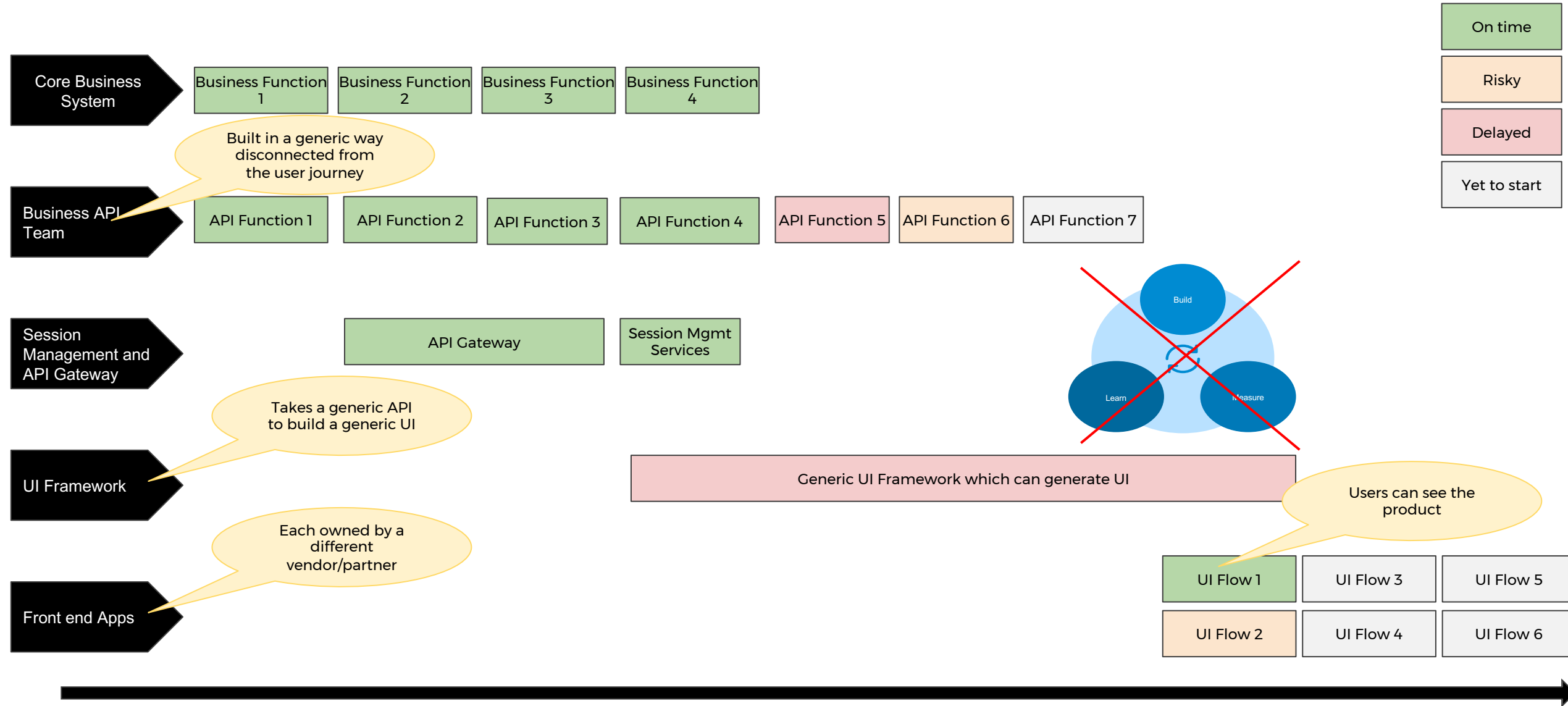


Cost of ownership of the Digital Assets **goes up**
over time without corresponding
“Return on Investment”



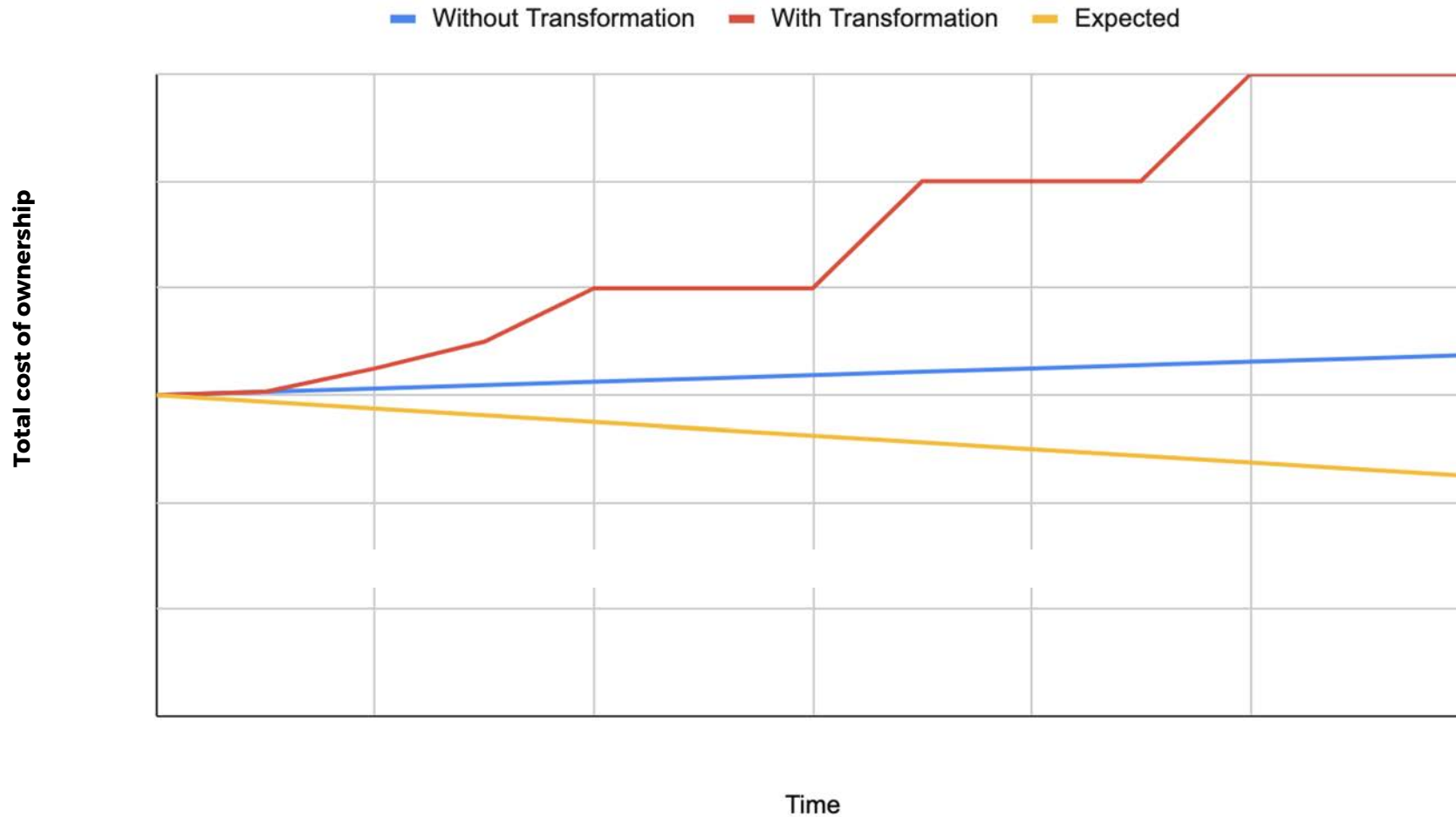
#3

The well planned 'Transformation Roadmap'



After doing this for 3 years•

The impact to the cost of ownership as part of the program



Business outcomes weren't met, though almost all of the projects were delivered on time

The multi-year program has failed.
What to do instead?

If an organisation want to be digital first, doesn't it make sense to start with building digital products?

What's common amongst the successful digital organisations



What does being product centric mean?

Teams are organised around product(s) with each product being an independent solution to a customer problem

Manifesto for to become a product centric organisation

Return on Investment over Project Delivery

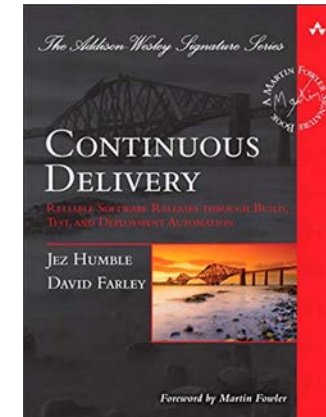
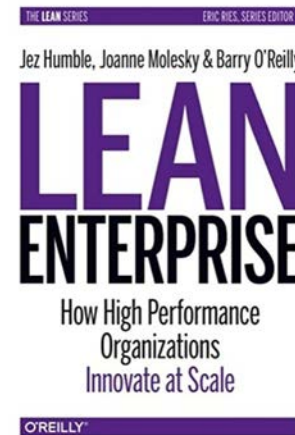
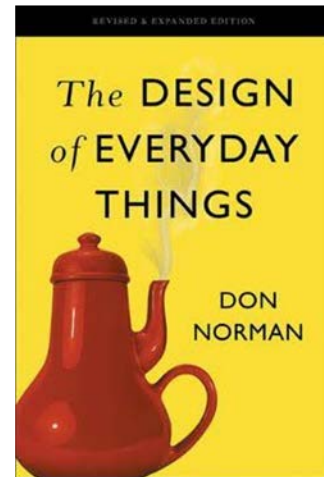
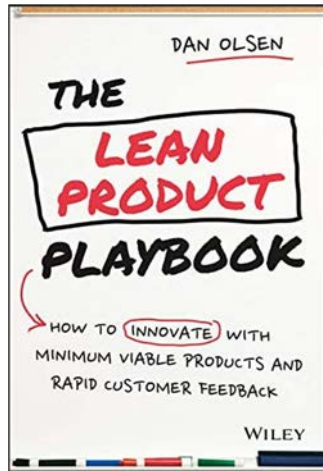
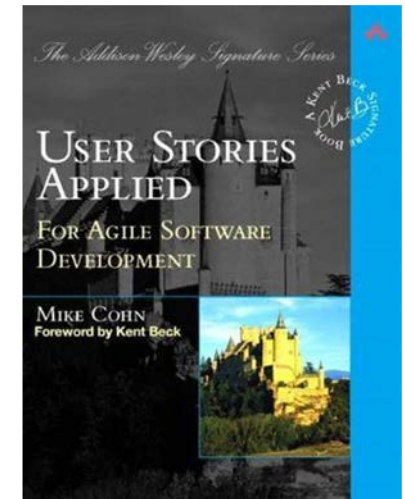
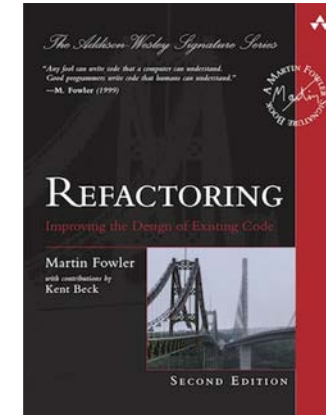
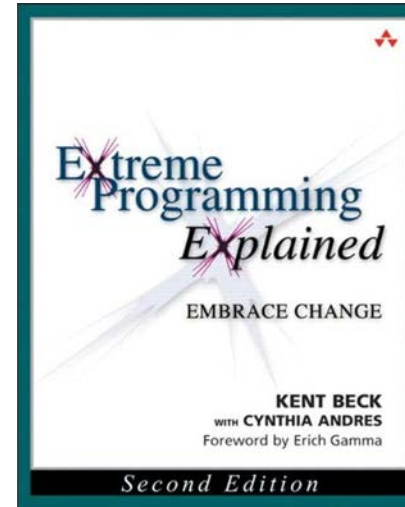
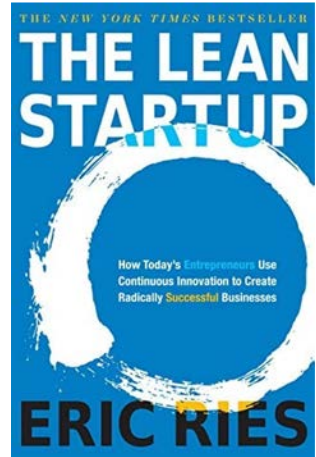
Use over Reuse

Customer Success over Process

Internalise IP over Total Outsourcing

Approach

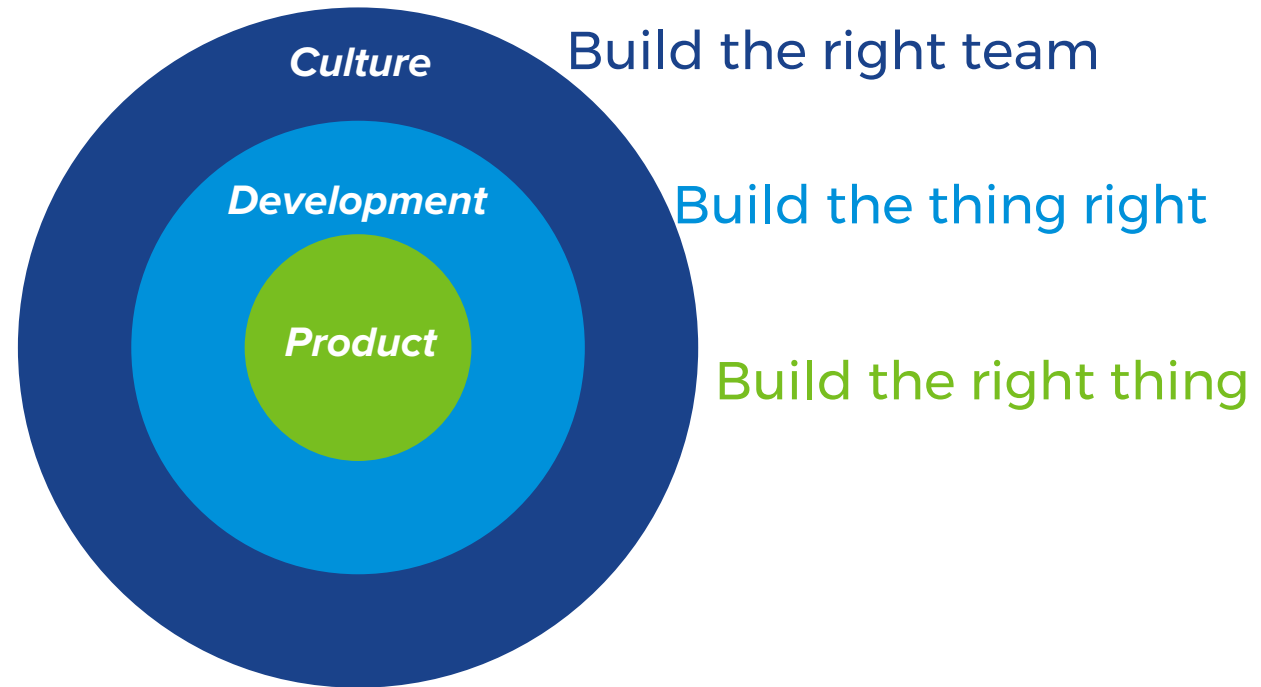
Theory is everywhere



The secret sauce: From theory to practice

The secret sauce - become a learning organisation

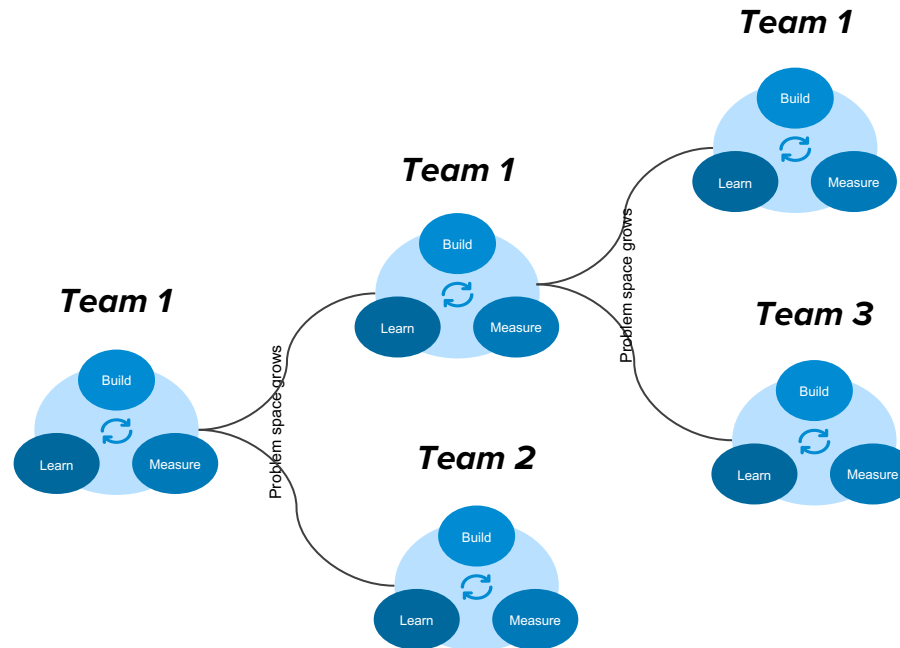
Step 1: Create a successful product team as an atomic unit which models the future organisation



Benefit: Create a recipe which suits your organisation to become successful: fail fast and small.

The secret sauce - become a learning organisation

Step 2: Expand the methods to the whole organisation by seeding teams and use technology to enable

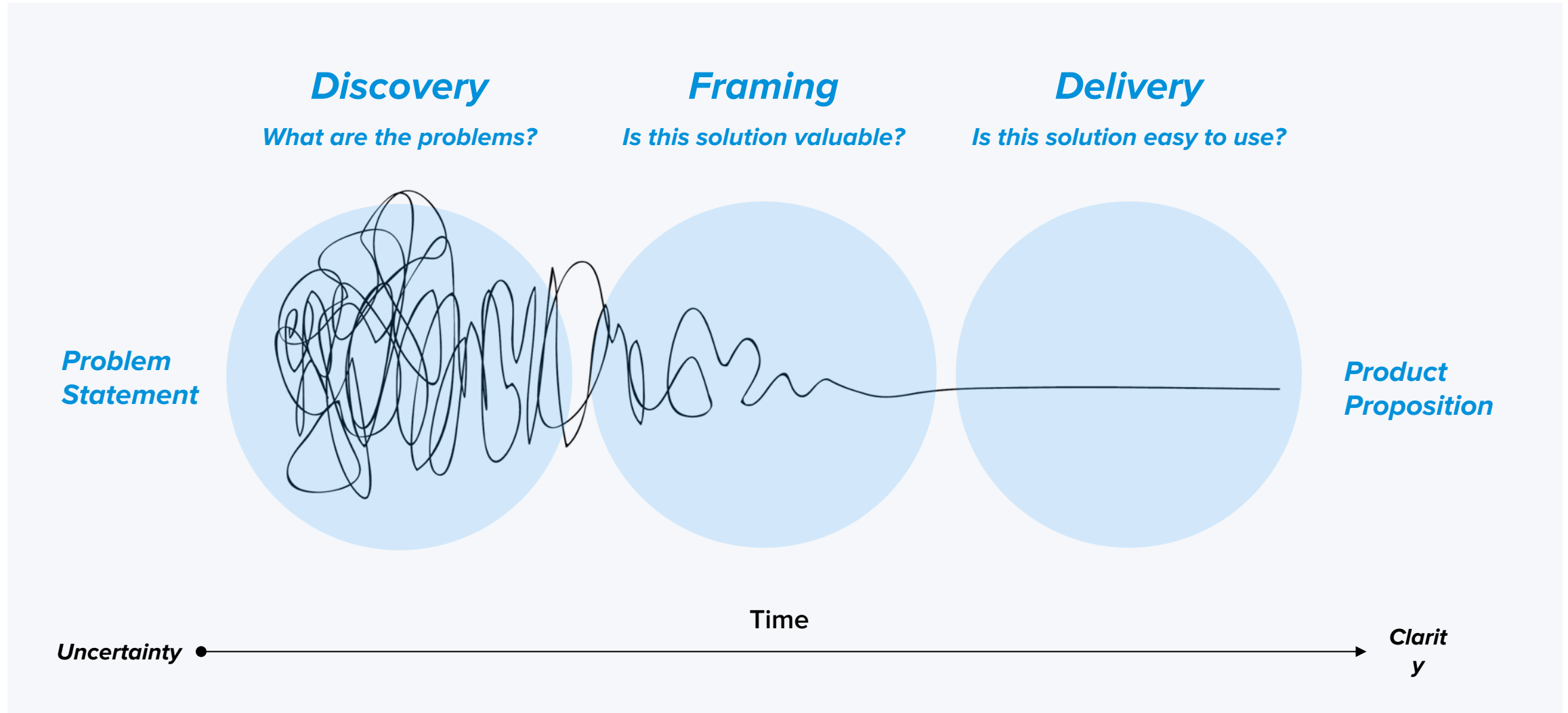


Benefit: Sustainable pace, internalised IP, ROI at each step

Product - Build the right thing

- ★ Viable
- ★ Desirable

Find the right problem - solution



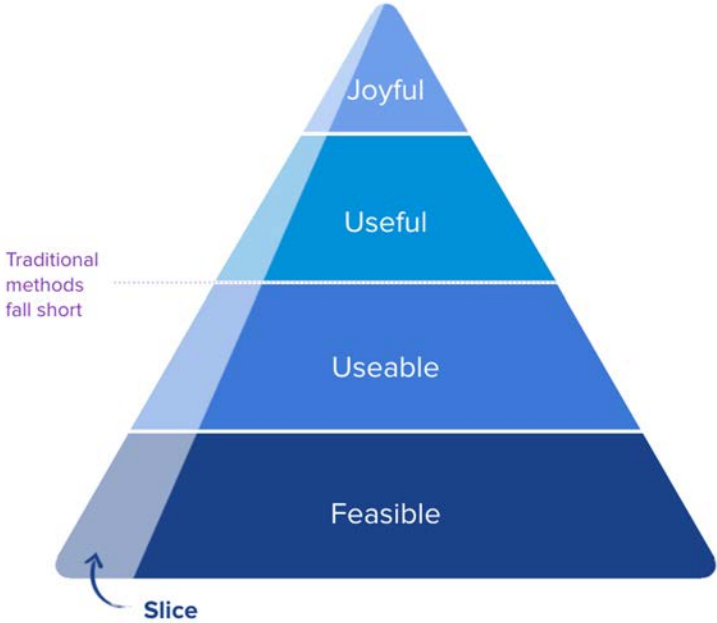
Involving the entire team in discovery builds empathy for the user/problem and creates a shared understanding within the team to create a joyful product for the customer



Understanding problems through user interviews



Listening to users in observation room



Development - Build the thing right

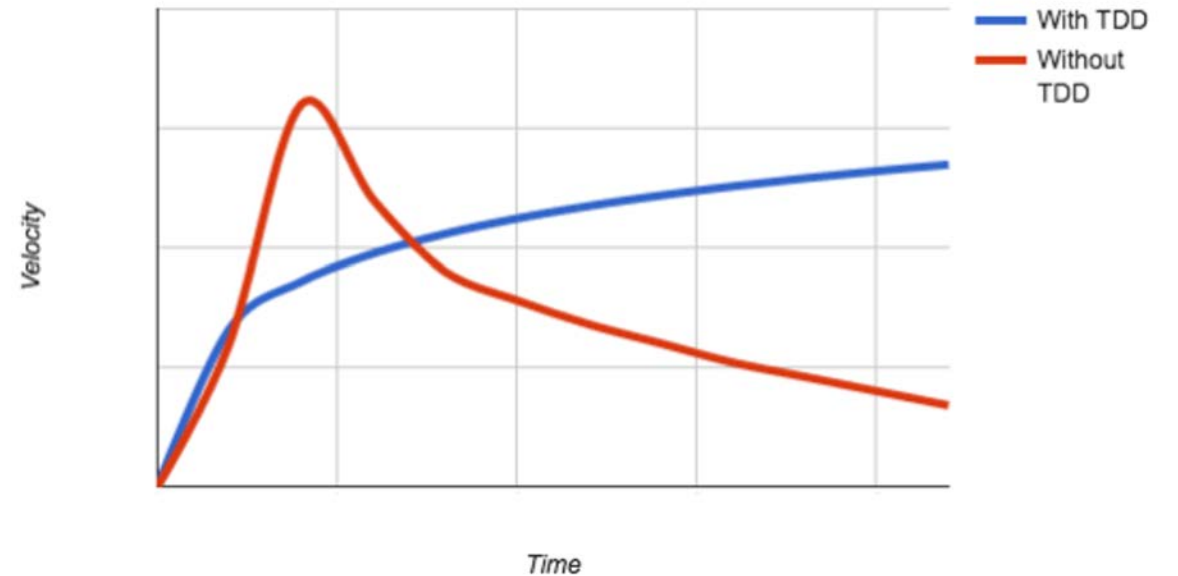
- ★ Small
- ★ Independent
- ★ Simple
- ★ Scalable
- ★ Secure

Engineering practices to reduce risk

- Evolutionary design
- CI, CD
- Collective ownership of code
- Test as you go along (TDD)
- Always be in a releasable state & release frequently
- Short iterations/feedback cycle
- Self Documentation



Velocity Over Time



Decompose and scale when needed

Unstructured

Classic “Layered” Architecture

Modular Monolith

Microservices

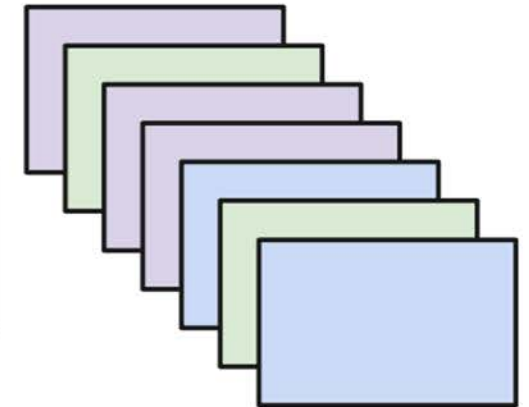
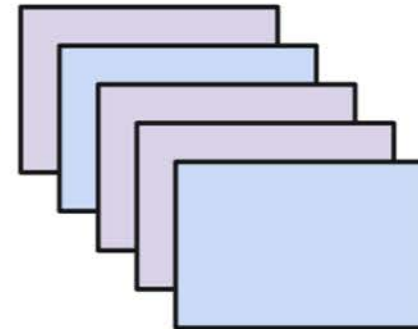
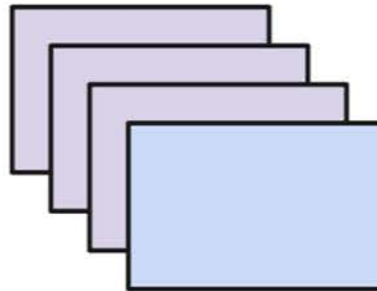
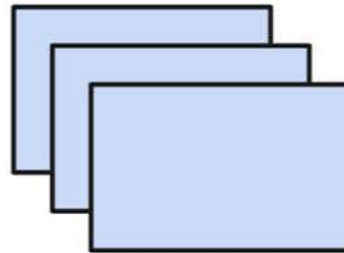
Unstructured

Namespace

Application + Libraries

Applications + Libraries

Applications + Libraries + Services



Continuum

Avoid, except for small tactical scripts

Consider where appropriate eg within modules

Technical separation, but hard to isolate and refactor across domains

Maximum speed while maintaining future options.

Ability to iterate on domain boundaries quickly

Well understood domains/boundaries

Need to scale (team & load)

A common question we get asked

“Why can’t I use low code or no code solutions which prevents me from having to depend on engineers? What’s the business value?”

Build it if is a key differentiator to your business

Why custom software is better as differentiators

- *Key differentiators need to be built to provide ROI quickly and adapt to the environment*
- *The testing can be automated easily*
- *Easy to build teams which can hold end-end responsibility*

Where 'no-code' and 'low-code' software doesn't measure up?

- *Hard to retain talent*
- *Automated testing is challenging*
- *Customisation to fit need is flaky*
- *Innovation is much harder and slower*
- *High Initial investment*

Culture - Build the right team

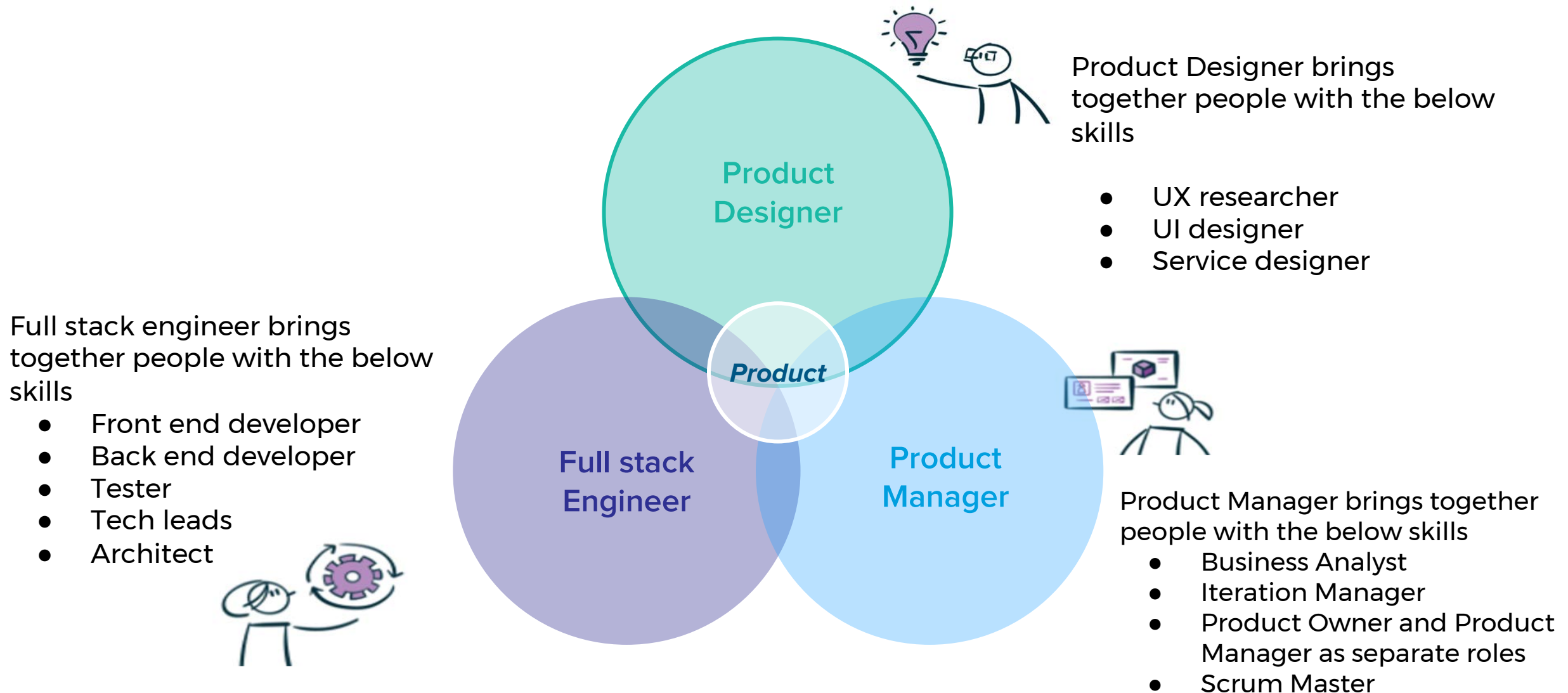
- ★ Small
- ★ Empowered
- ★ Independent
- ★ Aligned

It is an **autonomous & lean** team that has people with a variety of **skills and perspectives** that support each other towards a **shared goal**.

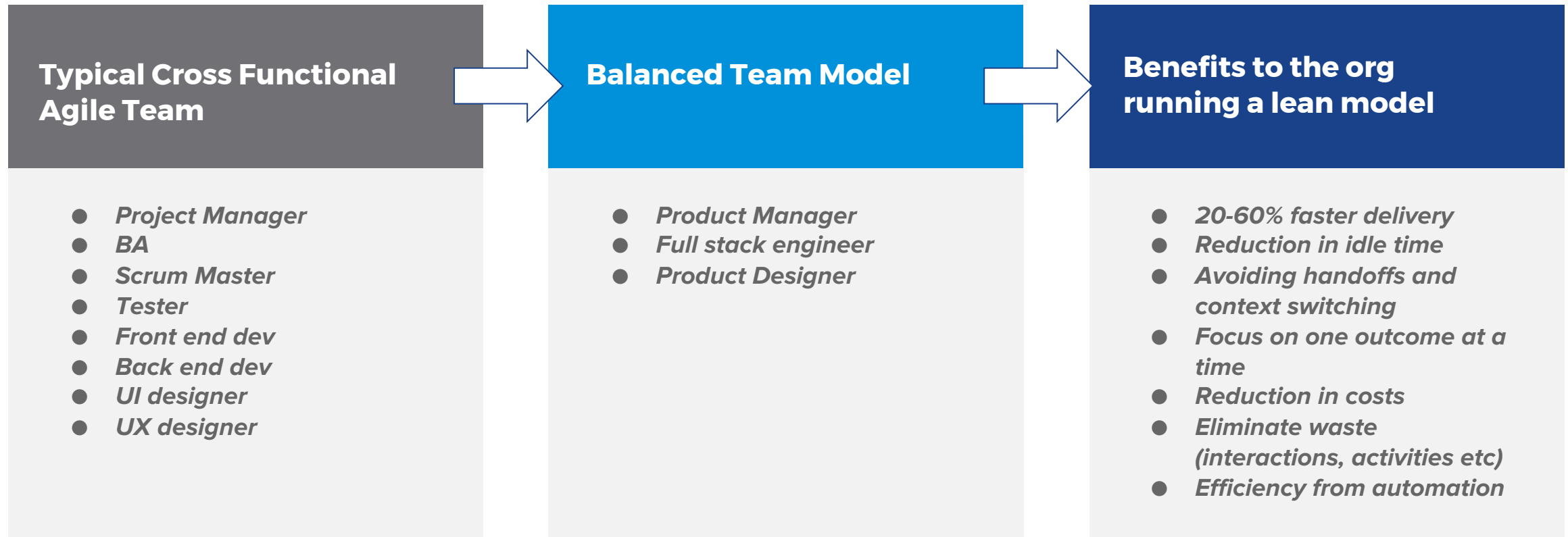


Balanced
Team

A balanced team: A lean operating model with T-Shapers



Benefits as compared to a typical team



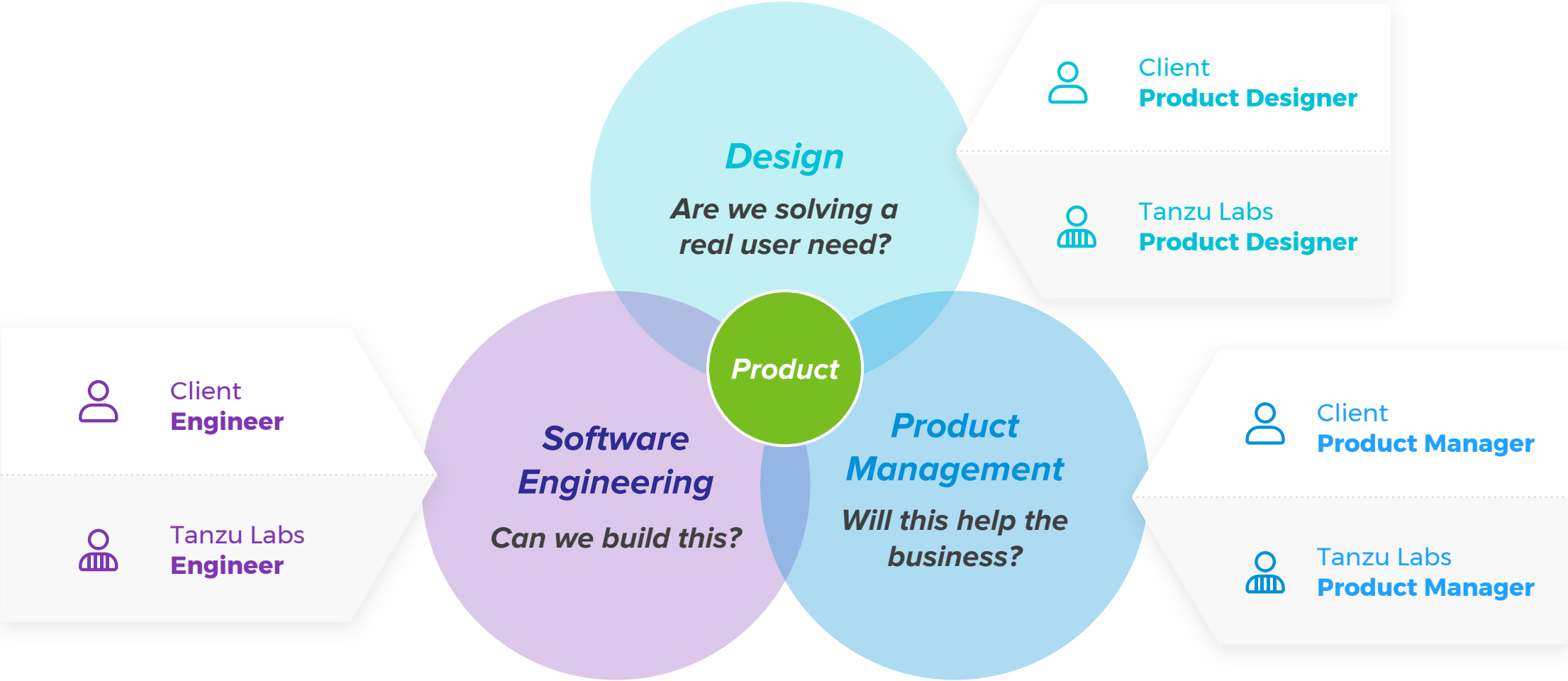
Estimate cost saving per team is \$1,500,000/year

Step 1: Creating the recipe

- ★ Creating recipe through pairing and enablement

Enablement by pairing with an experienced professional

1:1 Pairing to Maximize Quality and Enablement Outcomes



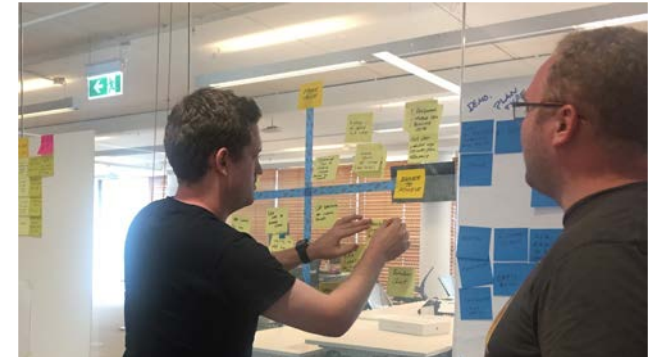
Enablement Approach



Observe and learn from an expert



Work with the expert

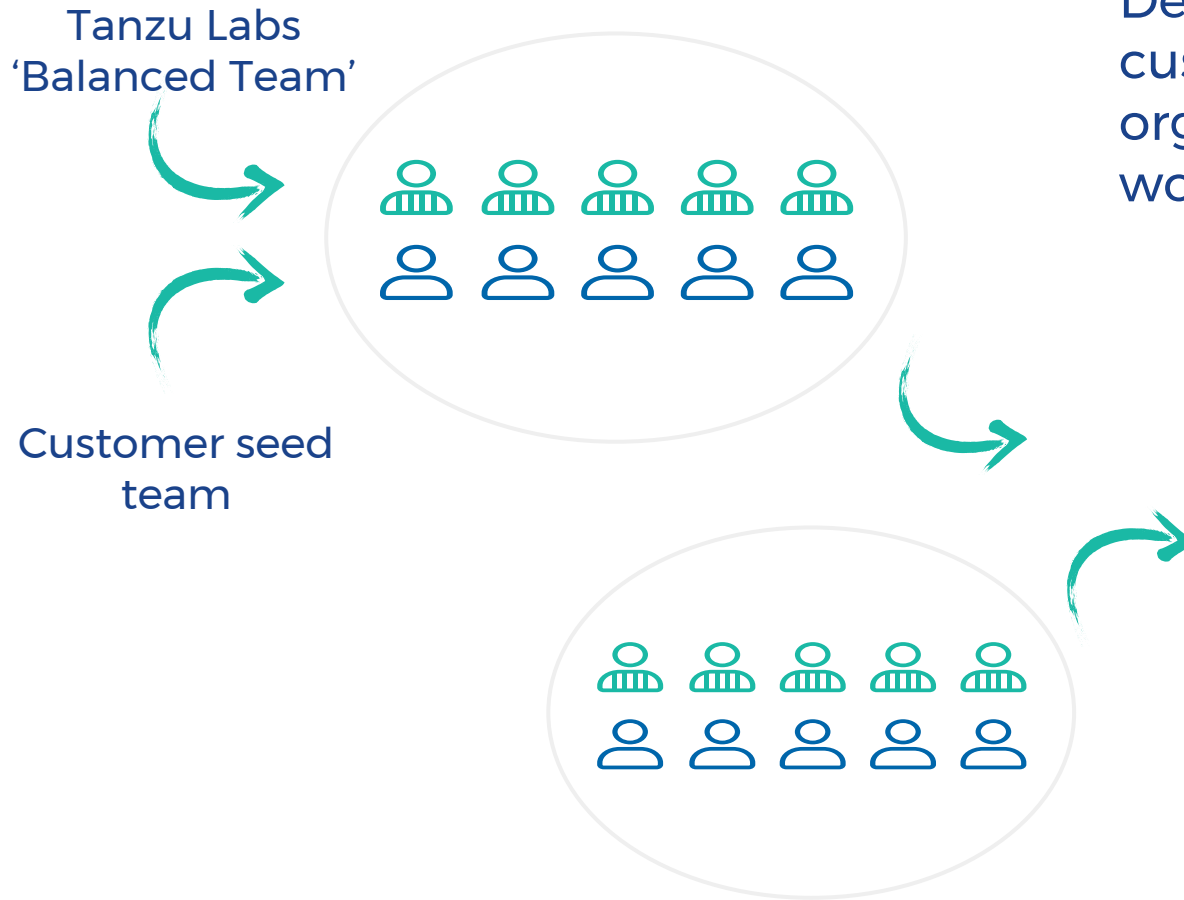


Drive alone and get feedback from the expert

Step 2: Scale the approach

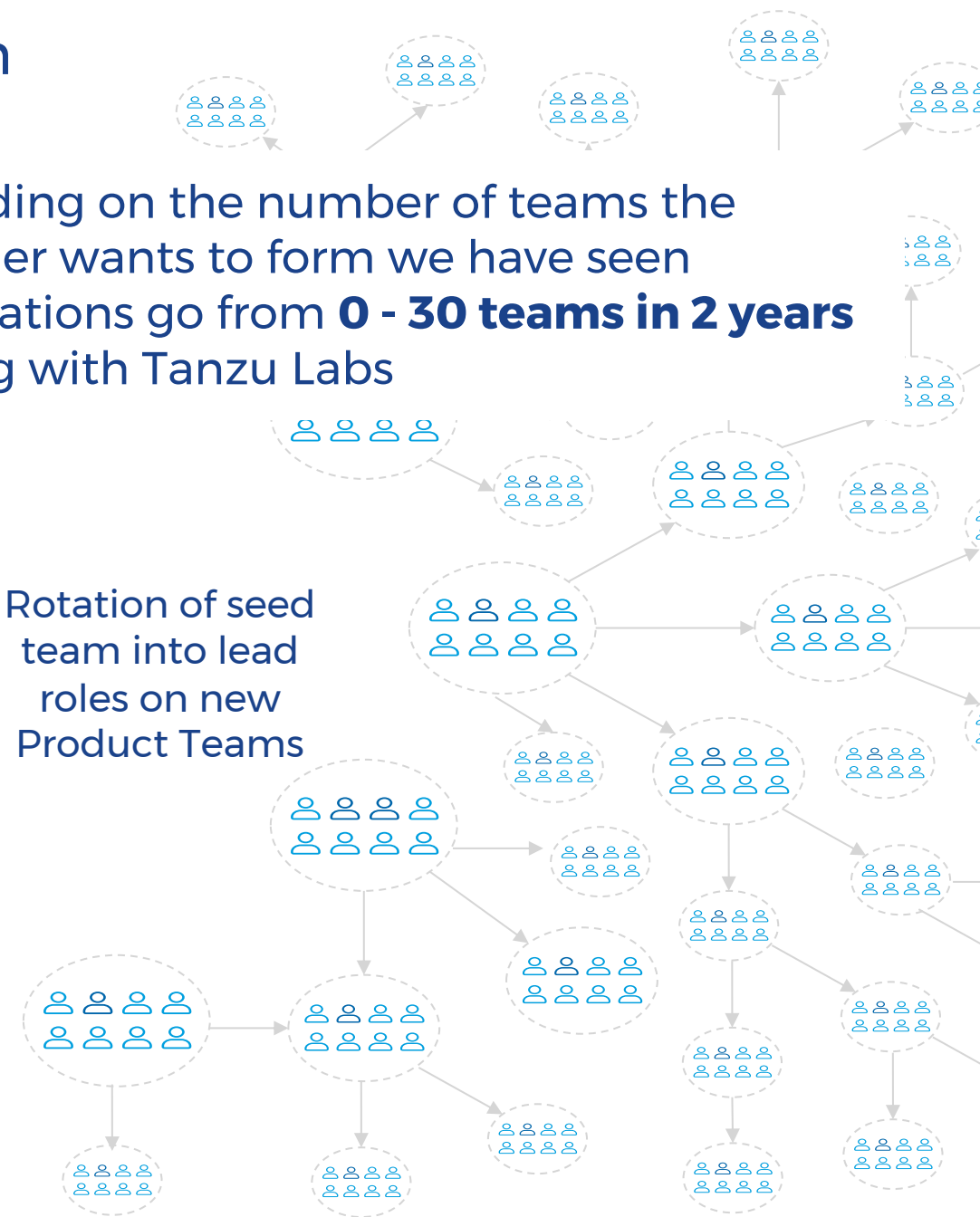
- ★ Decompose products
- ★ Seed teams with experts

Approach to scaling the model involves both Client & Tanzu Labs



Depending on the number of teams the customer wants to form we have seen organisations go from **0 - 30 teams in 2 years** working with Tanzu Labs

Rotation of seed team into lead roles on new Product Teams

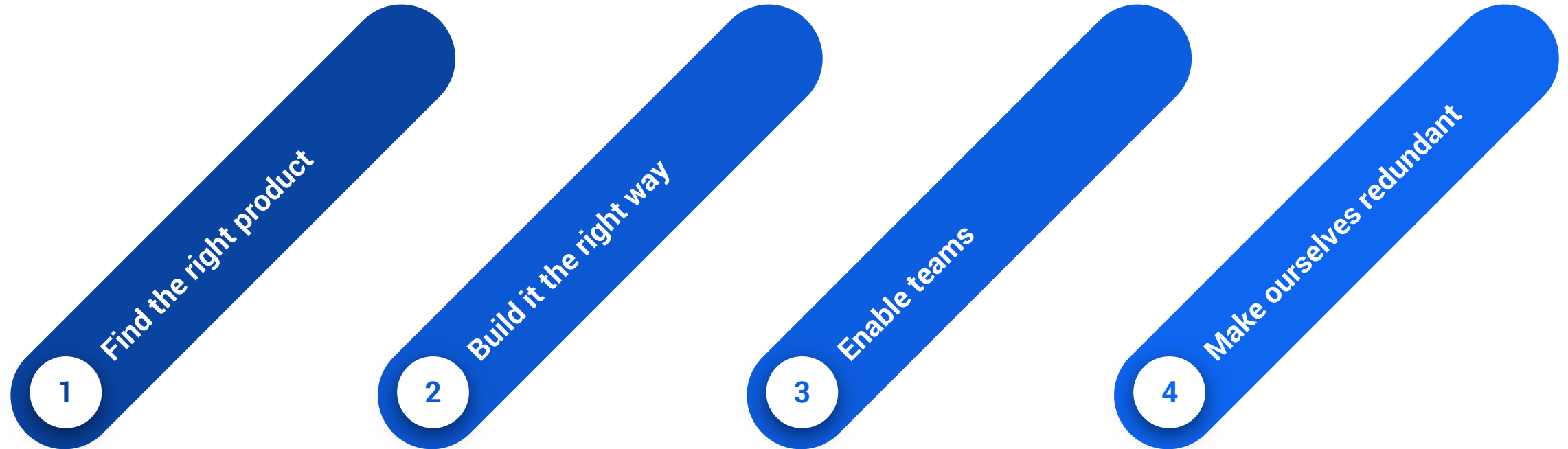


To conclude ...

Our recommendations

- Digital strategy starts with developing digital products not with process changes
- Start transformation with a single product to identify the recipe for your organisation and not with scaling in mind
- Create more product teams when there is a need to add more functions
- Seed teams with experts to avoid erosion of values and the culture
- Organise product teams around your business
- Form small, autonomous teams
- Setup a decentralised governance model

We are here to help



VMware Tanzu Labs

Thank You



email : anupa@vmware.com

LinkedIn: [anu-pandey](#)



email: varunachalam@vmware.com

LinkedIn: [venkiarun](#)