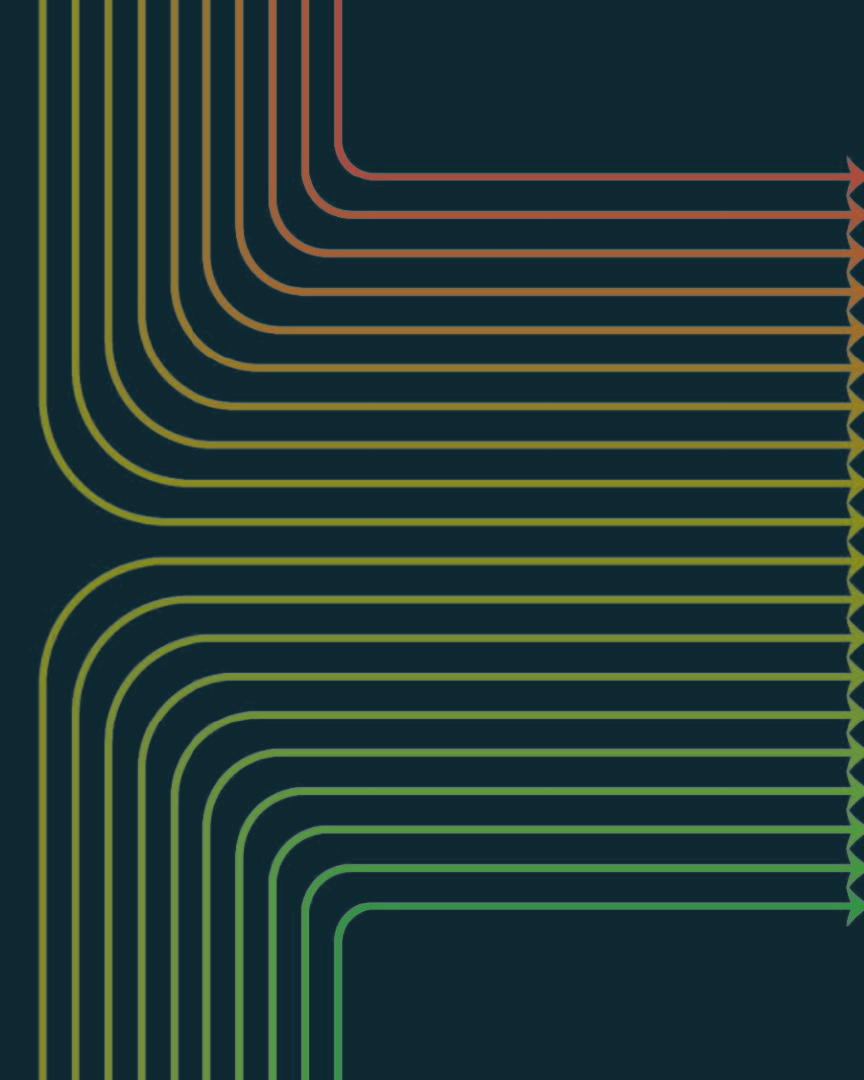




# Mapping Flow to Reveal, Navigate, and Overcome Constraints

Steve Pereira · Visible Flow Consulting



# Start with the smallest possible scope

<8%

of a contributor's time is spent producing deliverable work.

Ramp  
up

VALUE  
ADD

Teams  
msg

Rework

Teams  
request

Unblock  
someone

Re-ramp  
up

VALUE  
ADD

Status  
update

# So we made the 8% faster with AI



## What genuinely got faster

First drafts and boilerplate

Documentation

Research and summarisation

Individual output per hour



## What did not move

Funding and re-scoping decisions

Environment and access provisioning

Authorisation and approval queues

Integration, handoffs, and sign-off

Cut the fastest 8% of a system and total lead time falls about 1%.

# Suboptimization is eating your flow

Improving one part of a system in a way that does not improve — and often degrades — the whole.

# Suboptimizing is too easy



An AI-powered  
builder



A process-mapped  
segment



An output-focused  
program



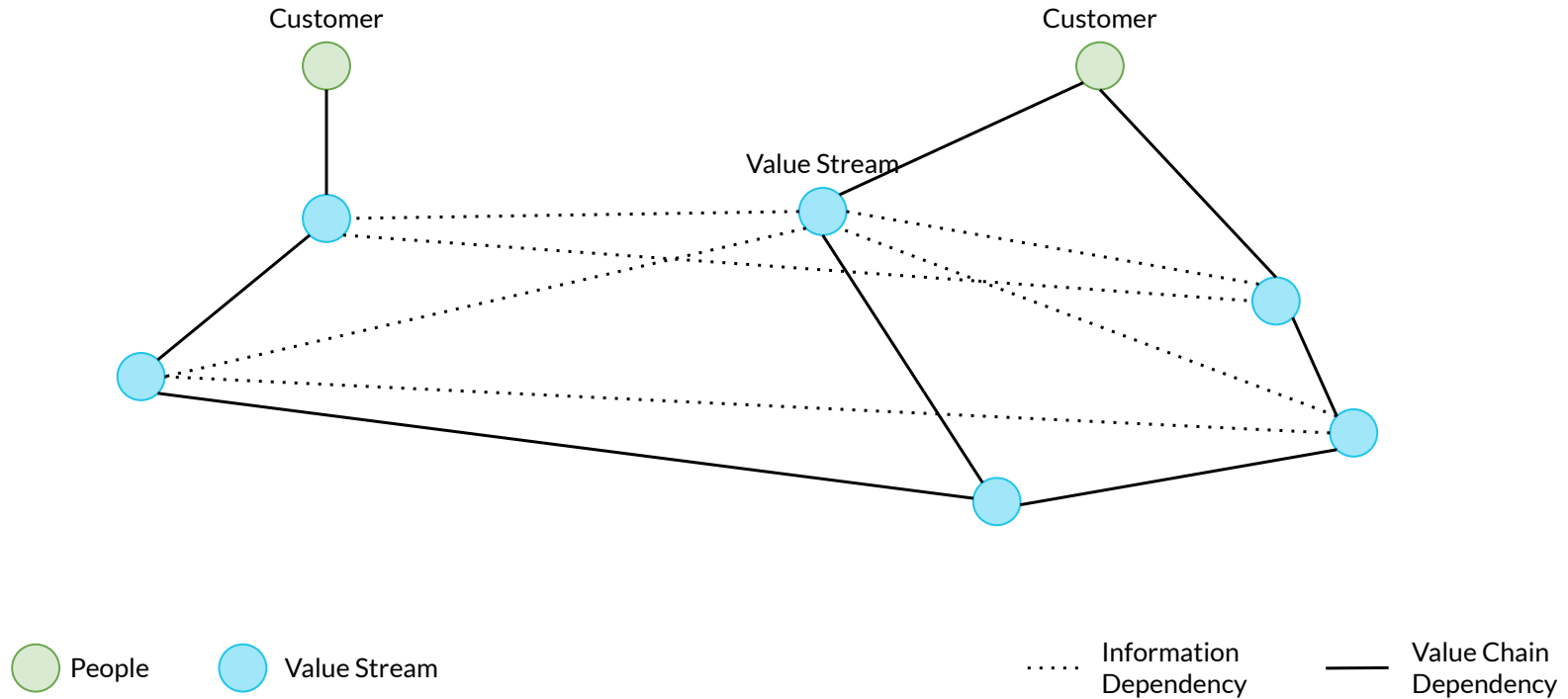
A framework  
rollout

# The default view

# The default approach

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# What delivery actually looks like



Find the constraint  
before funding fixes



# The constraint is rarely where the attention is



↑ 58% of total elapsed time  
sits in one stage

# Acquisition is a classic culprit

ATTENTION GOES HERE

5–10%

The inner loop

WASTE LIVES HERE

~18 months

The acquisition cycle

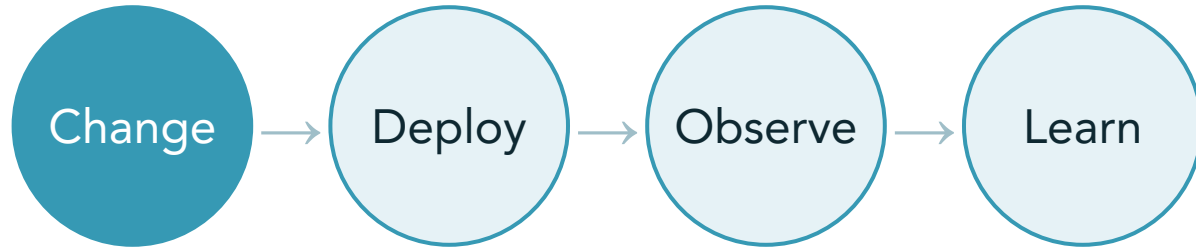
# What this means if you hold budget or authority

- 1 The constraint is usually upstream of the team you are funding.
- 2 Nobody owns end-to-end acquisition cycle time.
- 3 A 20% cut in acquisition lead time beats a 2x developer speedup.
- 4 You have authority your delivery teams do not.

Three layers in one system

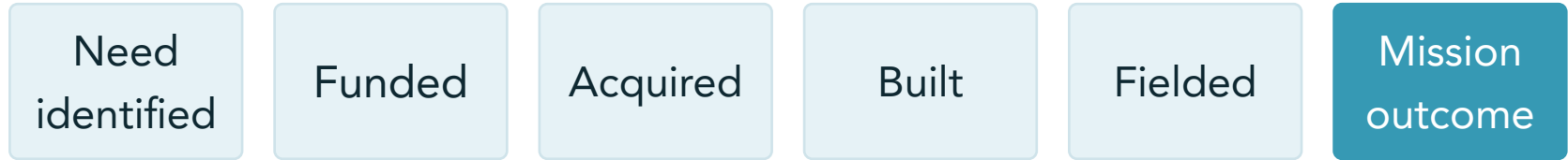


# Operational: Continuous delivery is your base layer



The loop matters only if learning happens

## Tactical: The value stream of the mission domain



# Strategic: Enterprise-scale bets

A

## Individual innovation

- Fast to start, visible in weeks
- Unbounded local experimentation
- Produces demos and champions
- Adds work in progress the system can't address

B

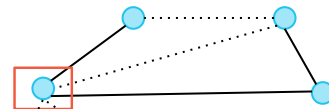
## End-to-end flow for Outcomes

- Slower to start, but compounds
- Bounded by the mission value stream
- Produces throughput and evidence
- Produces what the system can handle

# Layers of Flow

**Strategic**

**Value Stream Network / Multi-Portfolio scale**



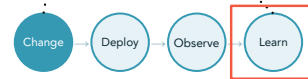
**Tactical**

**Value Stream / Portfolio scale**

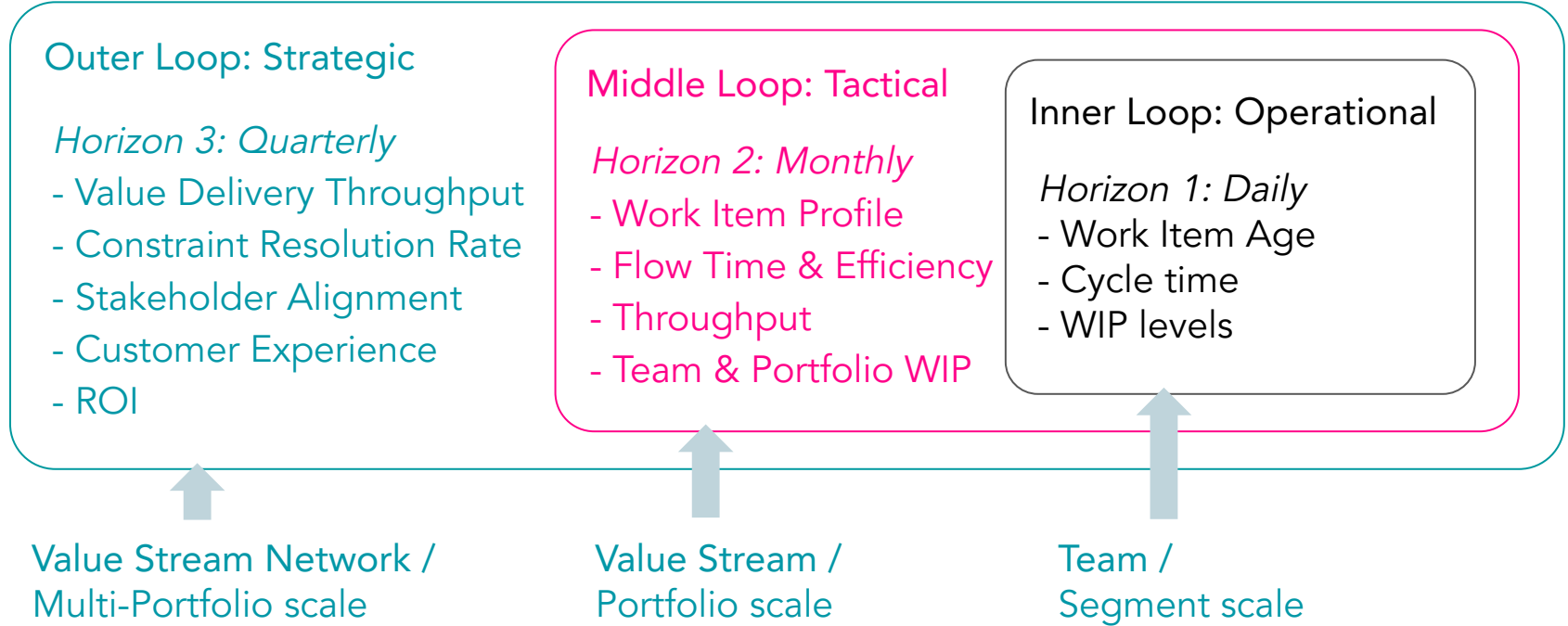


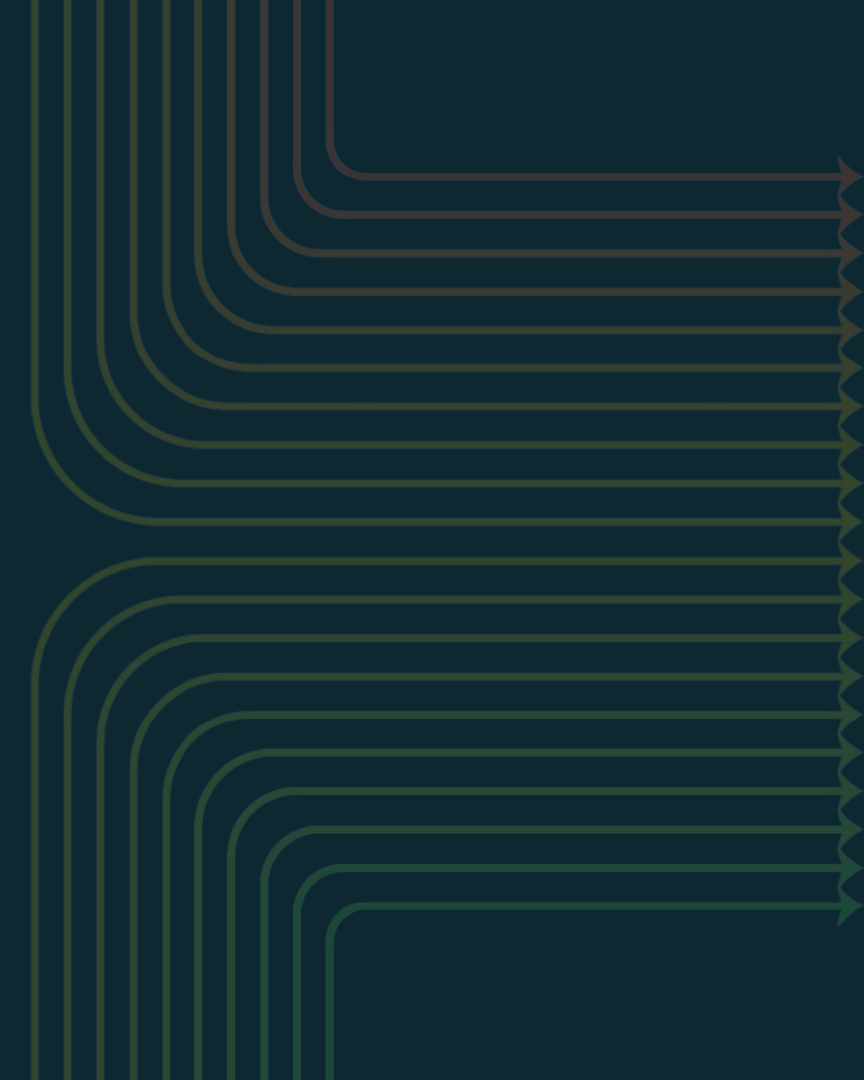
**Operational**

**Team / Segment scale**



# Measurement at scale: Layers as Loops





If you want to go fast,  
go alone.

If you want to go far,  
go together.

AI makes "alone"  
faster,  
and together  
harder than ever

# Mapping is the baseline



## Mapping for sense-making

- Alignment for outcomes
- Making work visible
- Decisions become obvious



## Value stream for outcomes

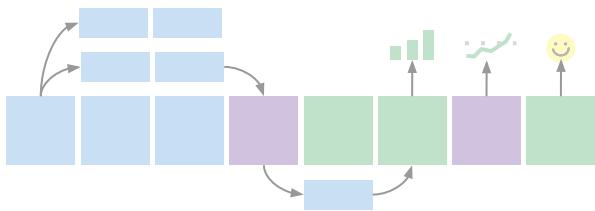
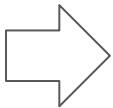
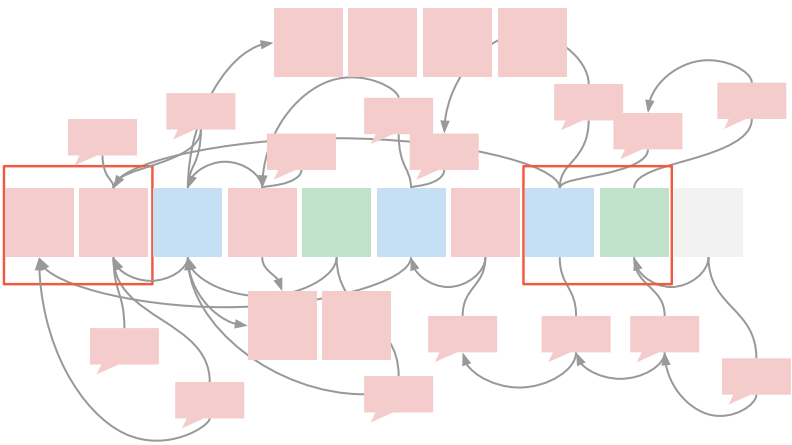
- Connect work to mission
- Clear metrics
- Focus on constraint

# Mapping for flow

Complexity

to

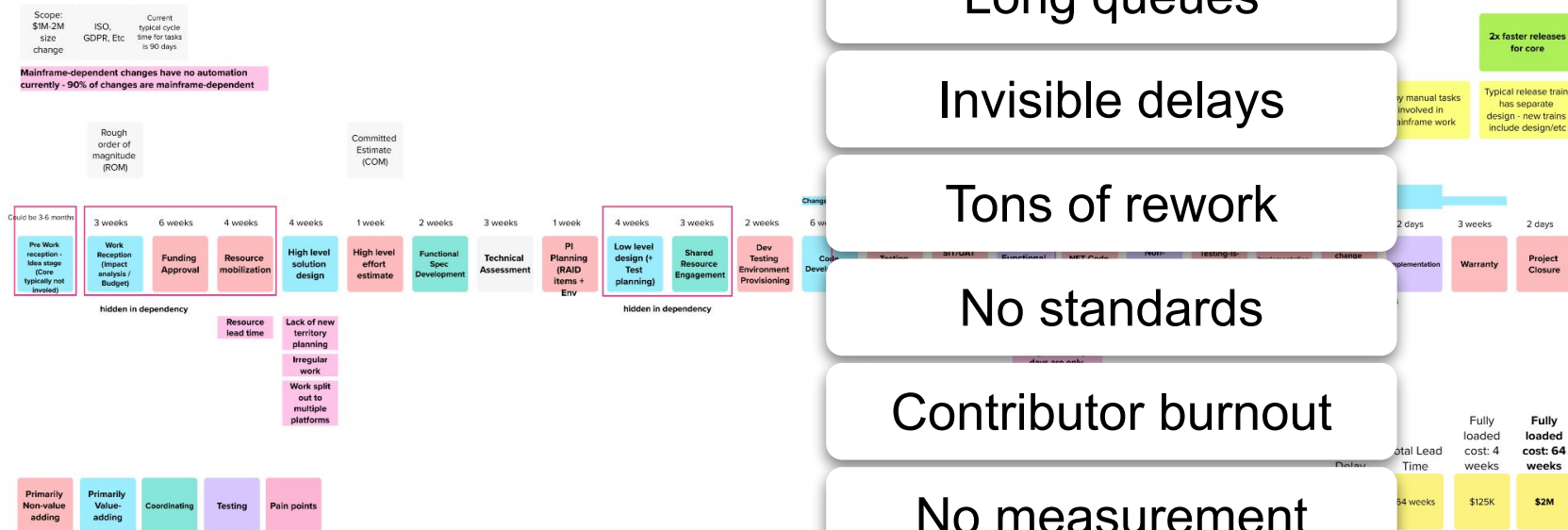
Clarity



# Before

64 weeks at \$2M per release

## Release Train: Current State



Painful handoffs

Long queues

Invisible delays

Tons of rework

No standards

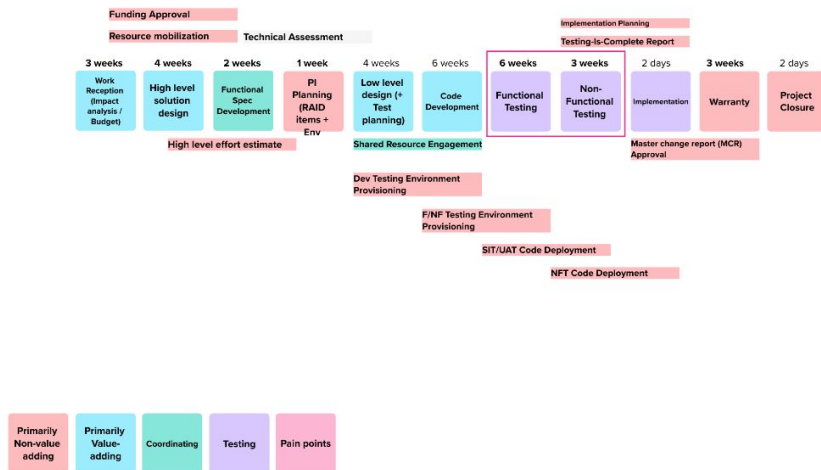
Contributor burnout

No measurement

# After

32 weeks at 50%, \$1M cost savings in year one

## Release Train: Future State



Waste elimination

Faster delivery

Faster learning

Higher eSAT & cSAT

More visibility & clarity

Effective alignment

Better collaboration



|                 |                             |
|-----------------|-----------------------------|
| Total Lead Time | Fully loaded cost: 32 weeks |
| releases core   | 32 weeks                    |
|                 | 50% Savings                 |

# What happens when the constraint is the target

BEFORE

64 weeks

AFTER

32 weeks

~\$1M saved in year one

–75%

Lead time

4x

Deploy frequency

–75%

Time to repair

2x

Throughput

# Map one value stream, end to end, in 90 minutes

1

Pick the  
outcome  
domain

2

Go to the  
contributors

3

Work  
backwards

4

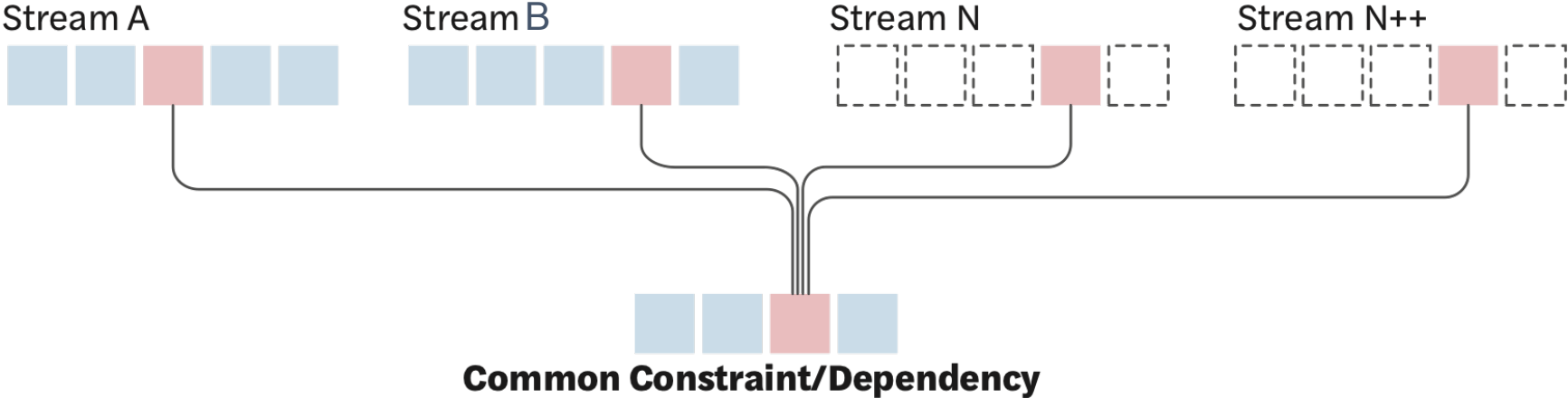
Stage time,  
Wait time

5

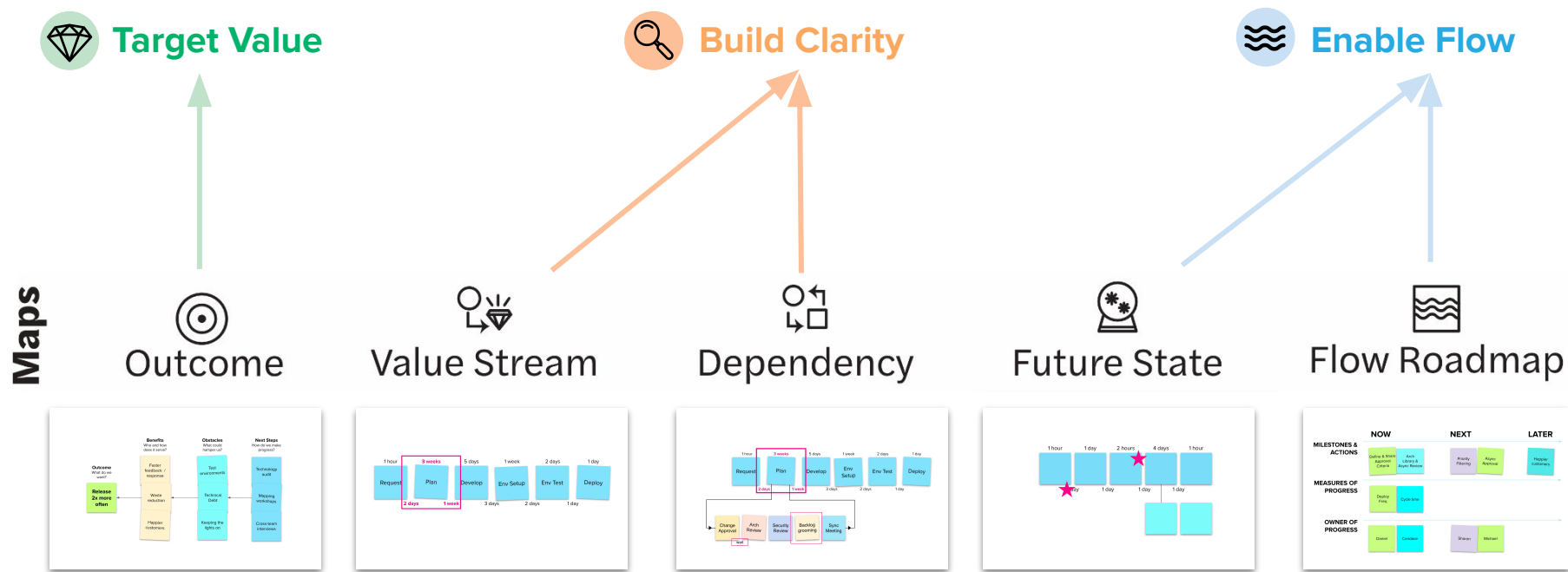
Run the  
numbers

The key is to act on the insights. Too many maps drive too little action.

# Mapping for ROI



# The Flow Engineering Path



# Don't go alone.

Going far is a team sport, and the teams that go furthest borrow a map from someone who has walked the path.

Want to learn the full flow of Flow Engineering?

*Join my Workshop!*

steve@visible.is · visible.is

