RUNDECK

DevOps Kaizen:

Practical Steps to Start & Sustain an Organization’s Transformation

Damon Edwards
@damonedwards

jax DEVOPS

3-6 April, 2017
Park Plaza Victoria London

RUN DECK

Tools

dto SOLUTIONS

DevOps Consulting

Operational Improvement

@damonedwards

Dev Ops Cafe
With
John Willis
Damon Edwards

DEVOPS ENTERPRISE SUMMIT

Community

ticketmaster® CSC EQUIFAX

KAISER PERMANENTE GUARDIAN® at&t omgeo

Adobe barclaycard emdeon® TD

U.S. Citizenship and Immigration Services intuit.

zynga® verizon LOYAL3 D&B

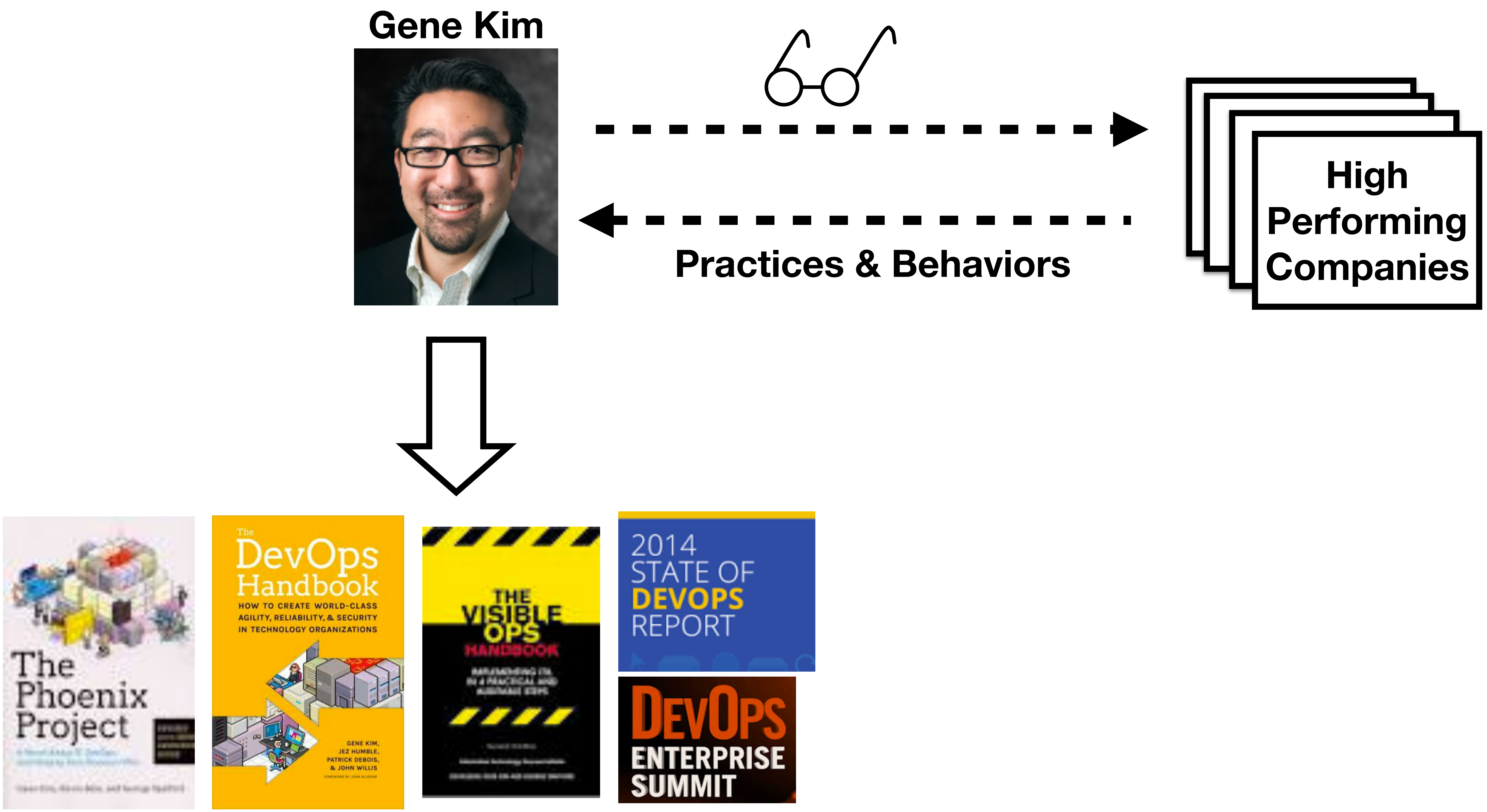
RESTORATION HARDWARE bwin.party digital entertainment

RMS Linkedin BT CCP®

vmware® digby™ Cisco webex TransUnion.

RueLaLa® backcountry ING

JPMorgan Chase & Co. E*TRADE FINANCIAL® Fidelity INVESTMENTS

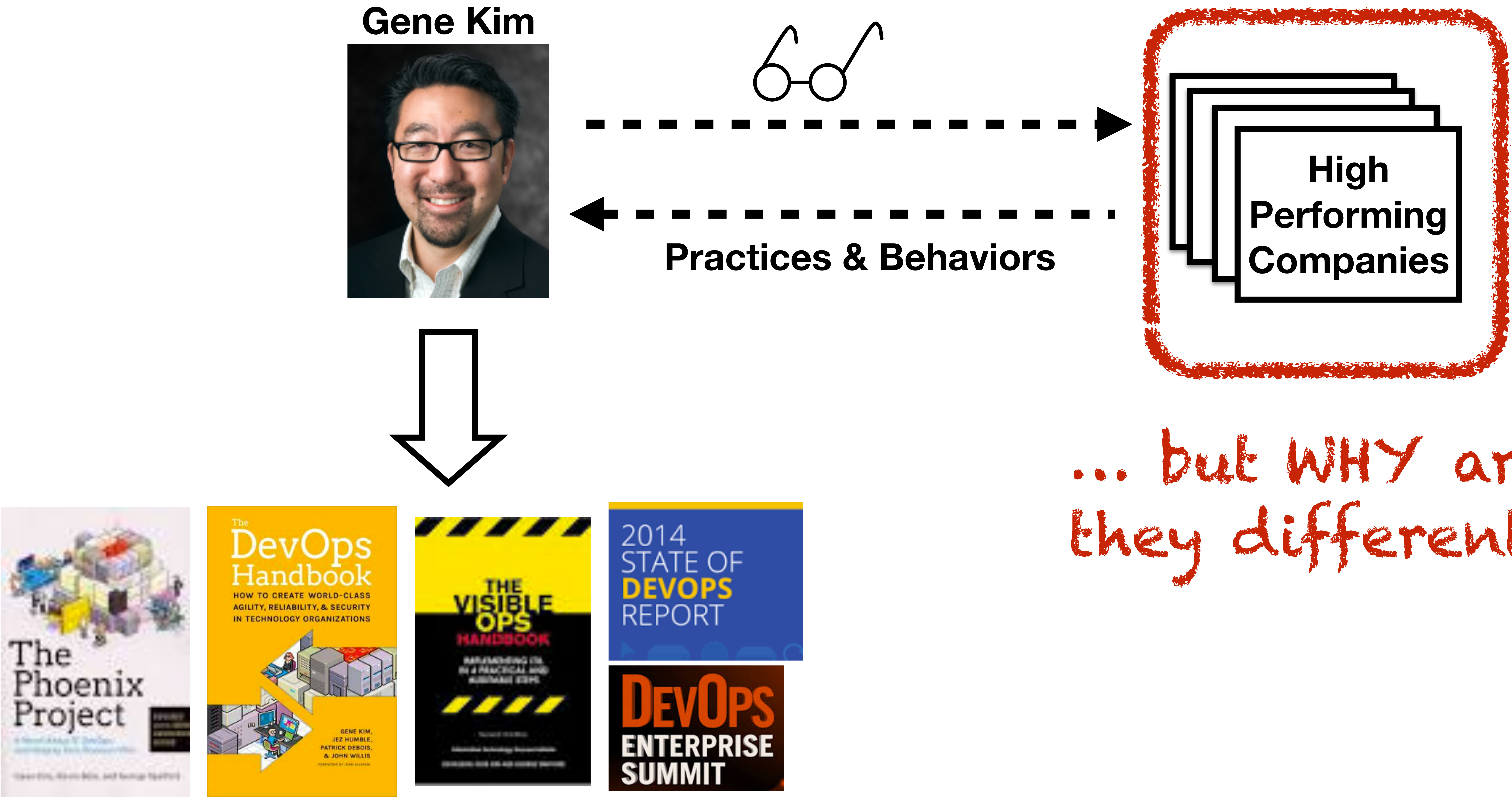


Gene Kim

Practices & Behaviors

High Performing Companies

The Phoenix Project
The DevOps Handbook
HOW TO CREATE WORLD-CLASS AGILITY, RELIABILITY, & SECURITY IN TECHNOLOGY ORGANIZATIONS
GENE KIM, JEZ HUMBLE, PATRICK DEBOIS, & JOHN WILLIS
THE VISIBLE OPS HANDBOOK
2014 STATE OF DEVOPS REPORT
DEVOPS ENTERPRISE SUMMIT



Gene Kim

Practices & Behaviors

High Performing Companies

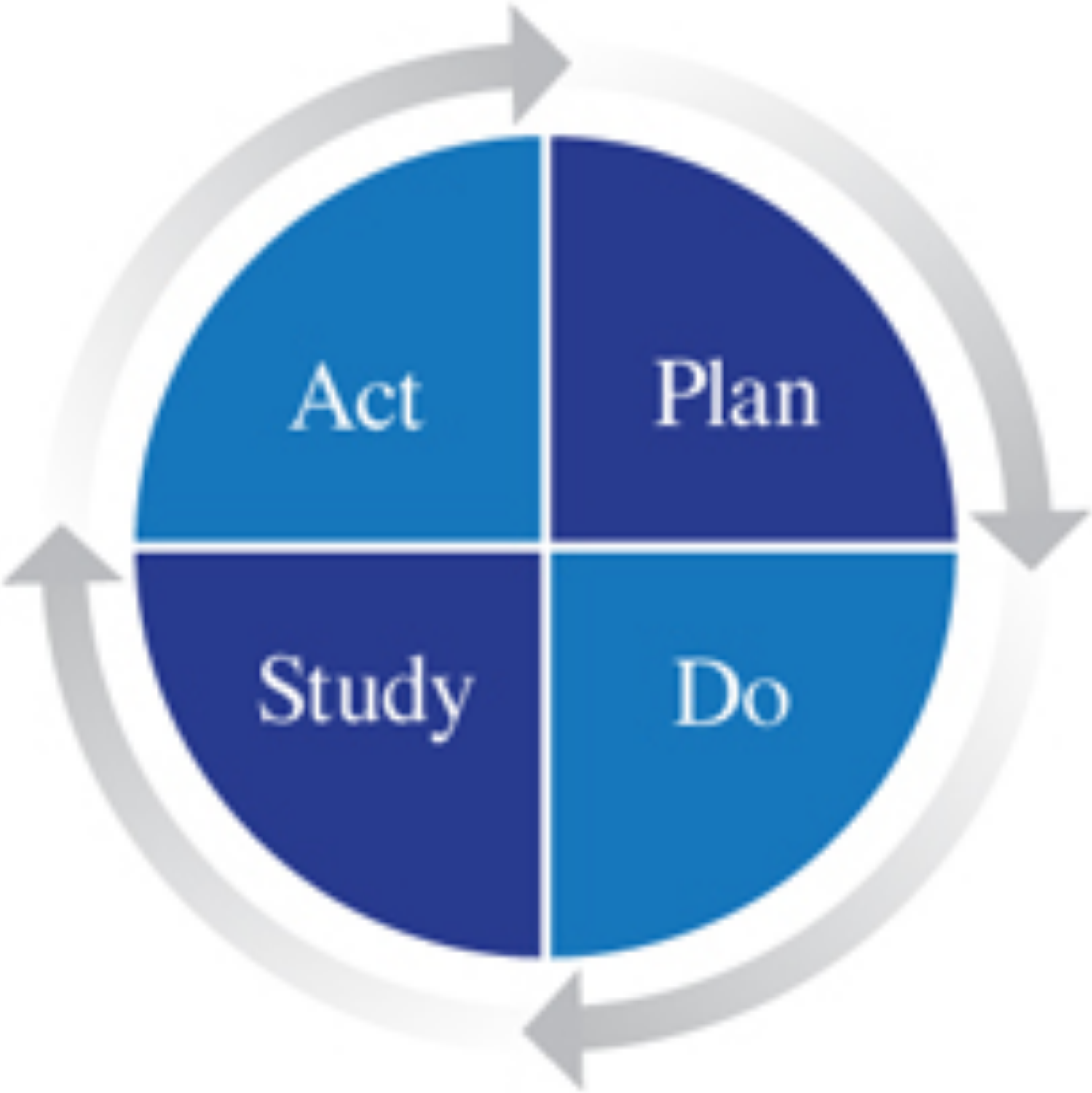
The Phoenix Project
The DevOps Handbook
HOW TO CREATE WORLD-CLASS AGILITY, RELIABILITY, & SECURITY IN TECHNOLOGY ORGANIZATIONS
GENE KIM, JEZ HUMBLE, PATRICK DEBOIS, & JOHN WILLIS
THE VISIBLE OPS HANDBOOK
2014 STATE OF DEVOPS REPORT
DEVOPS ENTERPRISE SUMMIT

... but WHY are they different?

The ability to **improve**.

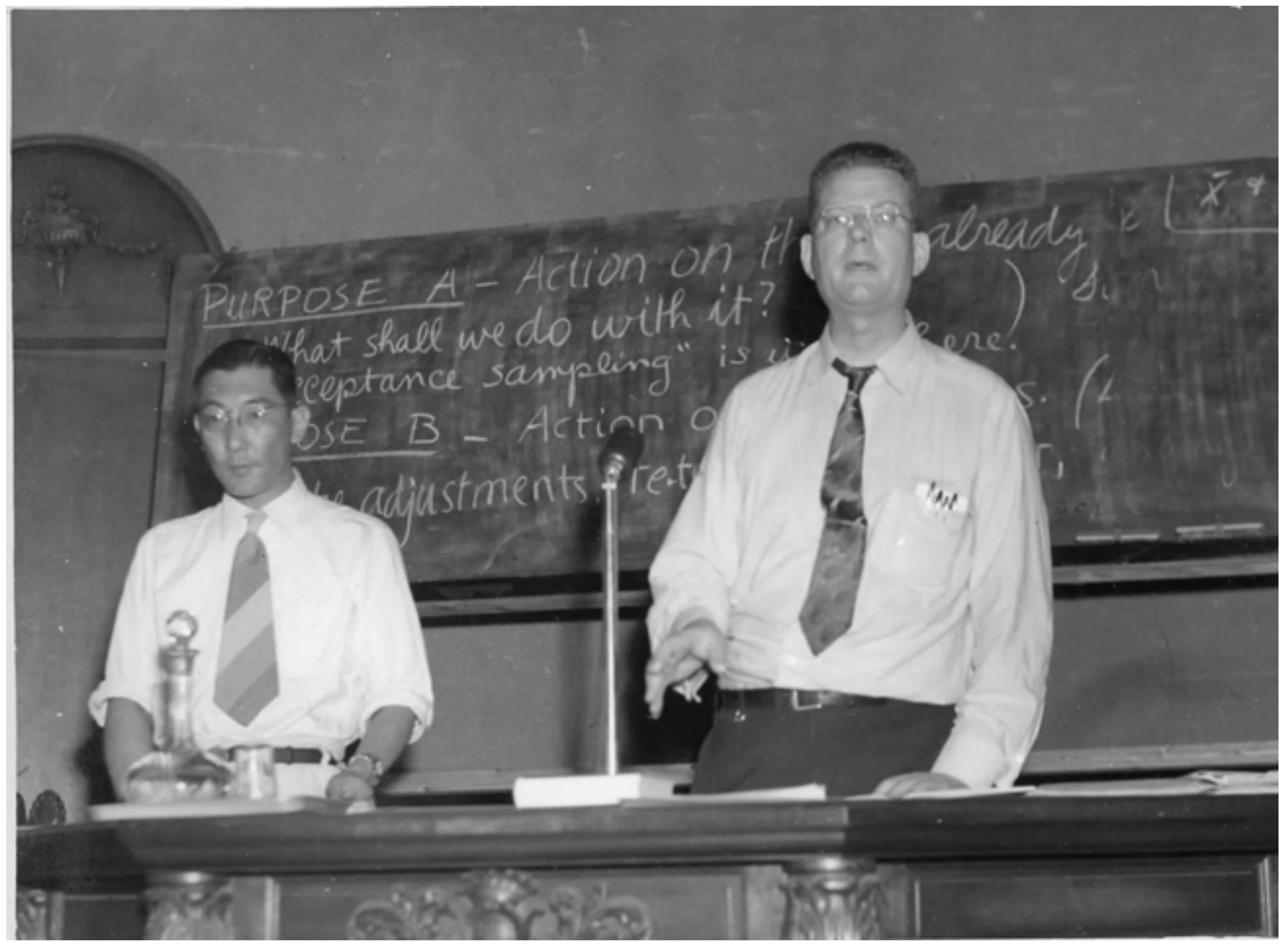
The unique trait of high-performing companies is that they are good at getting better.

Improvement already has a well known recipe: Plan - Do - Study - Act (PDSA)



Other variants:
PDCA
OODA

W. Edwards Deming - 1950


PURPOSE A - Action on the ... already
What shall we do with it?
Acceptance sampling is ... here.
POSE B - Action o...
adjustments re-t...

© The Deming Institute

Why are so many organizations unable to improve?

Why are so many organizations unable to improve?

1. The work isn’t visible

Why are so many organizations unable to improve?

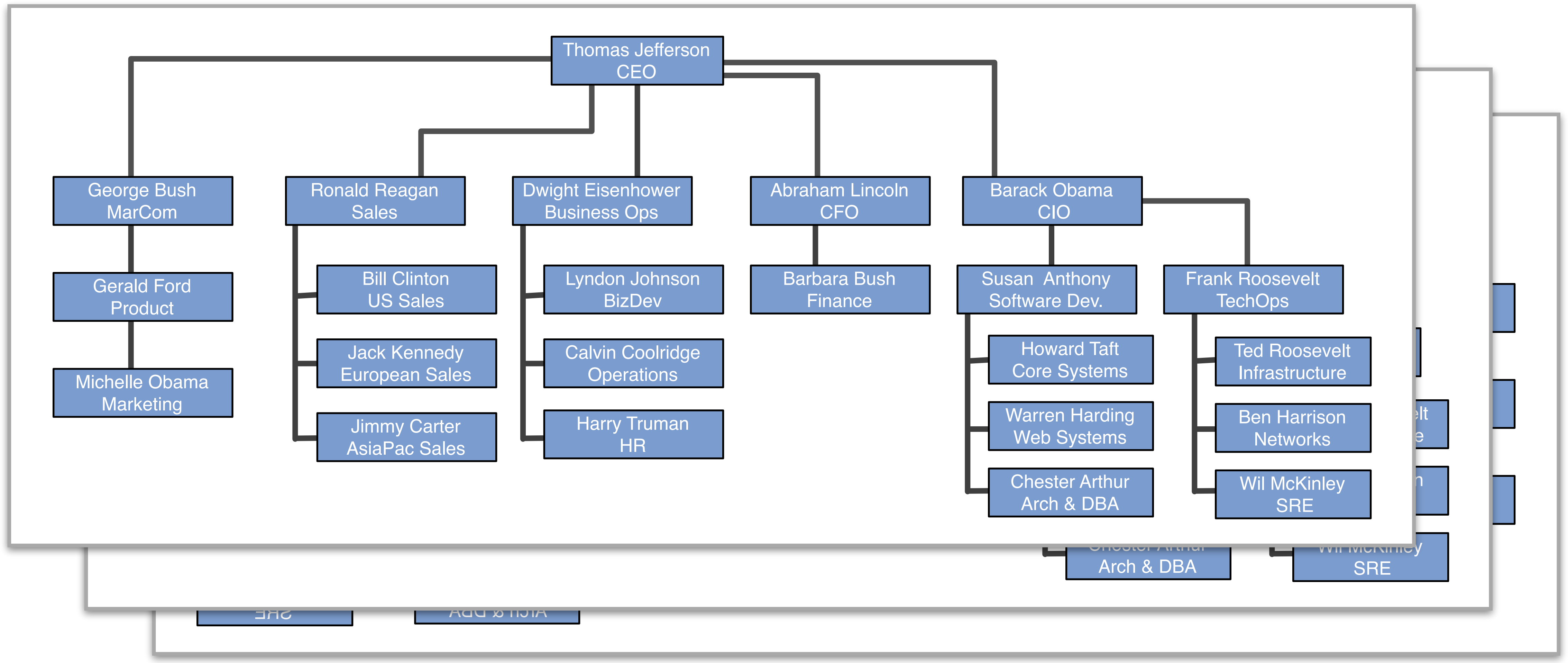
1. The work isn't visible
2. People are working out of context

Why are so many organizations unable to improve?

1. The work isn’t visible
2. People are working out of context
3. Inertia is pulling your org out of alignment

Traditional “visibility”

Org Charts



Thomas Jefferson
CEO

George Bush
MarCom

Gerald Ford
Product

Michelle Obama
Marketing

Ronald Reagan
Sales

Bill Clinton
US Sales

Jack Kennedy
European Sales

Jimmy Carter
AsiaPac Sales

Dwight Eisenhower
Business Ops

Lyndon Johnson
BizDev

Calvin Coolridge
Operations

Harry Truman
HR

Abraham Lincoln
CFO

Barbara Bush
Finance

Barack Obama
CIO

Susan Anthony
Software Dev.

Howard Taft
Core Systems

Warren Harding
Web Systems

Chester Arthur
Arch & DBA

Frank Roosevelt
TechOps

Ted Roosevelt
Infrastructure

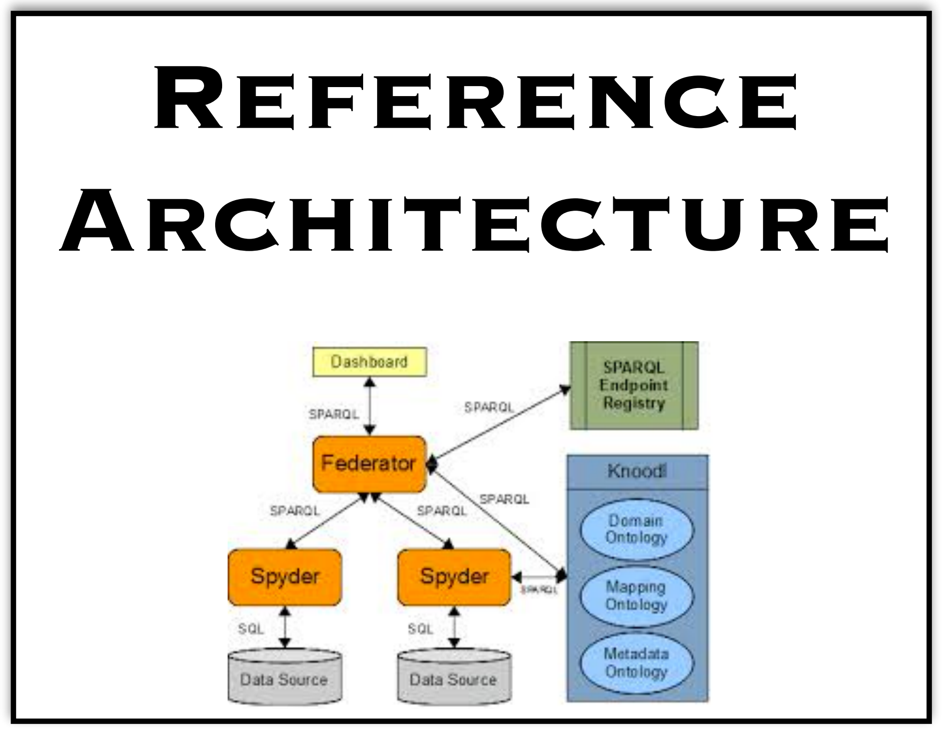
Ben Harrison
Networks

Wil McKinley
SRE

Strategy

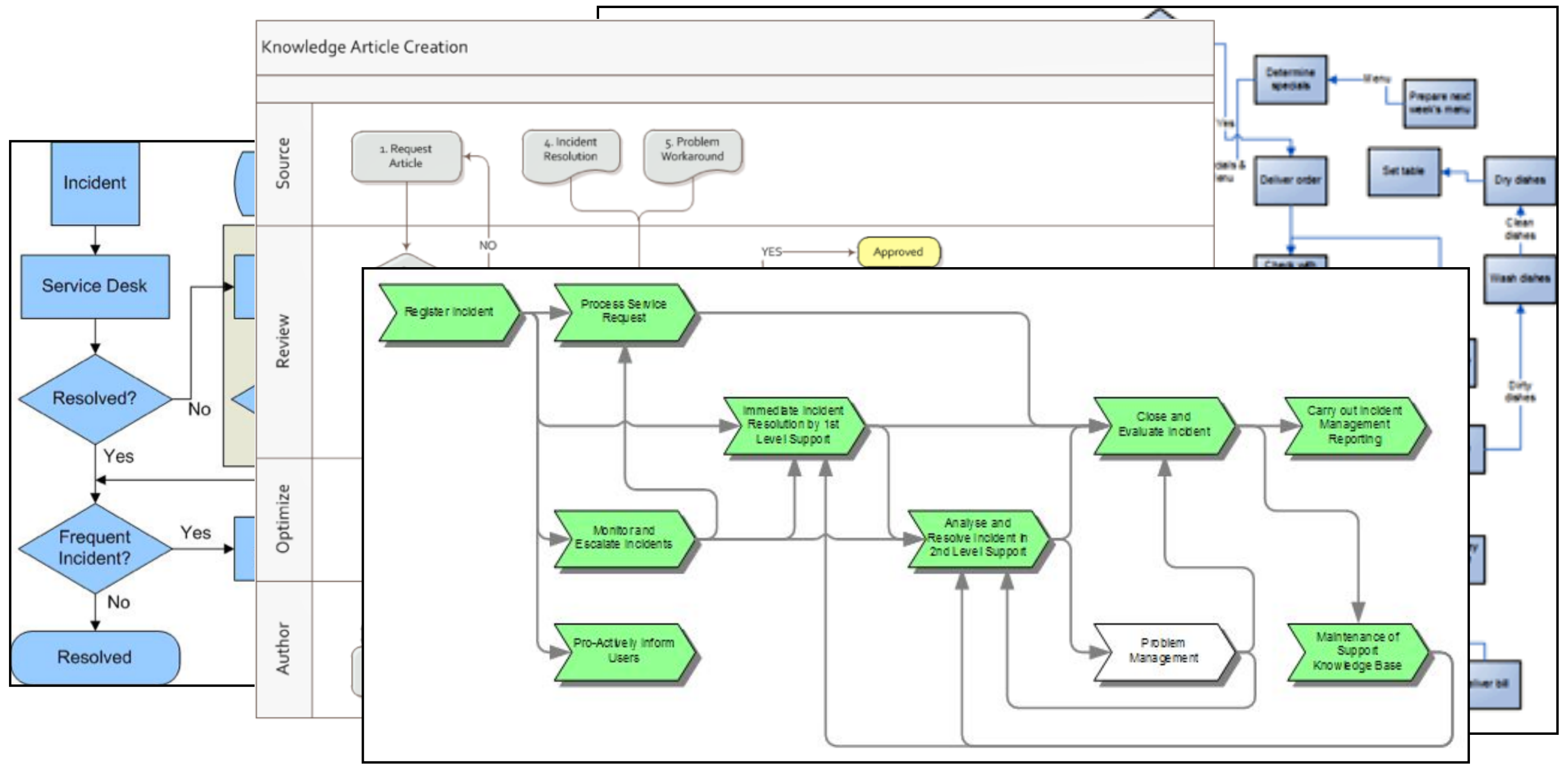
&

Budget

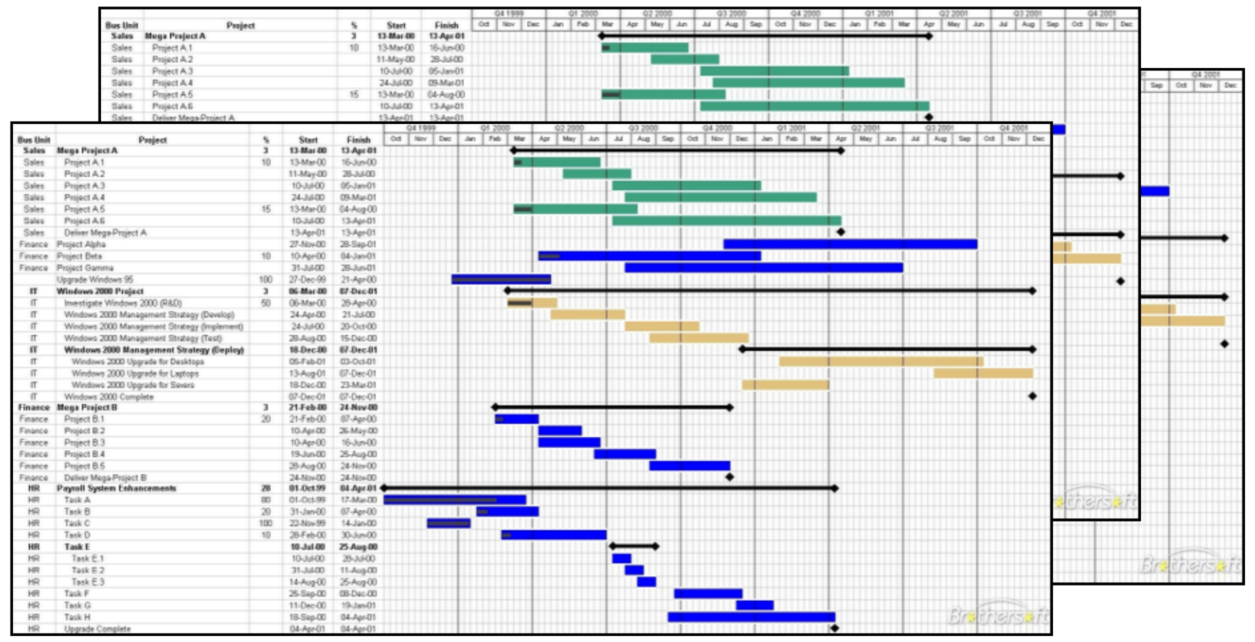


Reference Architecture

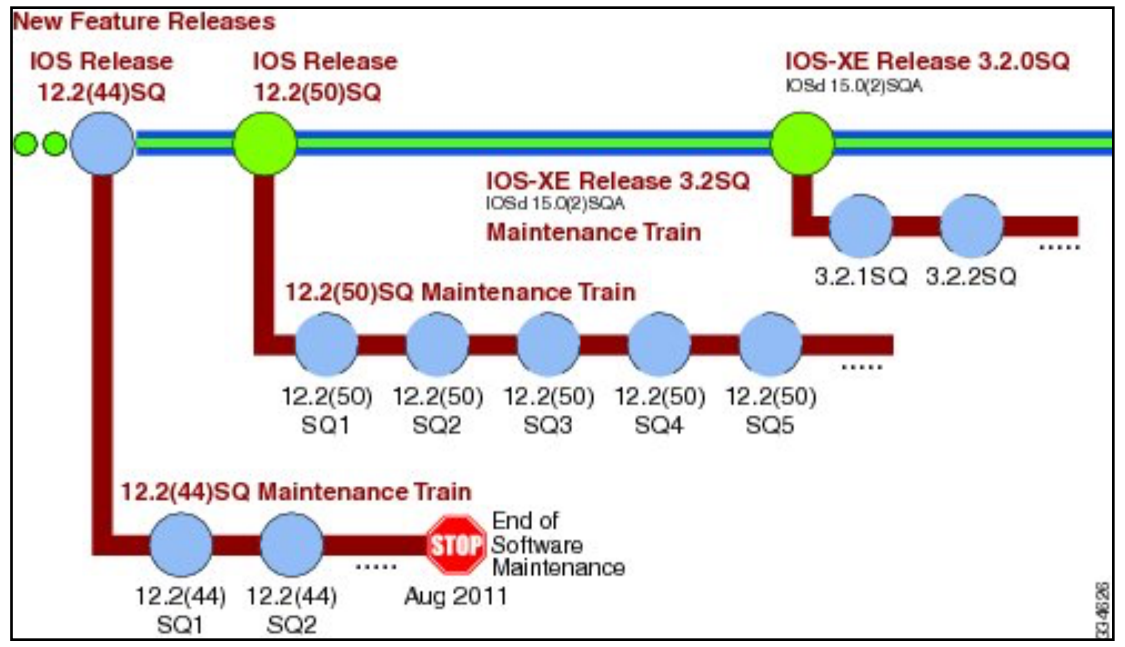
Documented Processes



Project Plans



Release Trains



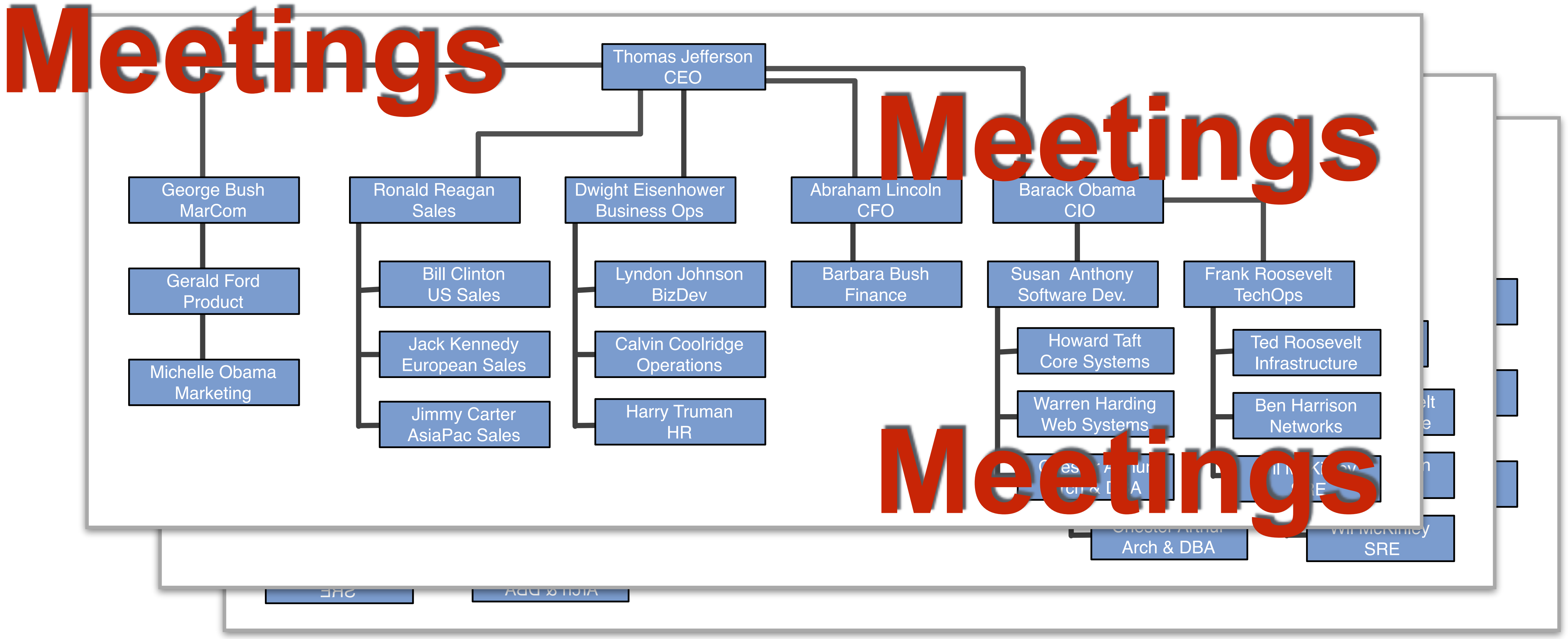
Traditional “visibility”

Org Charts

Meetings

Meetings

Meetings



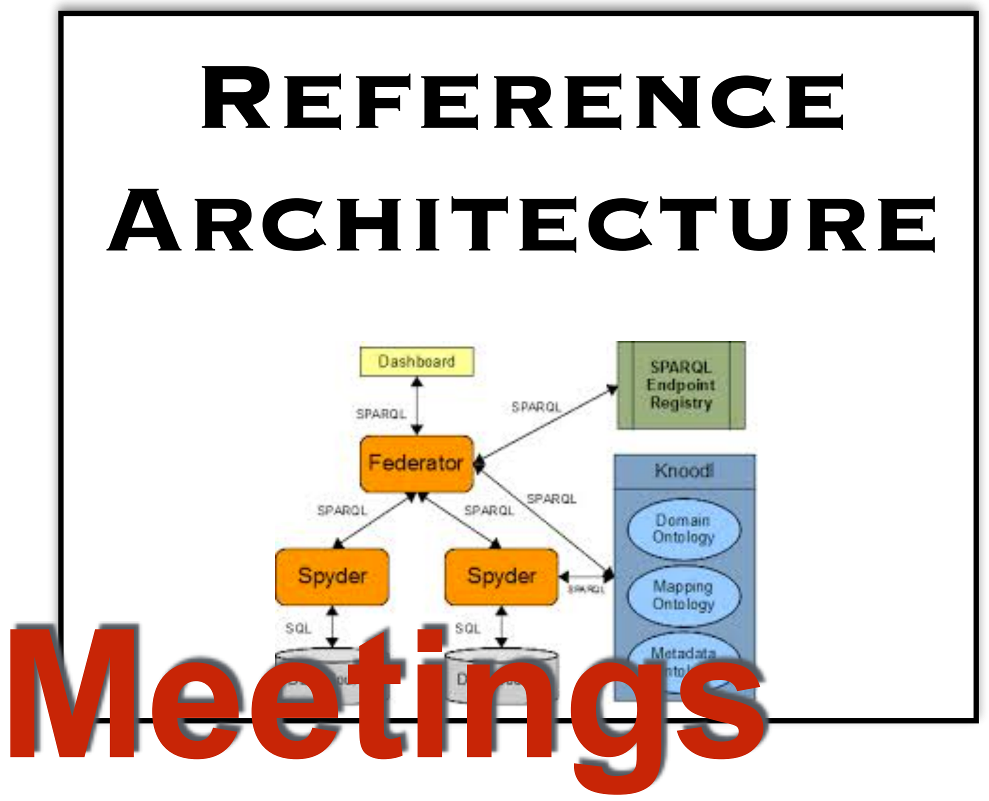
Meetings

STRATEGY
&
BUDGET

TOP SECRET

Meetings

REFERENCE
ARCHITECTURE



Meetings

Meetings

Meetings

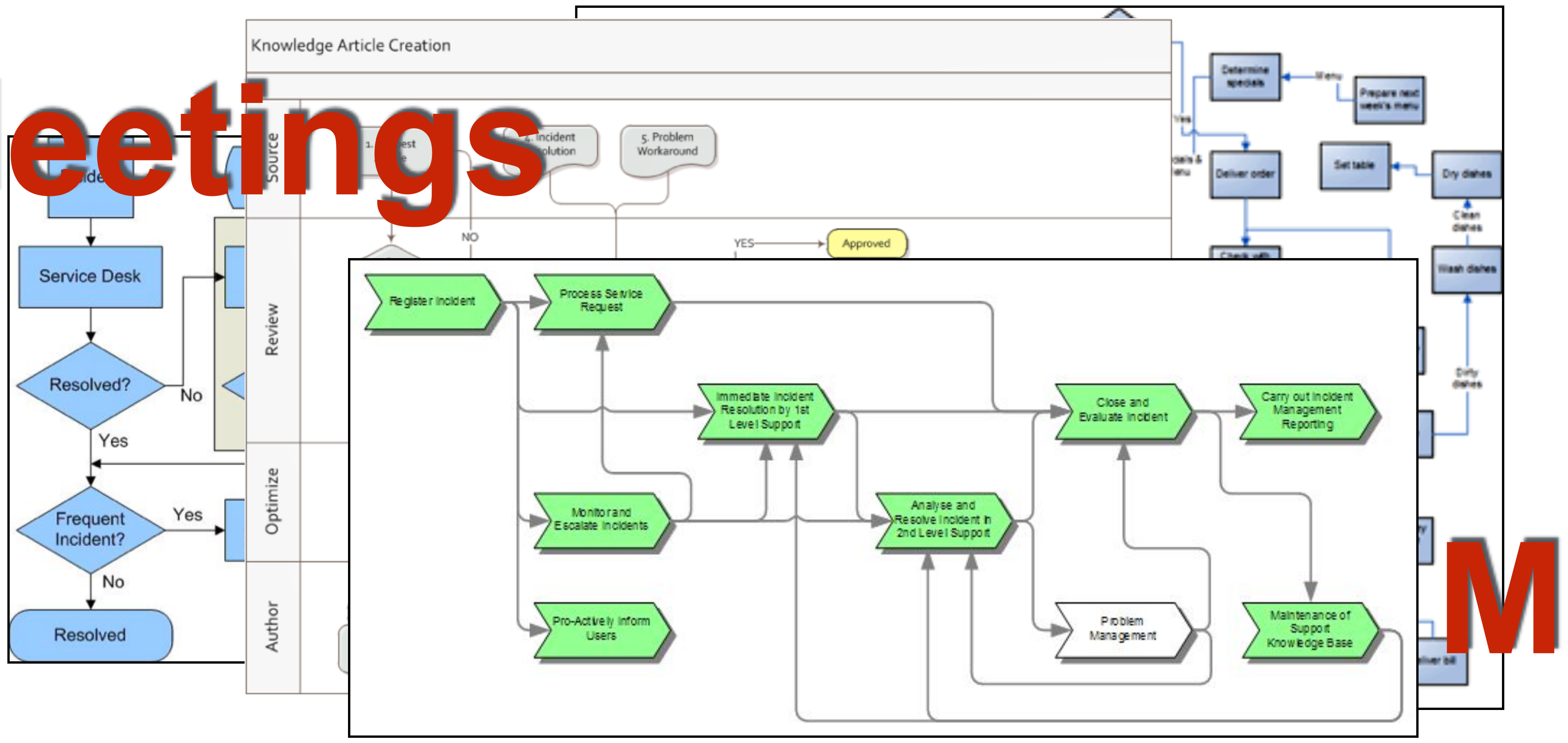
Meetings

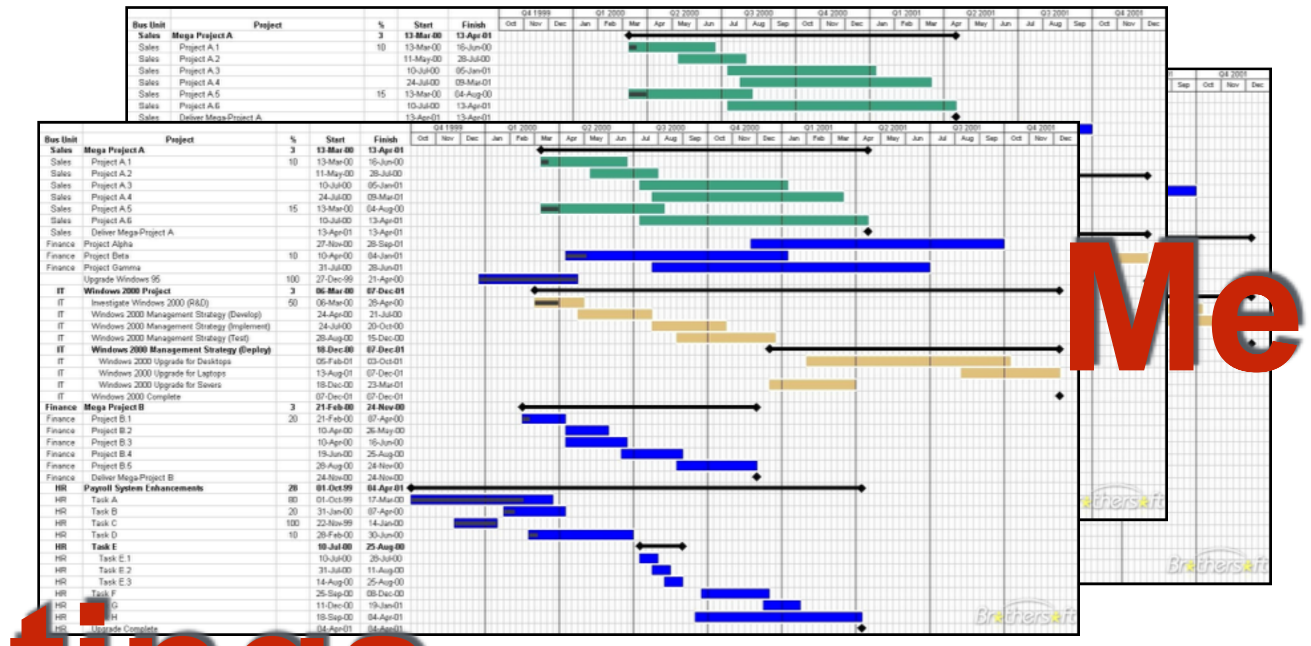
Documented Processes

Project Plans

Release Trains

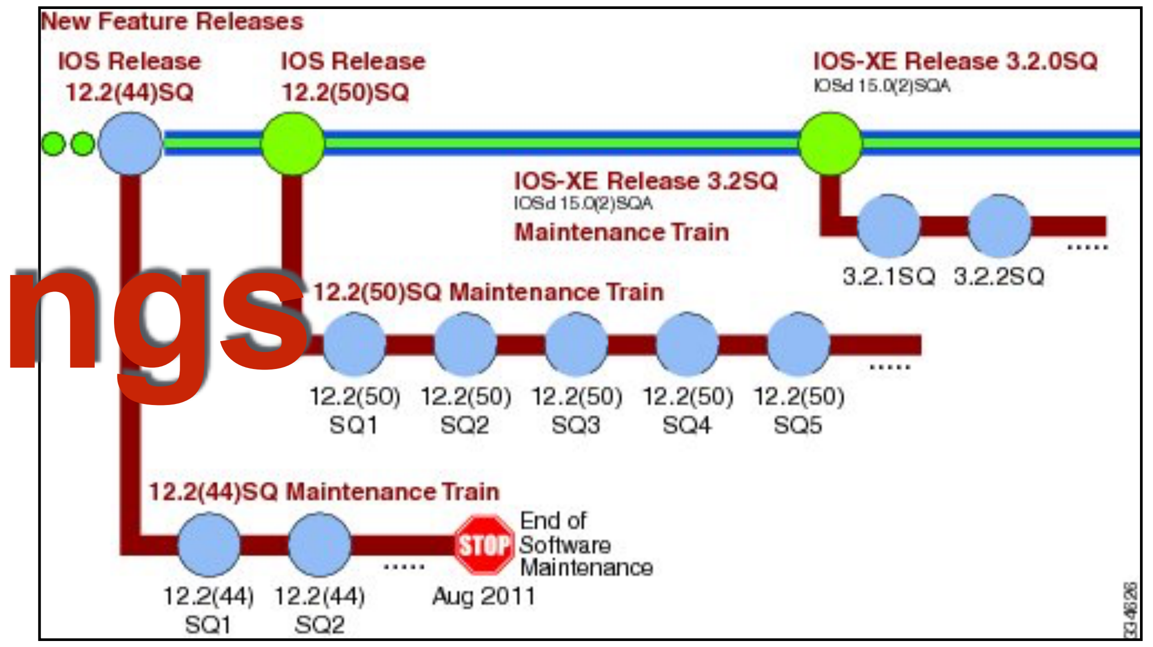
Meetings



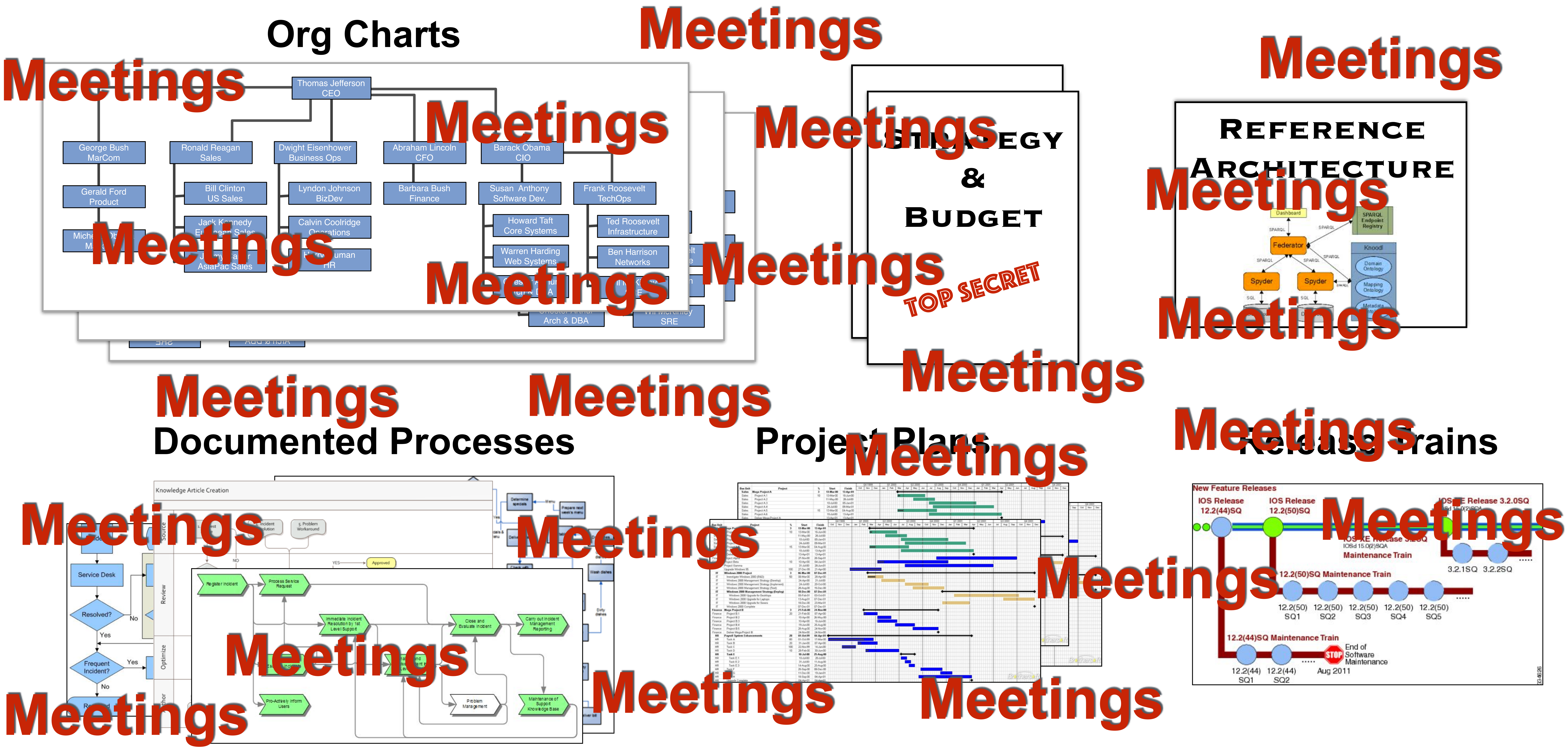


Meetings

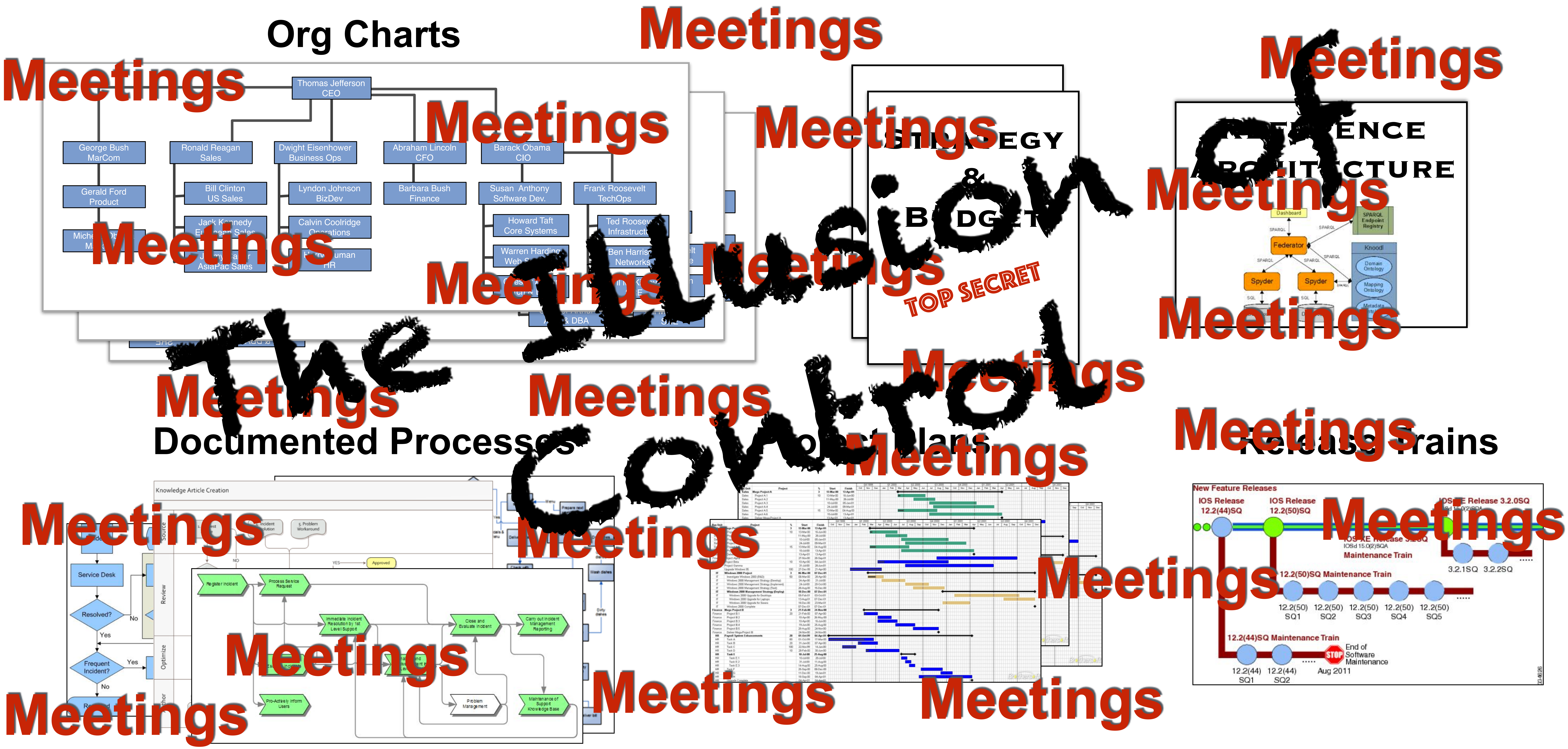
Meetings



Traditional “visibility”



Traditional “visibility”



Org Charts

Documented Processes

Project Plans

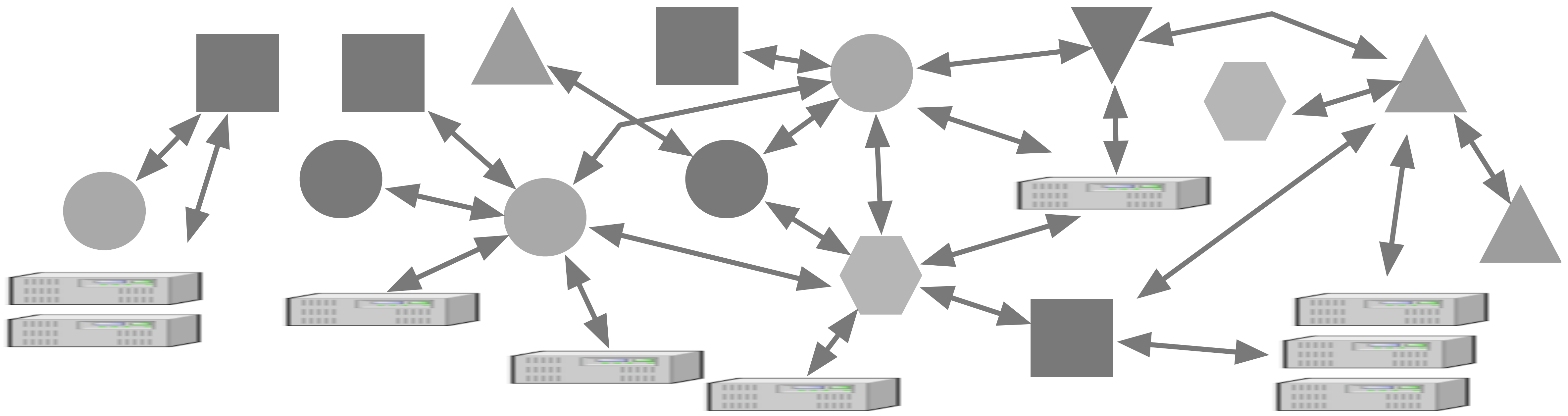
Release Trains

The Illusion of Control

It’s a complex system^2

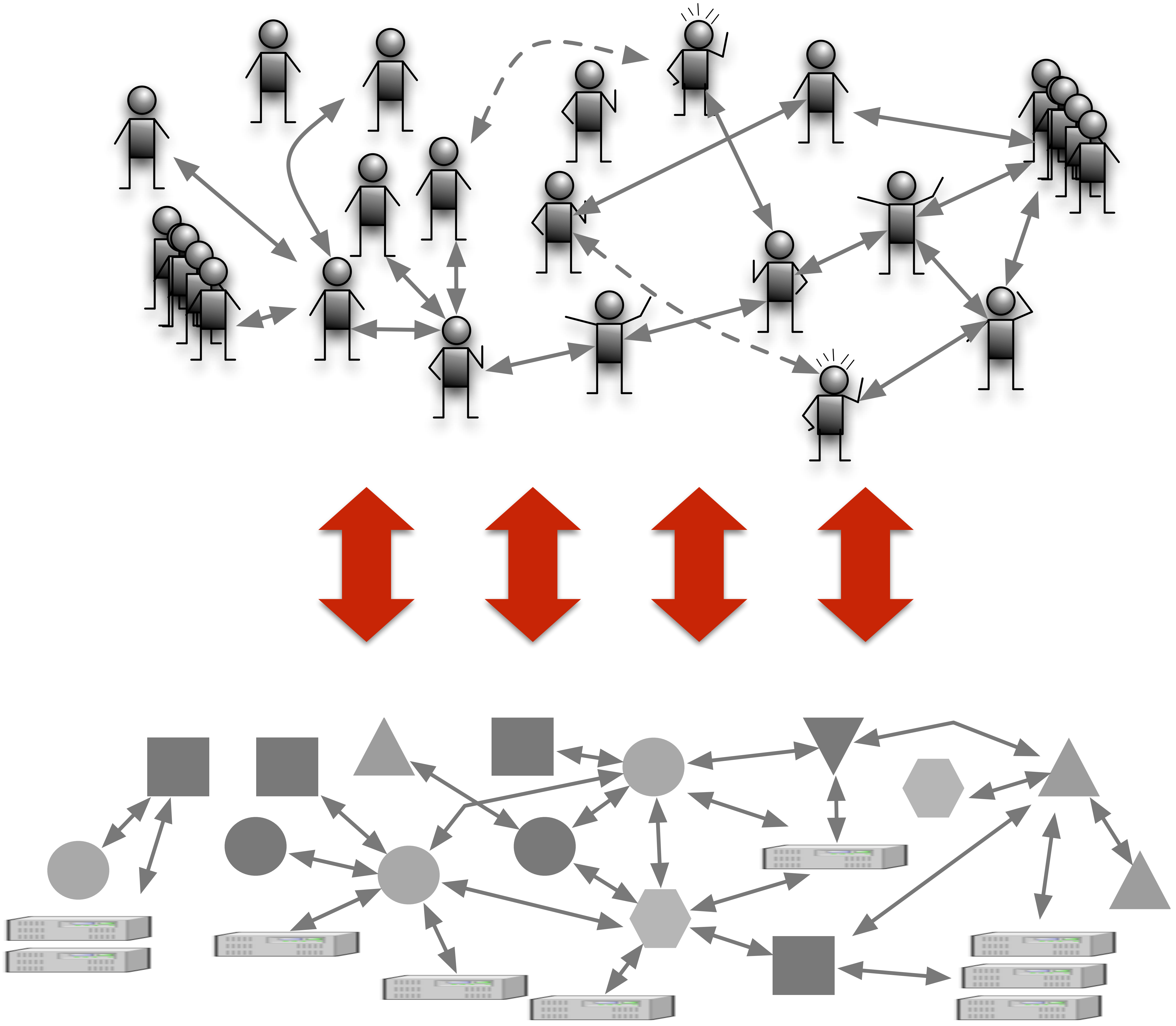
RUNDECK

It’s a complex system^2



Complex System

It’s a complex system^2

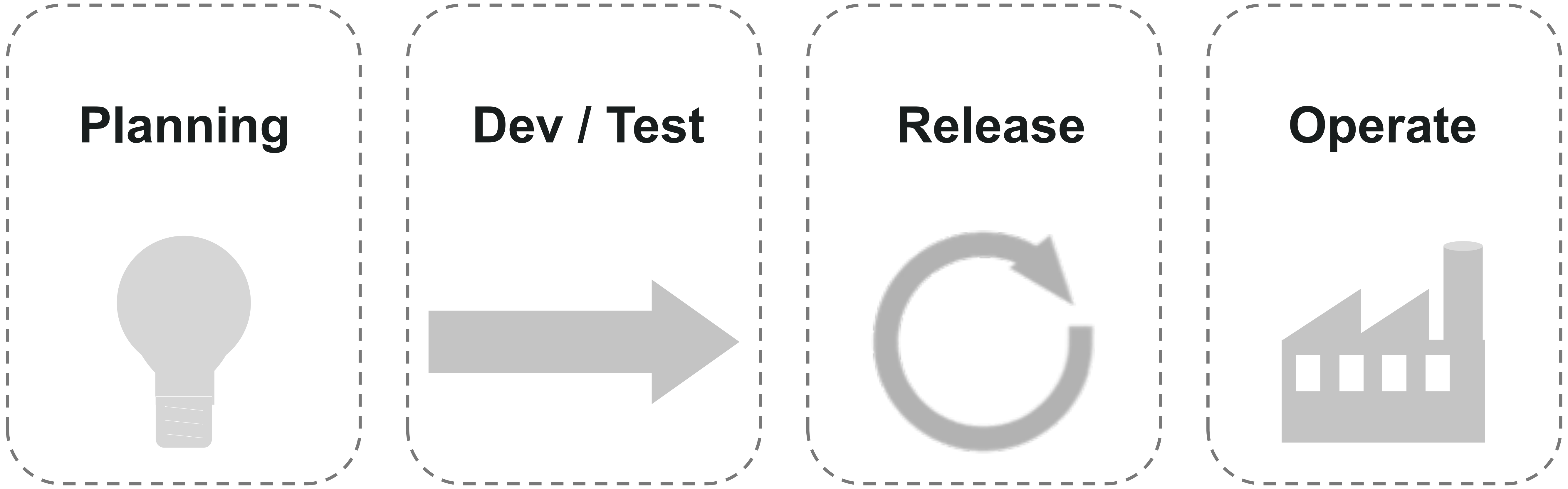


Complex System

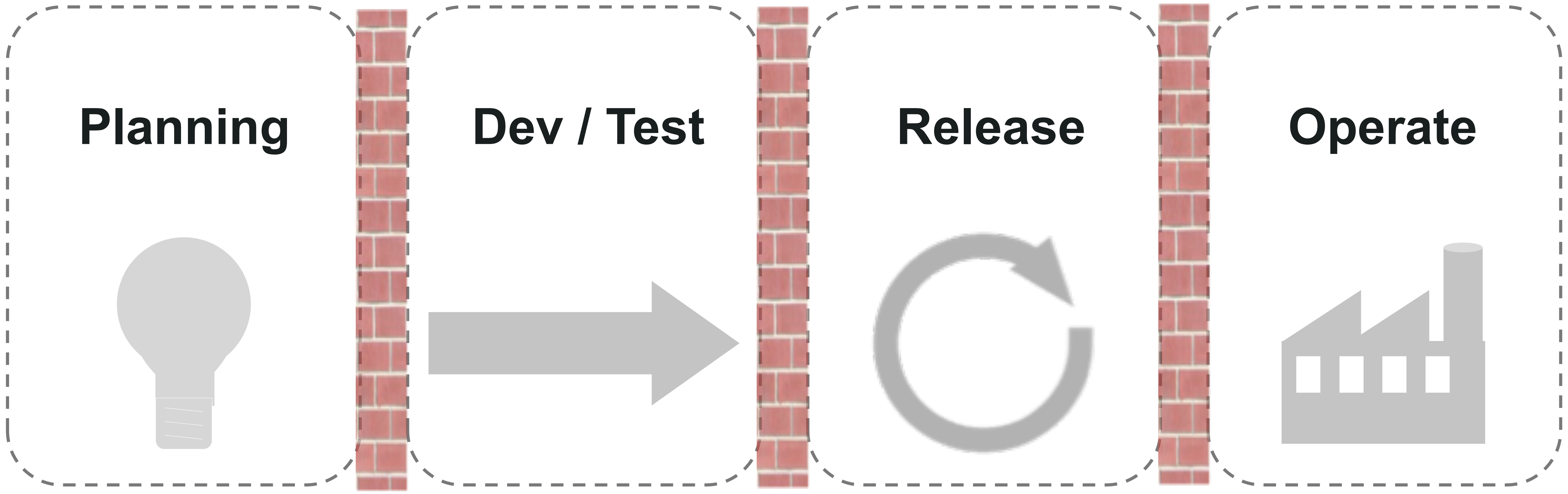
interacting with a

Complex System

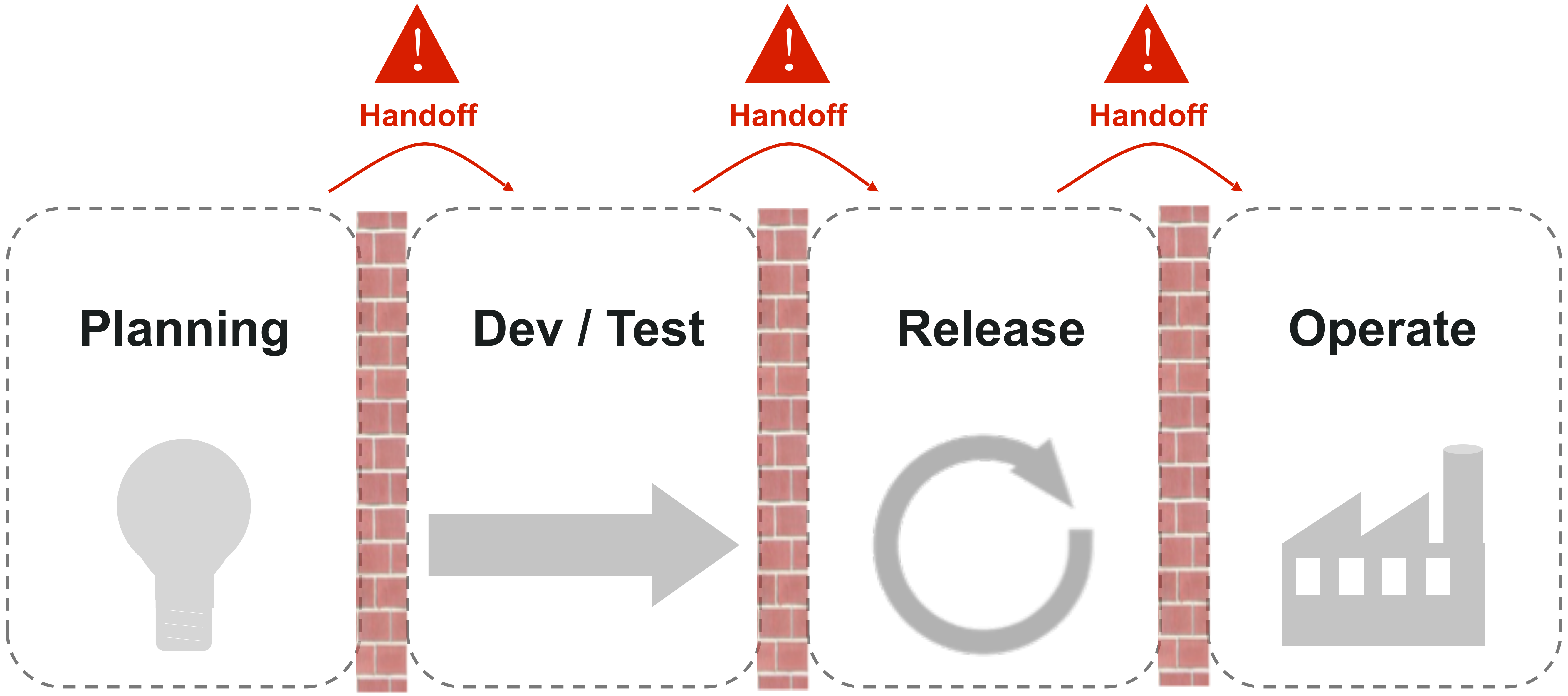
Enterprises naturally trend towards silos



Enterprises naturally trend towards silos

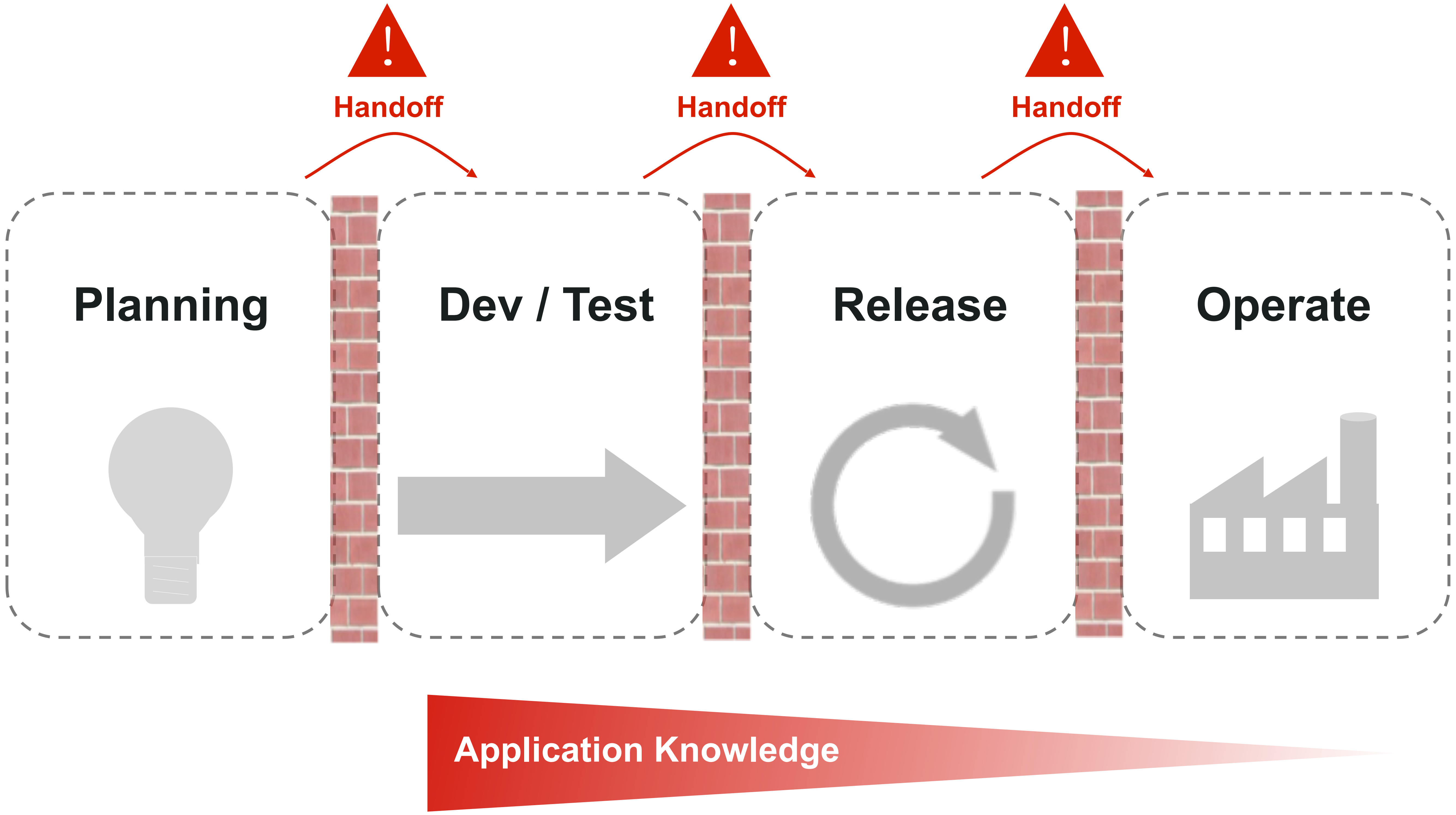


Enterprises naturally trend towards silos



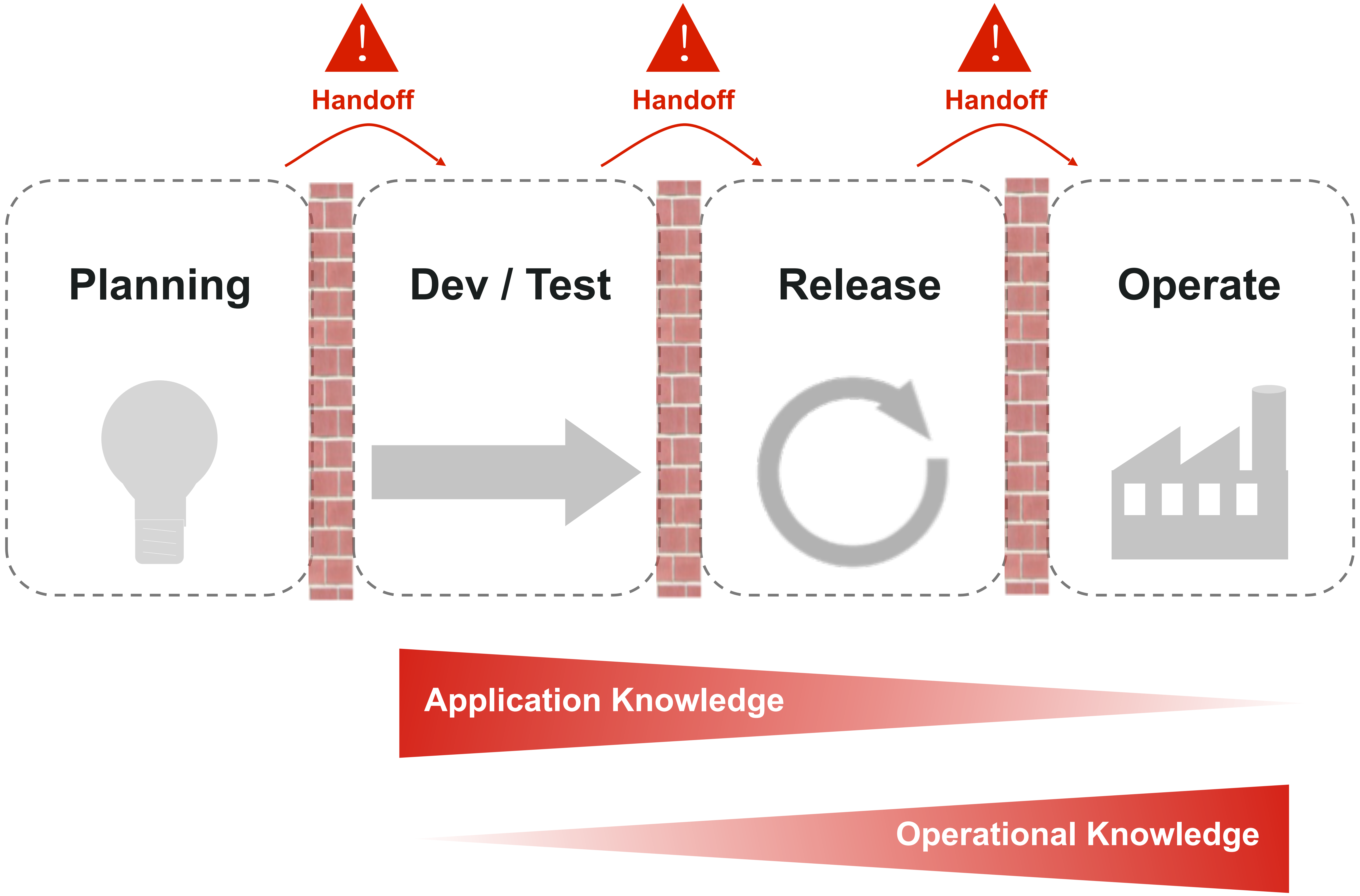
RUNDECK

Enterprises naturally trend towards silos

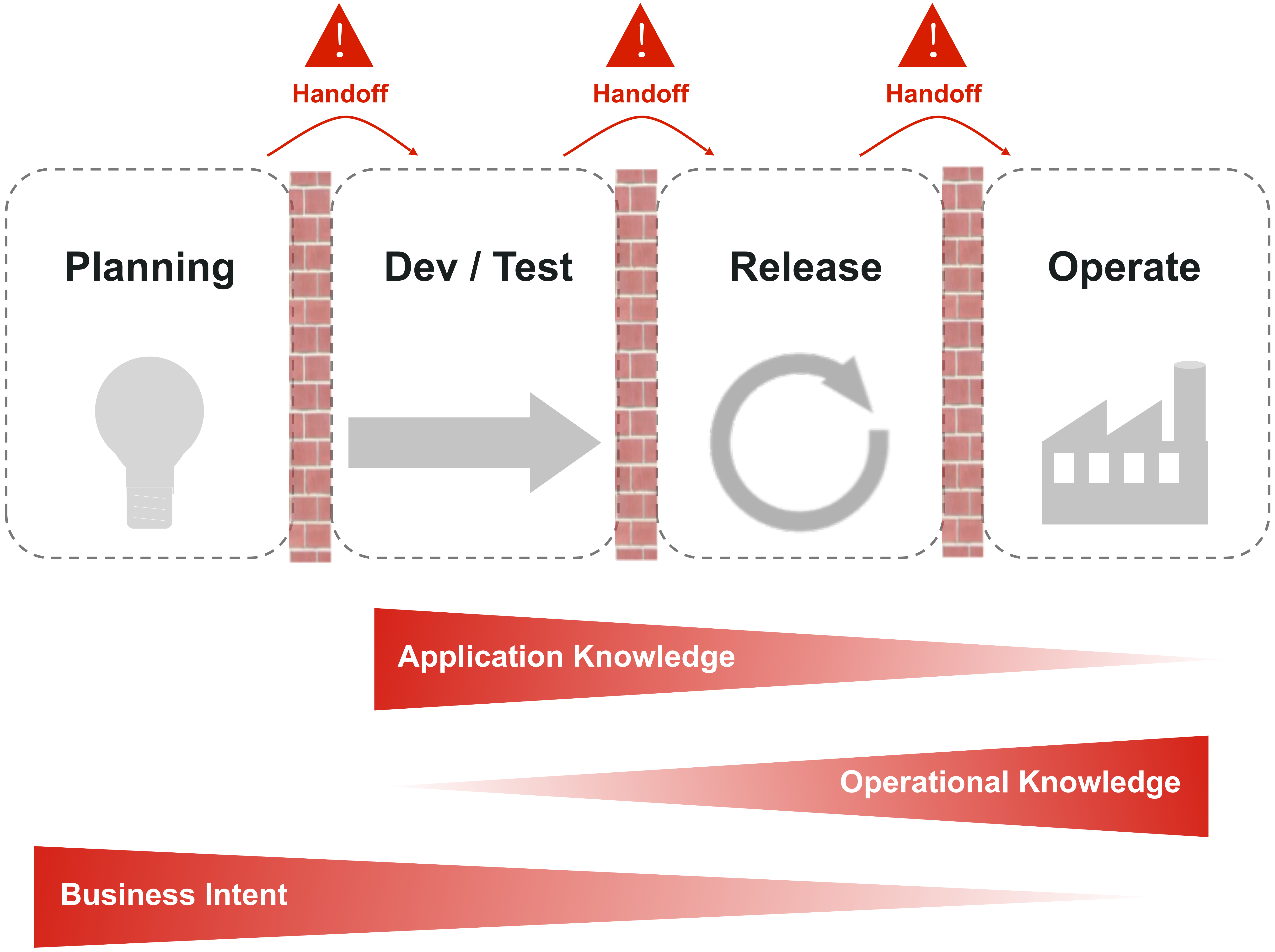


RUNDECK

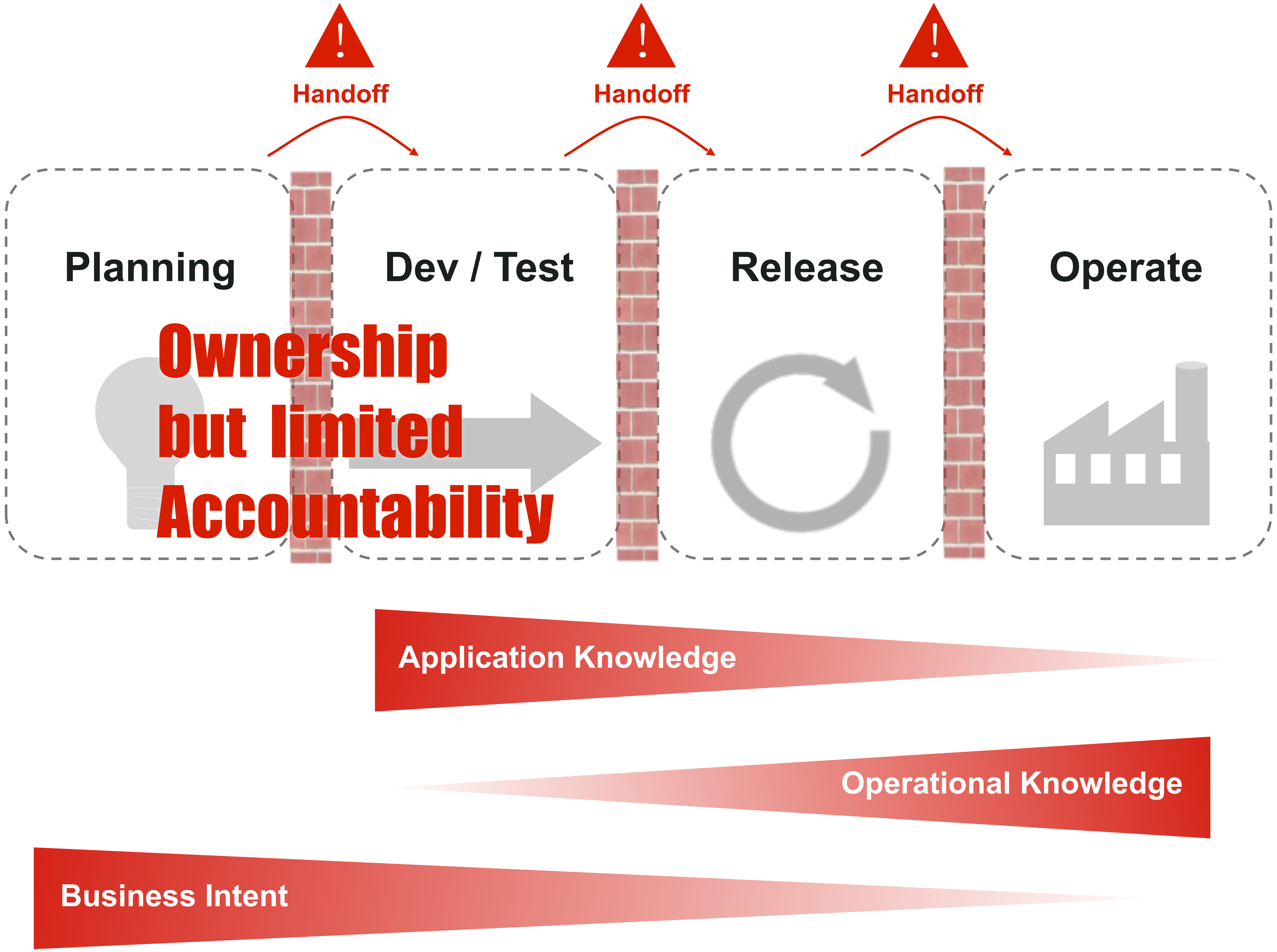
Enterprises naturally trend towards silos



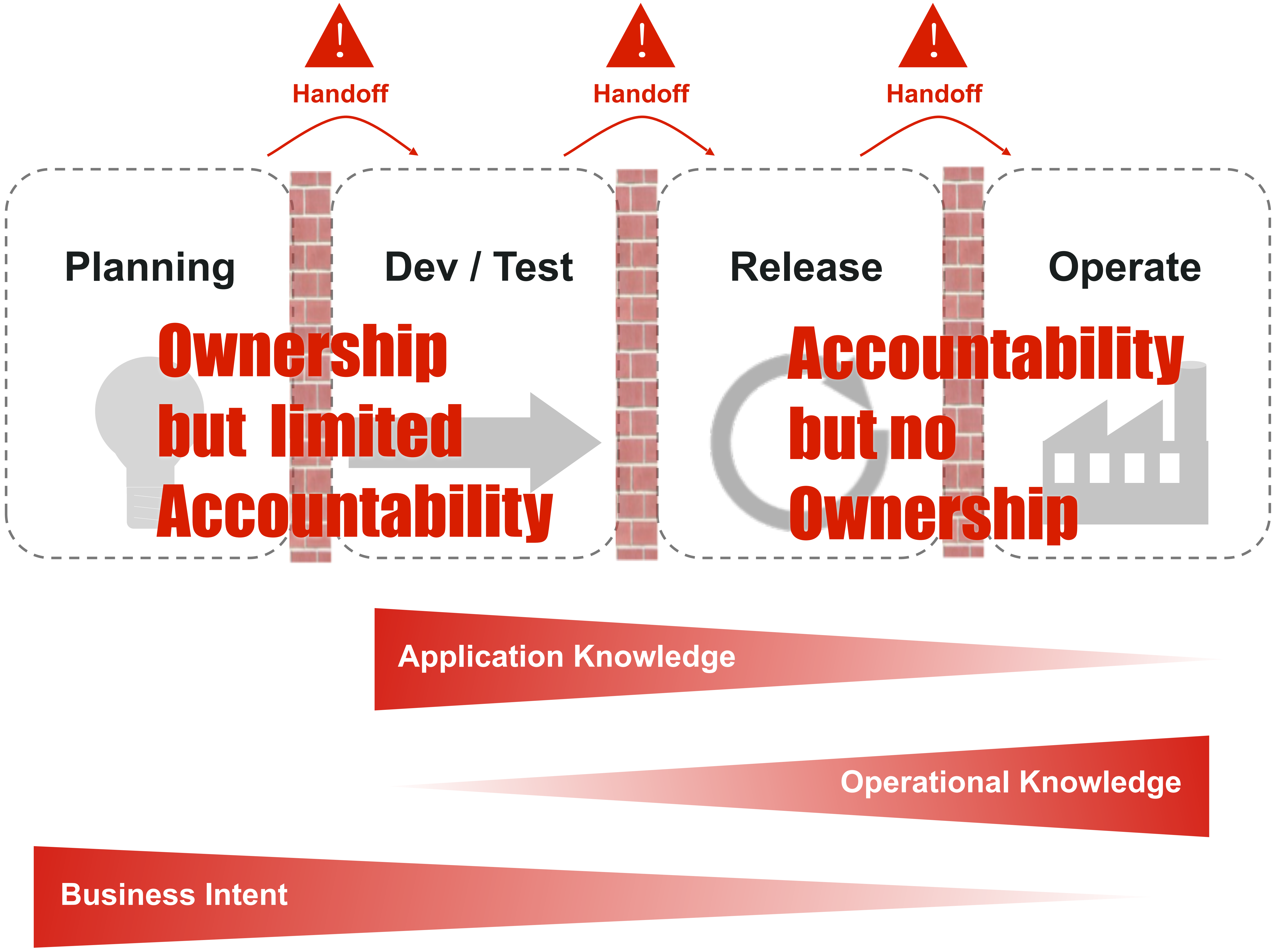
Enterprises naturally trend towards silos



Enterprises naturally trend towards silos

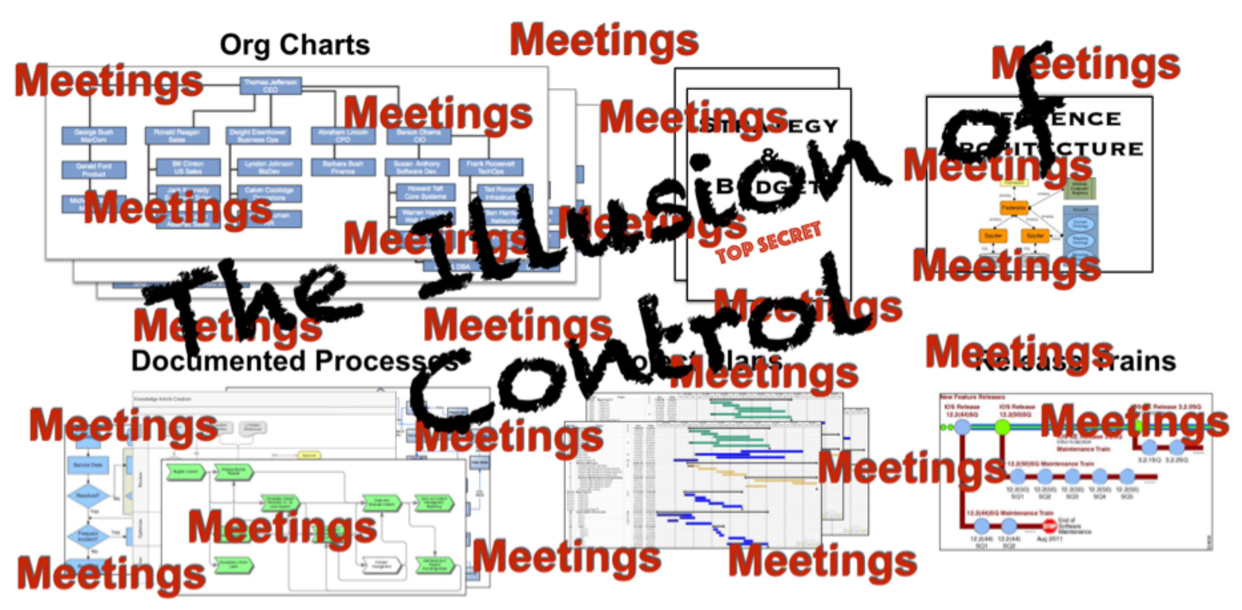


Enterprises naturally trend towards silos

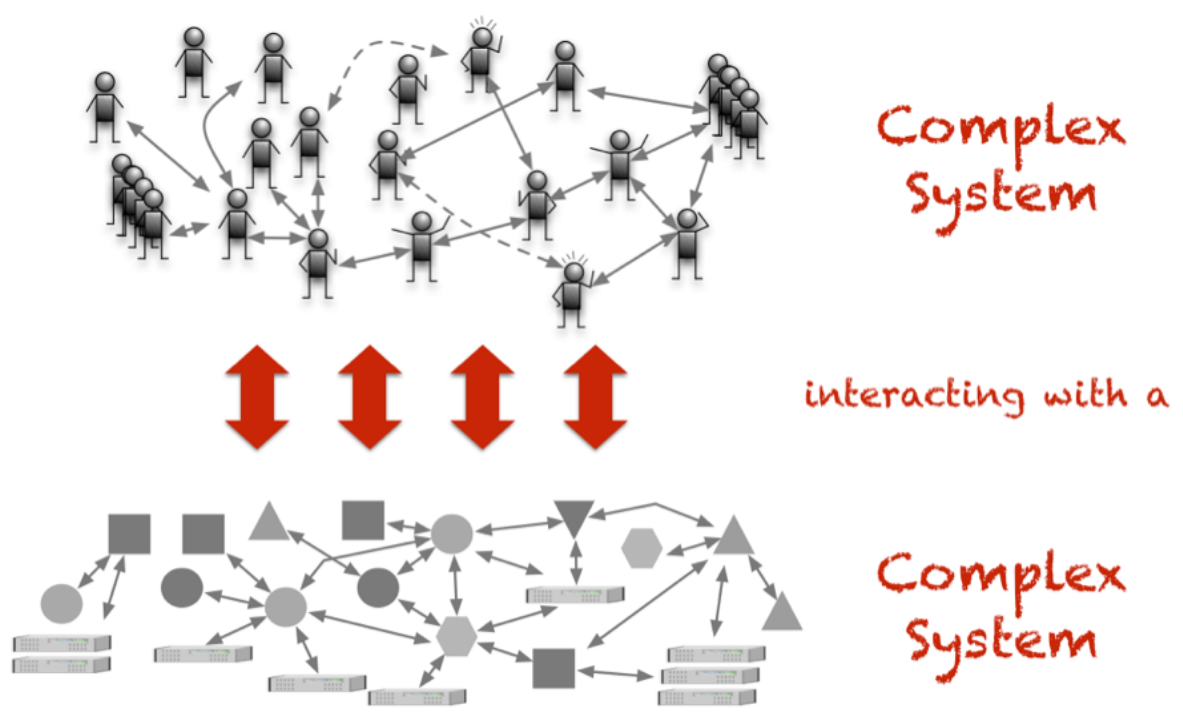


RUNDECK

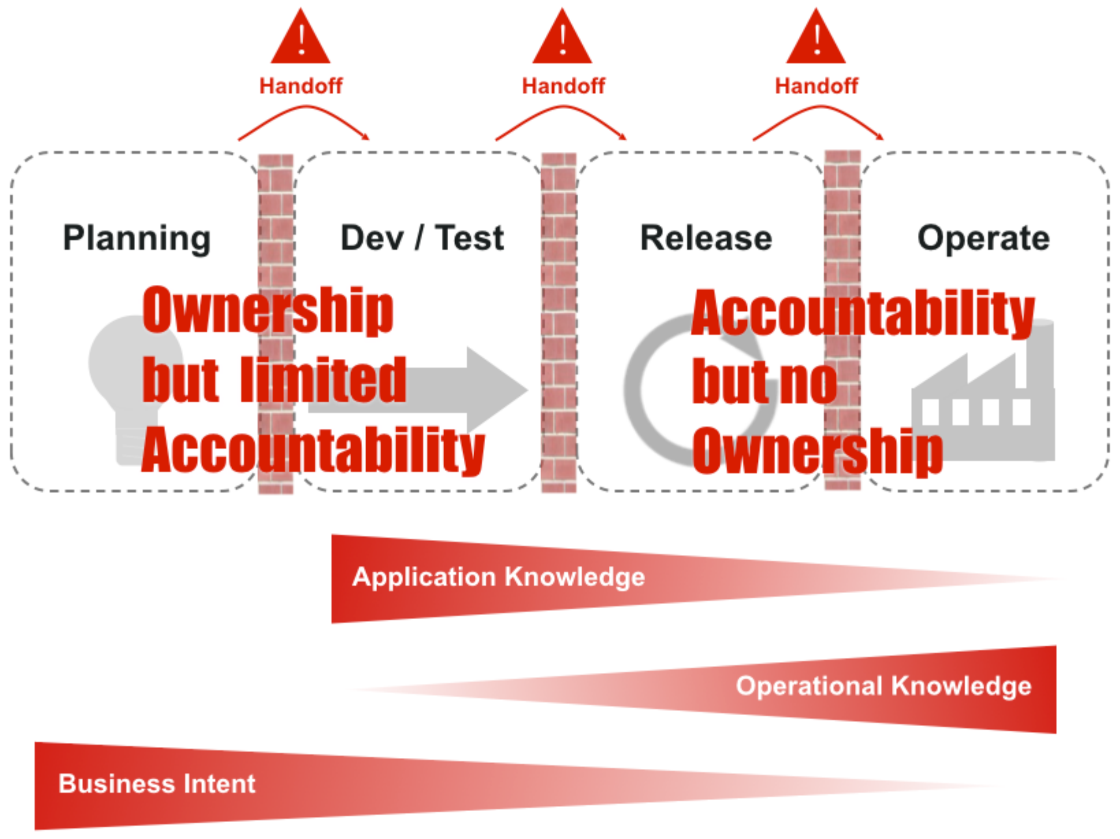
Why are so many organizations unable to improve?



+



+

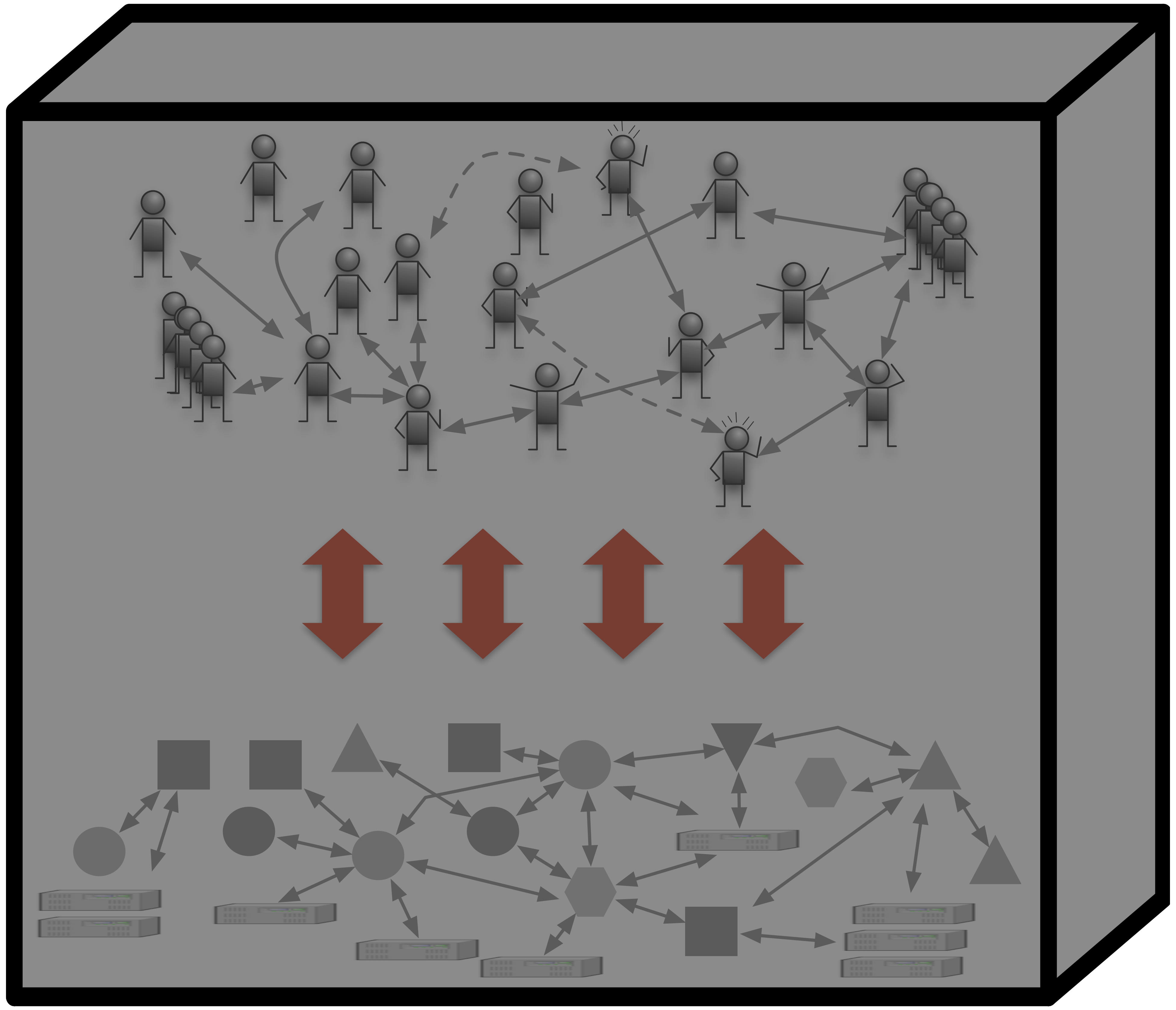


1. The work isn't visible

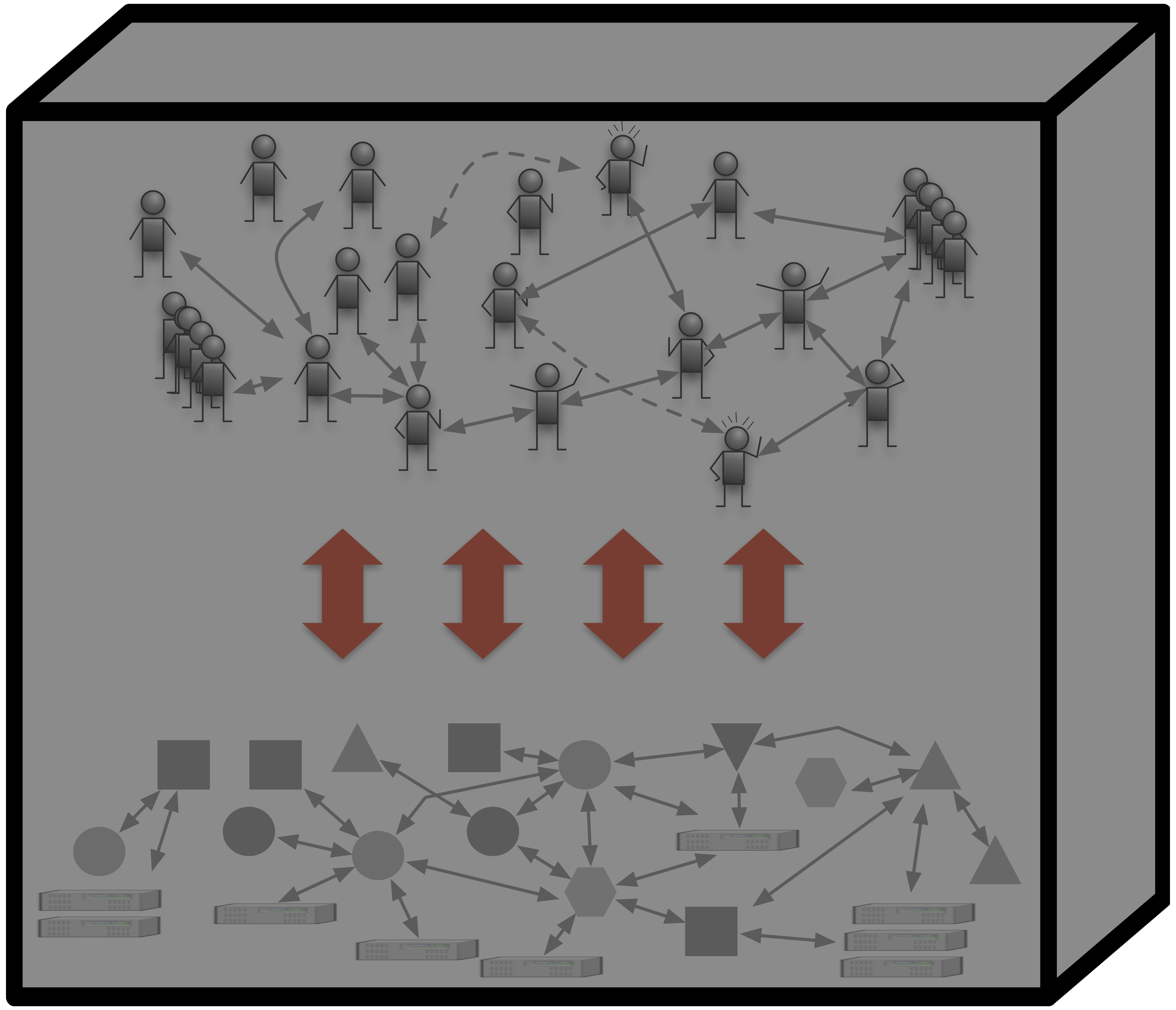
2. People are working out of context

3. Inertia is pulling your org out of alignment

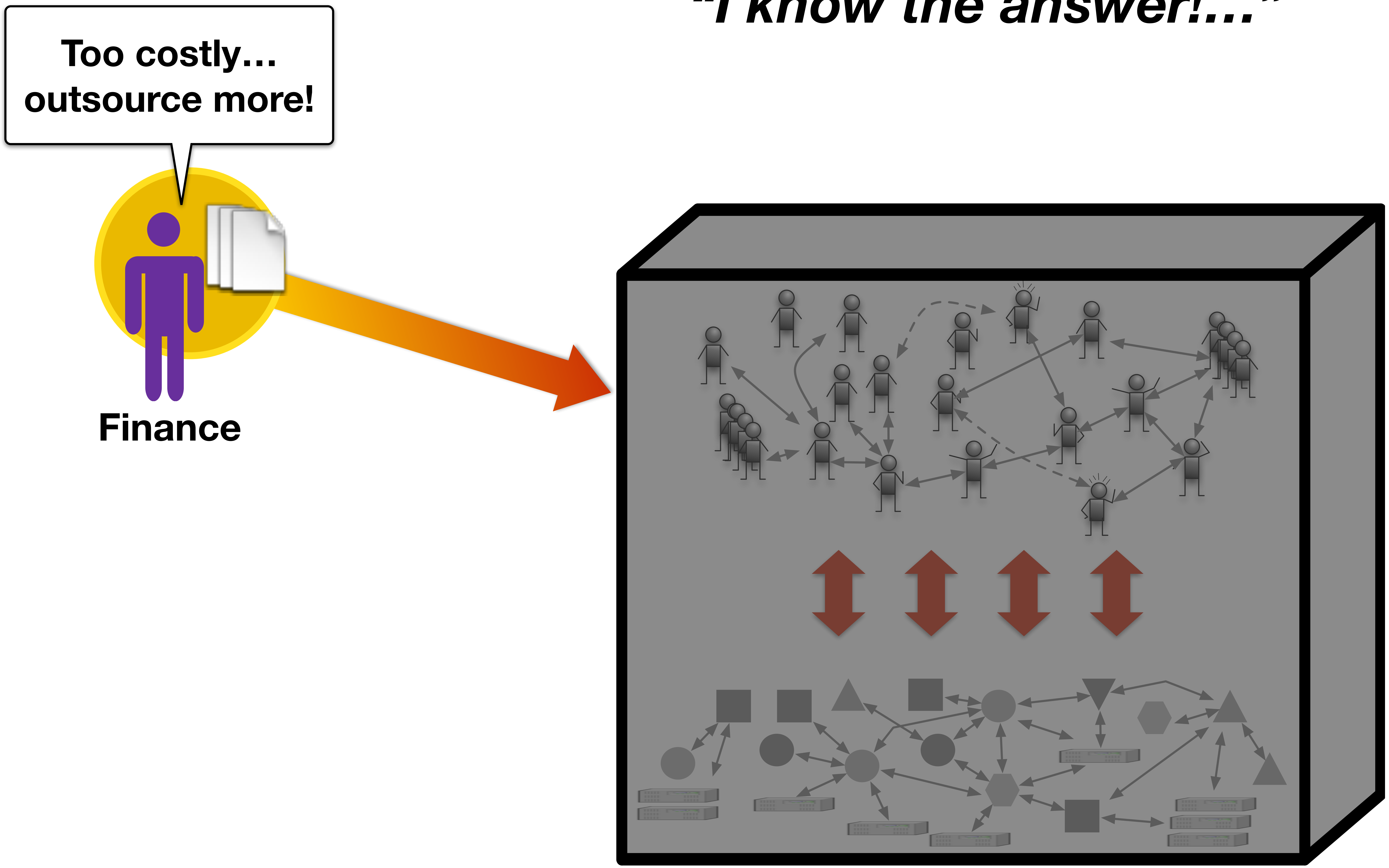
The only way to fix a sufficiently complex system is to create the conditions for the system to fix itself.



“I know the answer!…”



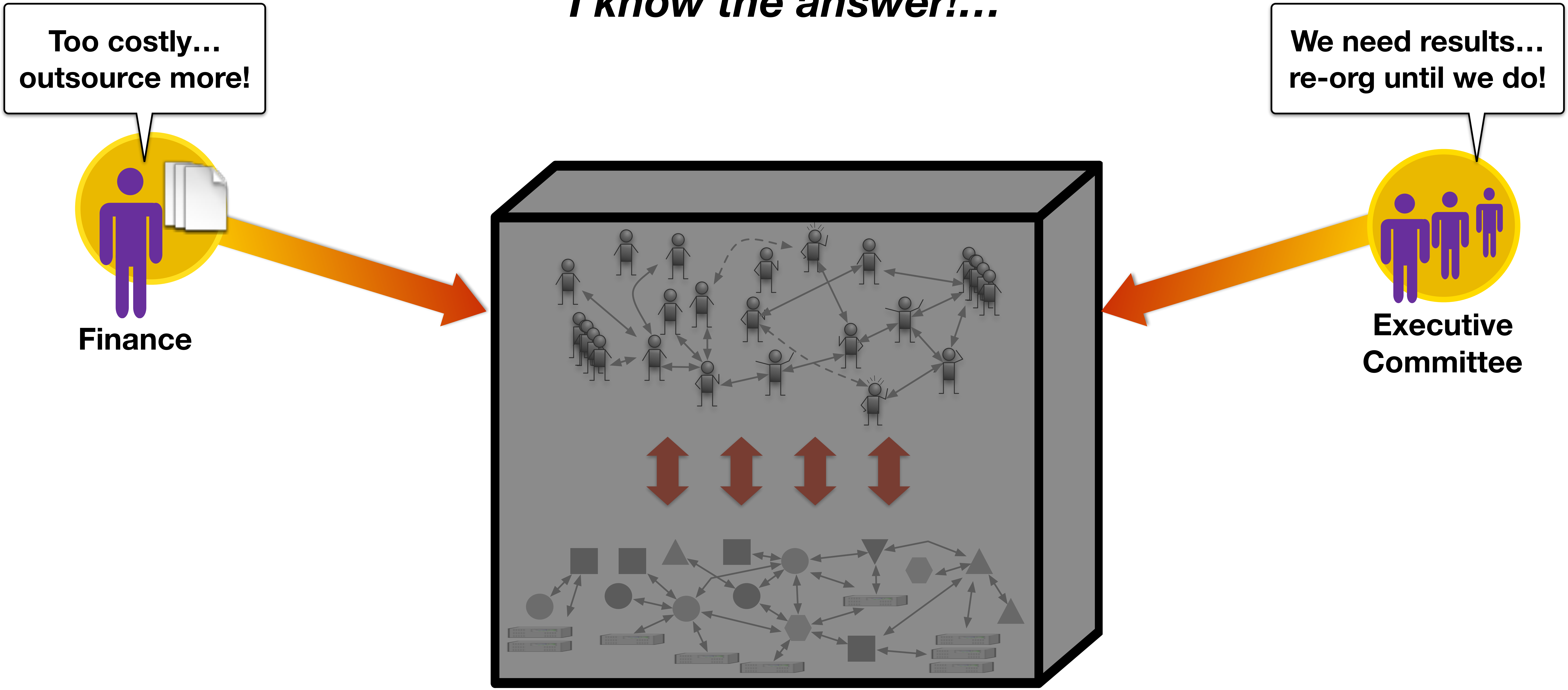
“I know the answer!…”



Too costly…
outsource more!

Finance

“I know the answer!…”



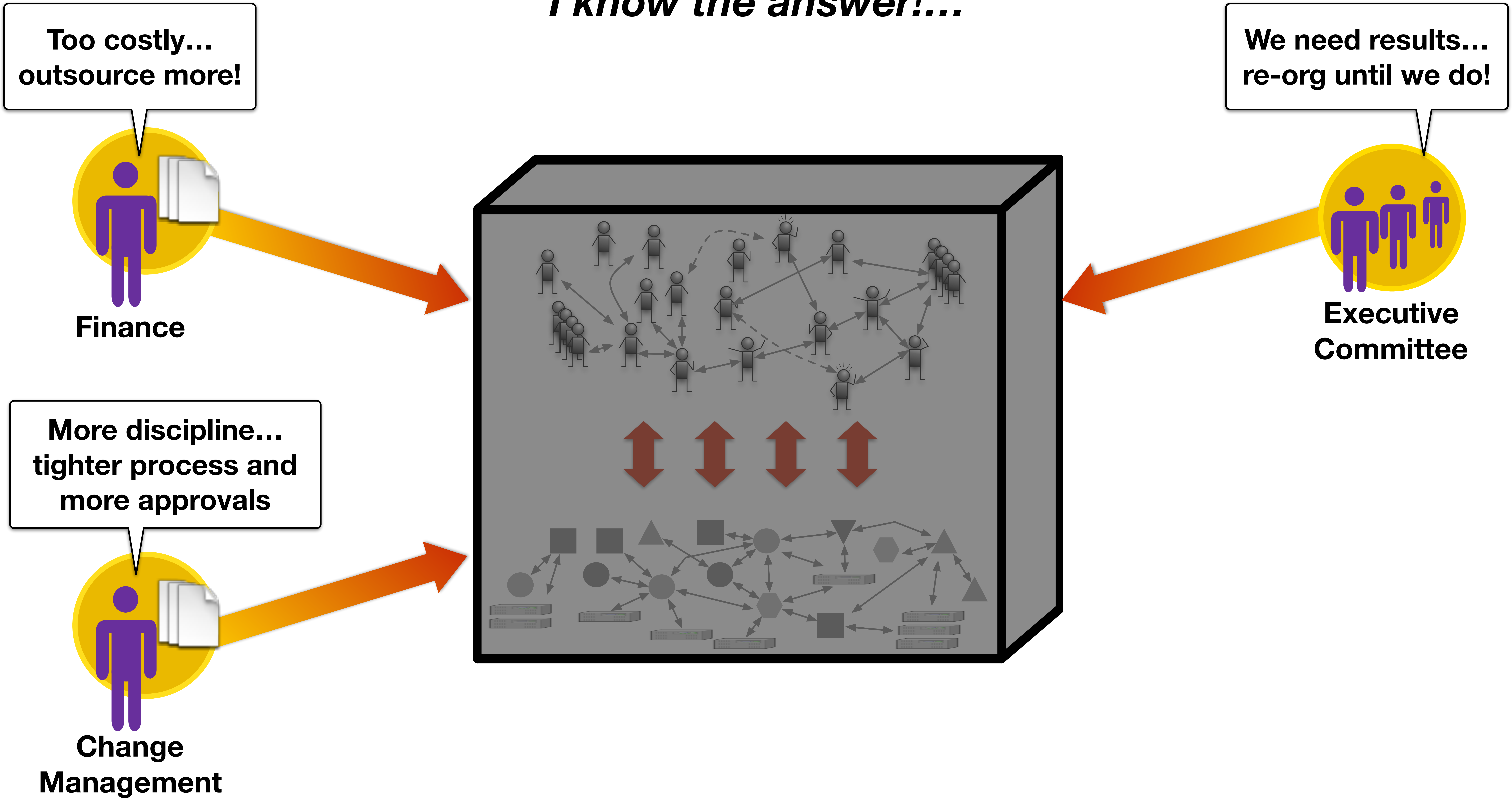
Too costly… outsource more!

Finance

We need results… re-org until we do!

Executive Committee

“I know the answer!…”



Too costly… outsource more!

Finance

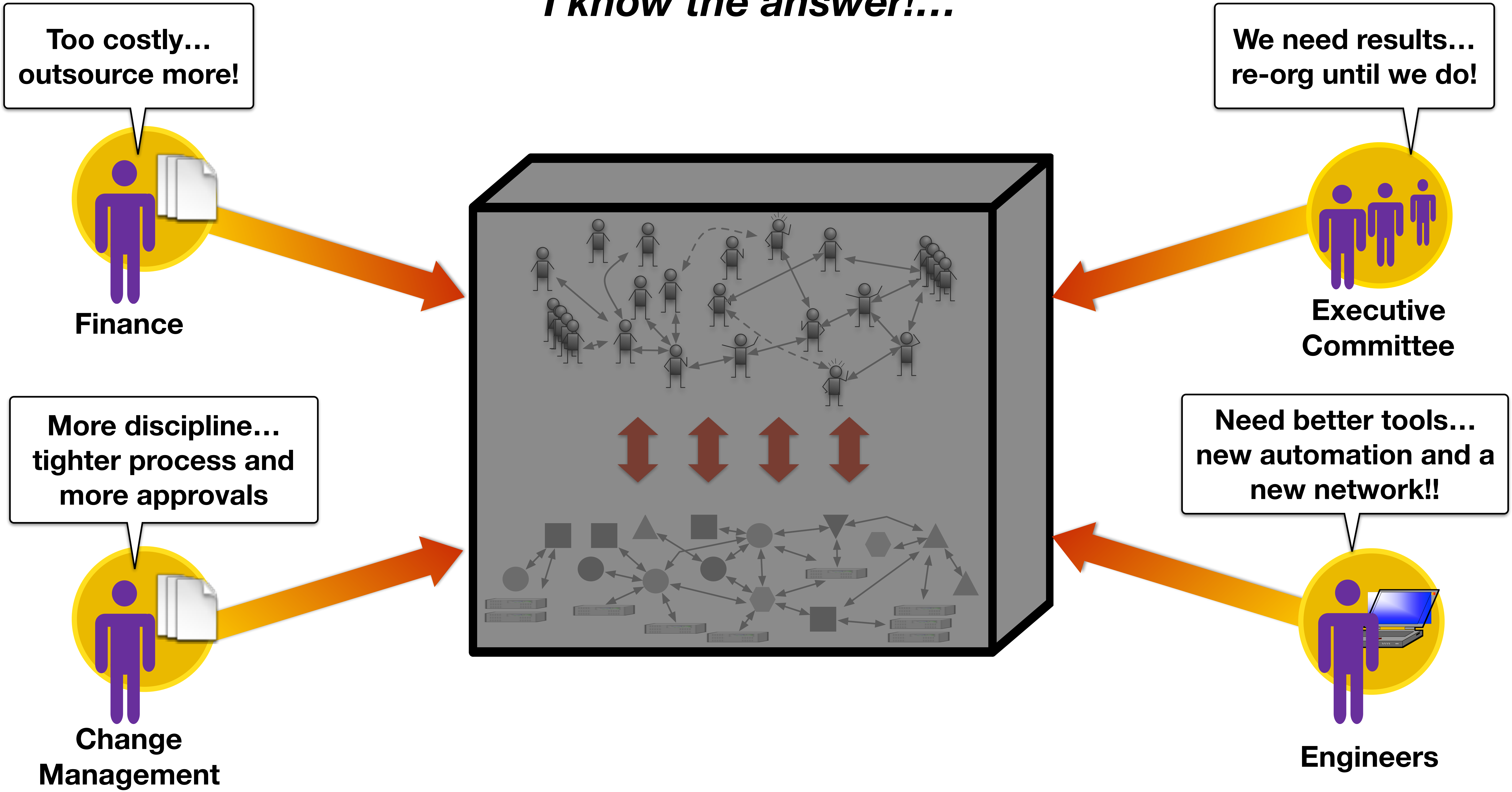
More discipline… tighter process and more approvals

Change Management

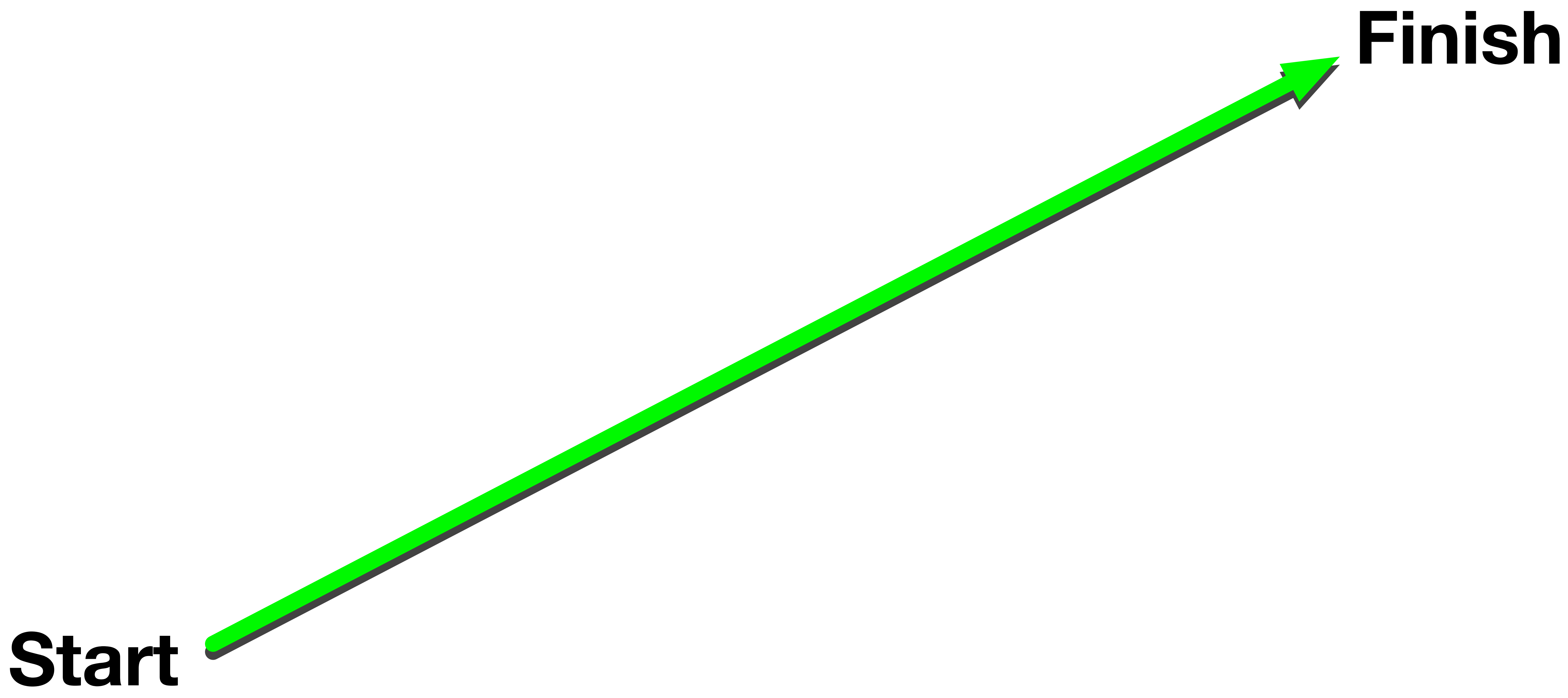
We need results… re-org until we do!

Executive Committee

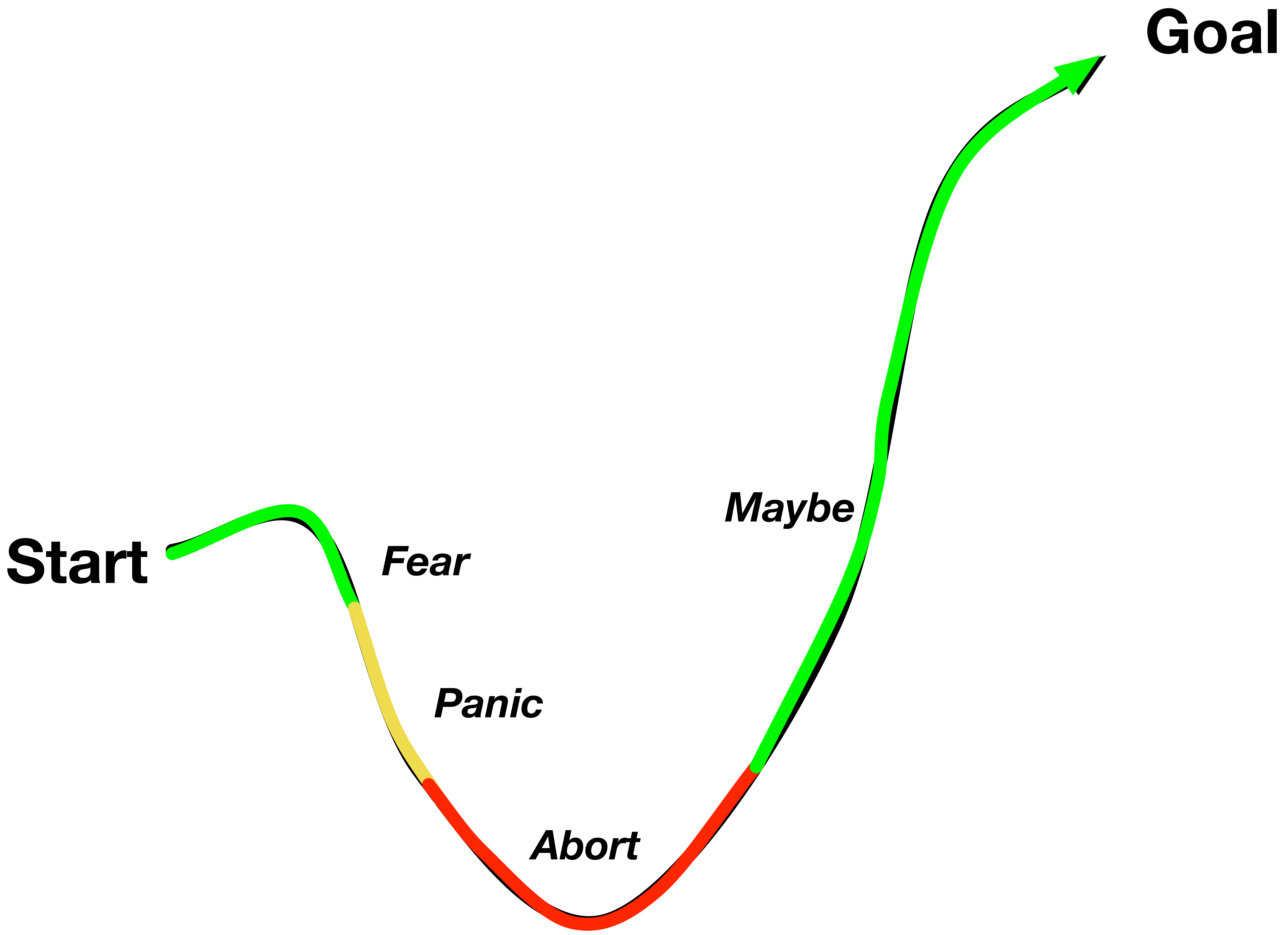
"I know the answer!..."



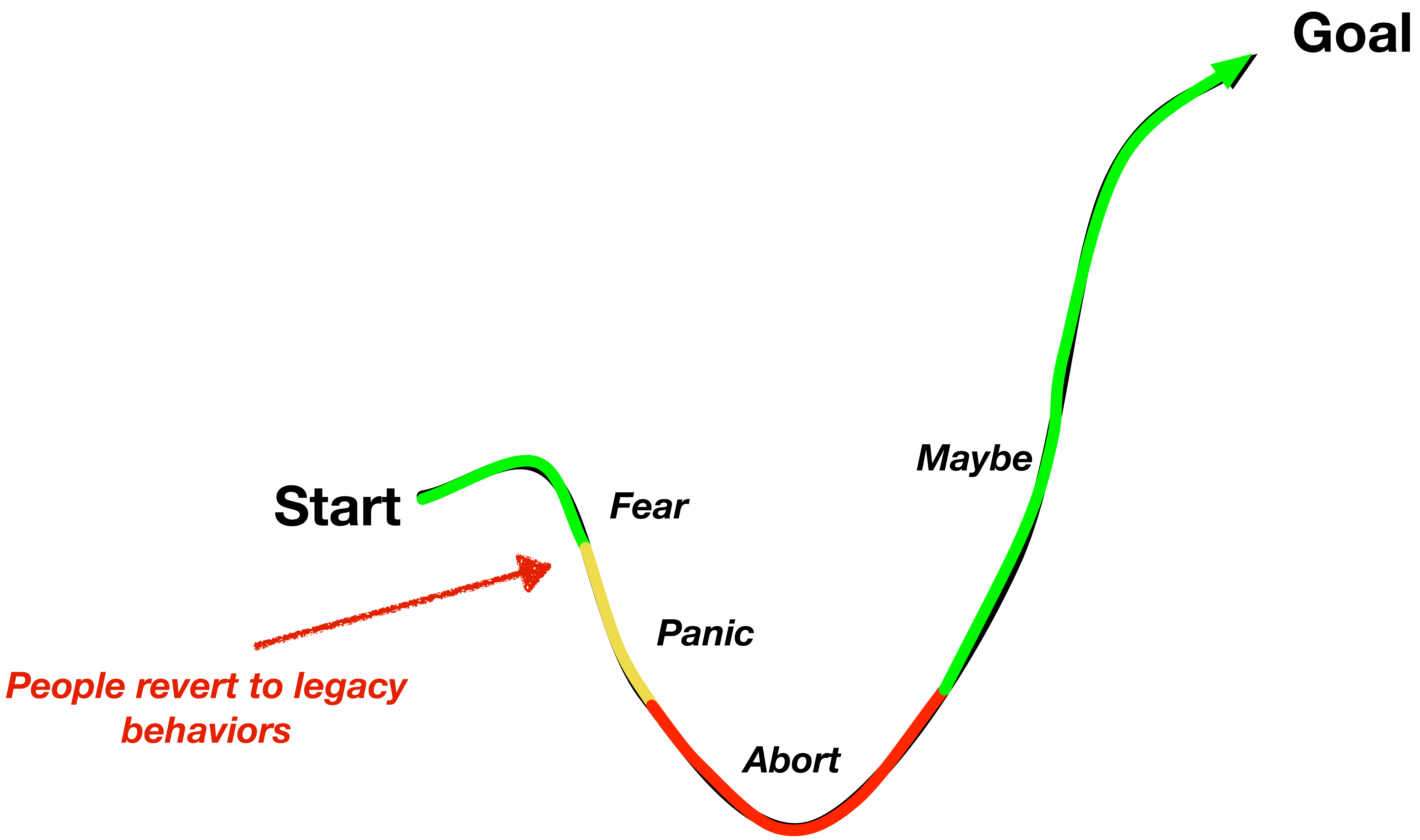
The “Big Bang” Transformation Dream



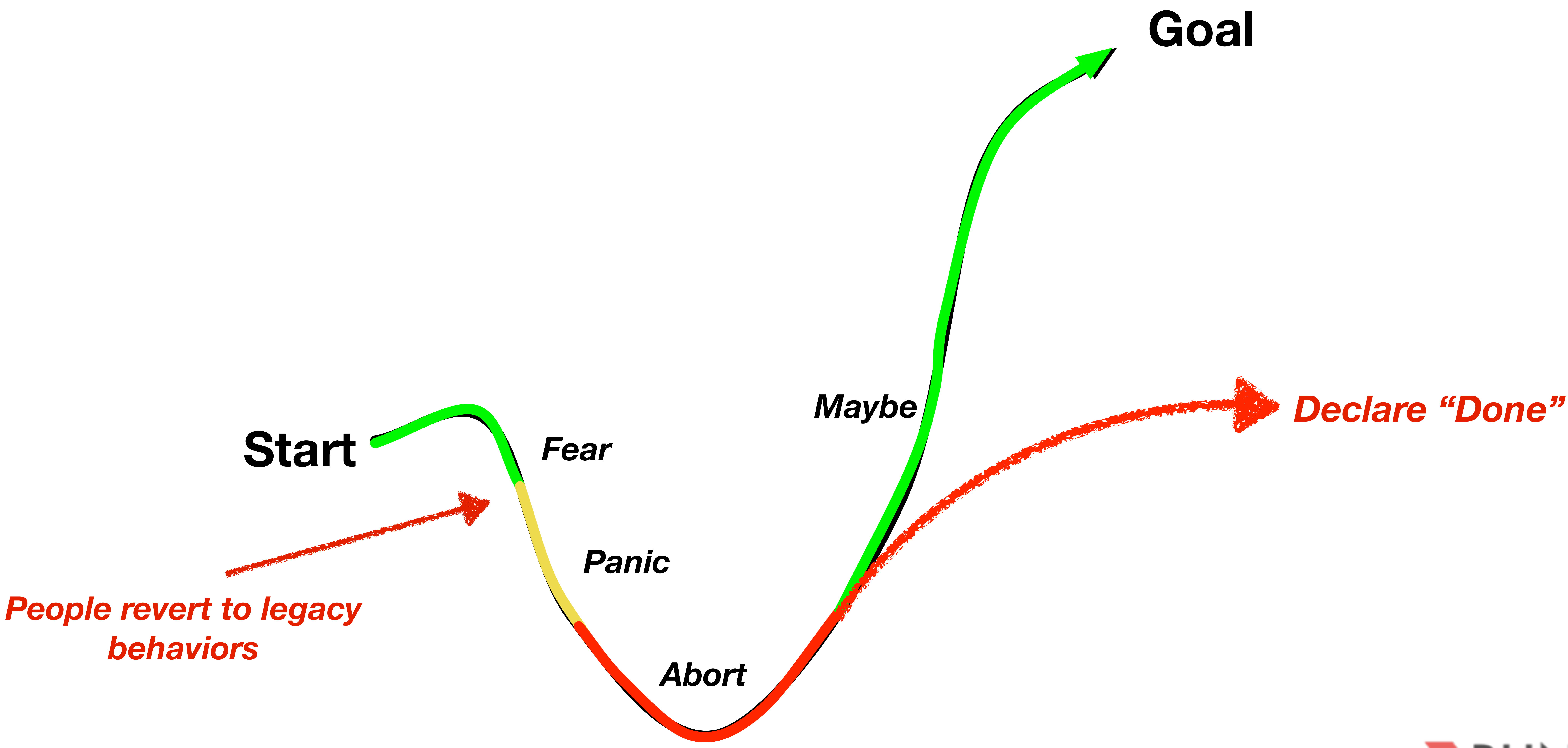
The “Big Bang” Transformation Reality



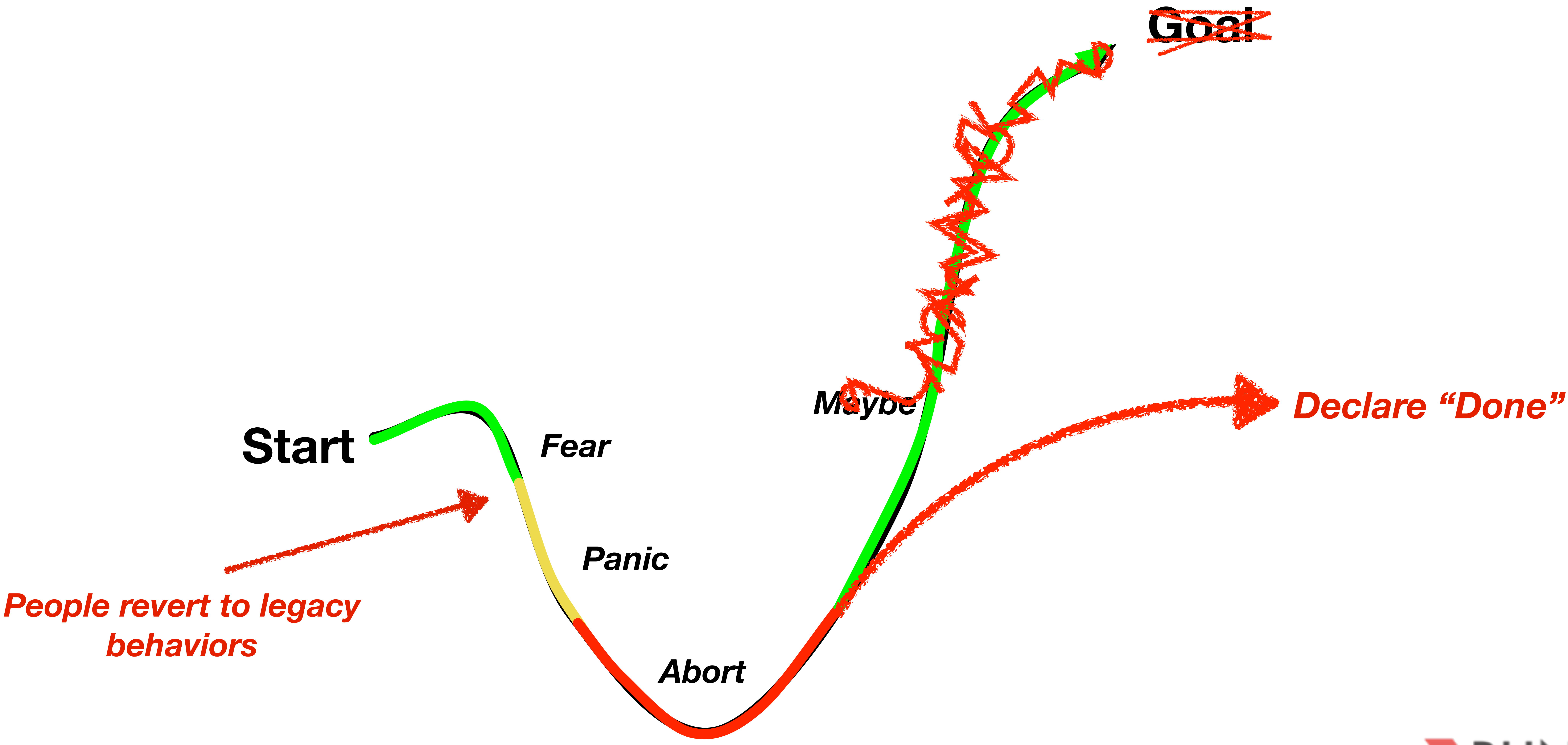
The “Big Bang” Transformation Reality



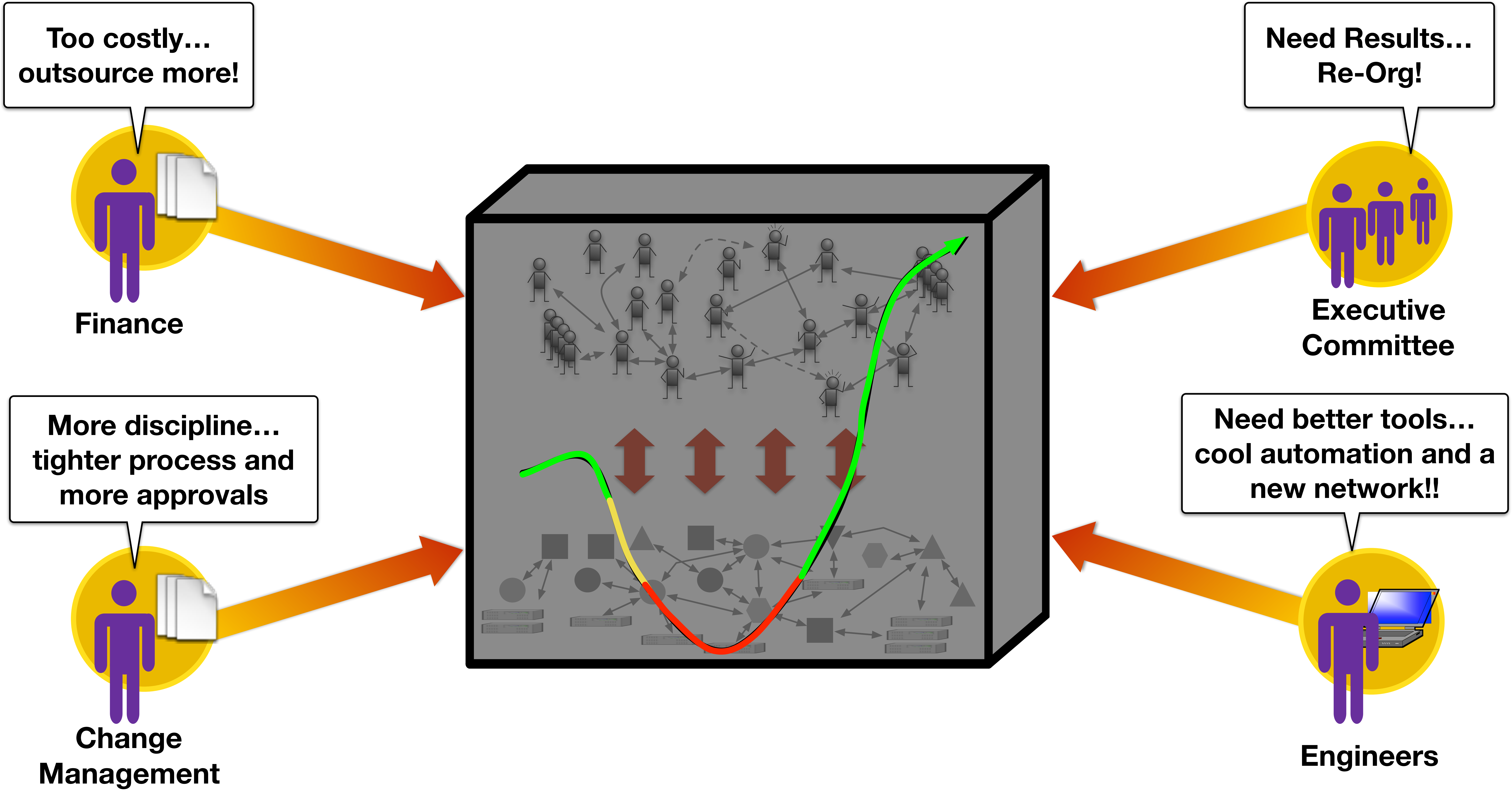
The “Big Bang” Transformation Reality



The “Big Bang” Transformation Reality



RUNDECK



Too costly… outsource more!

Finance

Need Results… Re-Org!

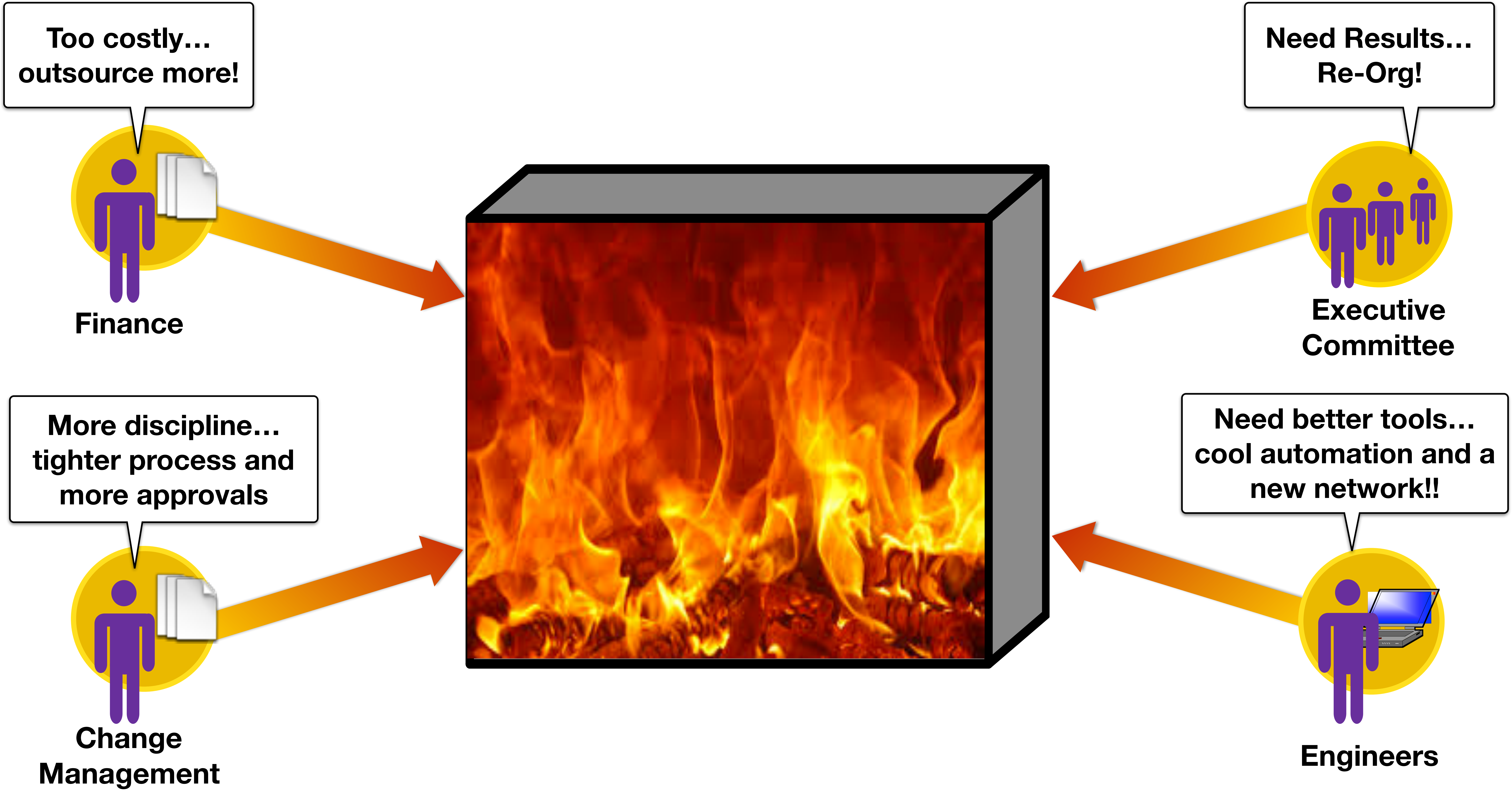
Executive Committee

More discipline… tighter process and more approvals

Change Management

Need better tools… cool automation and a new network!!

Engineers



Too costly… outsource more!

Finance

Need Results… Re-Org!

Executive Committee

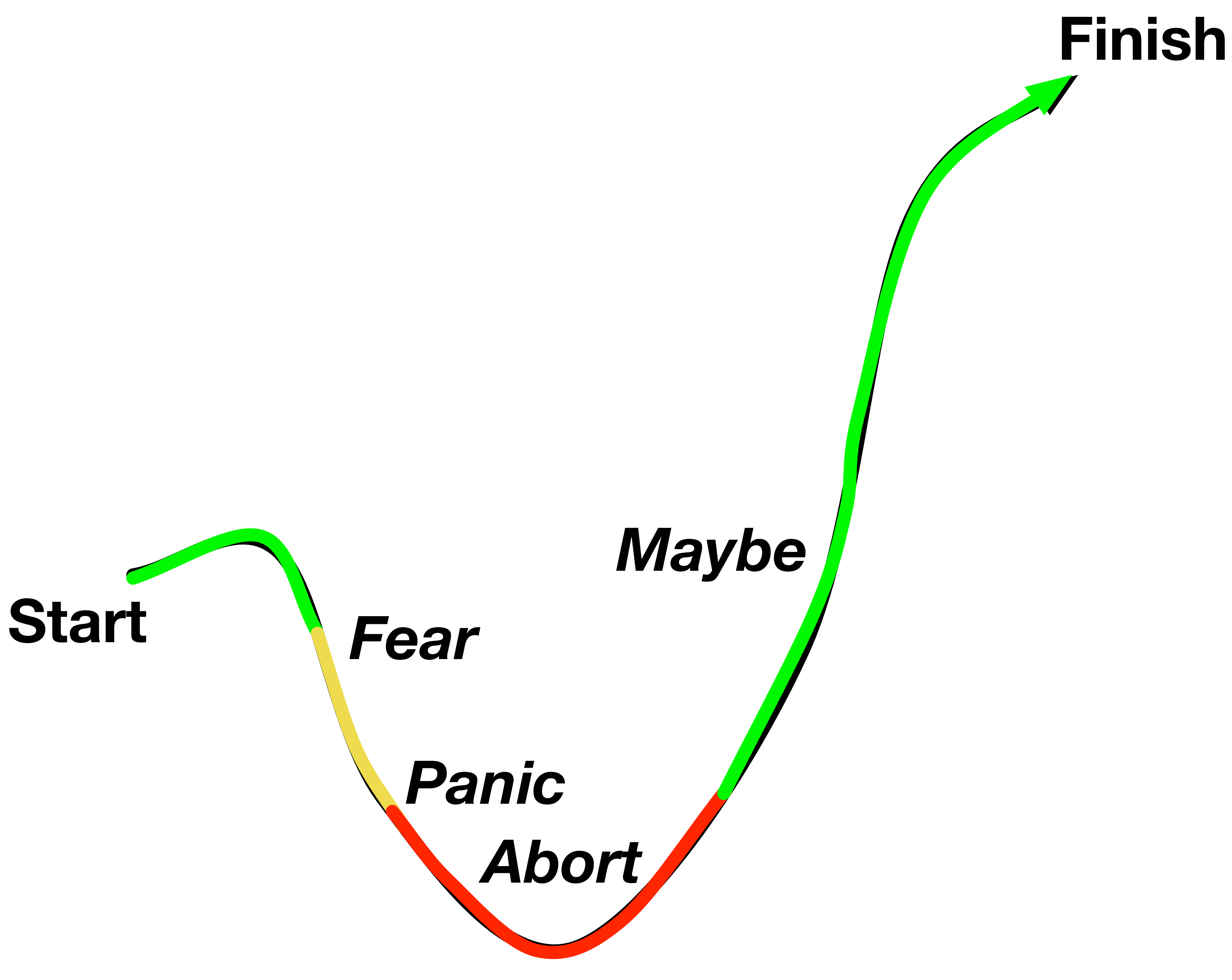
More discipline… tighter process and more approvals

Change Management

Need better tools… cool automation and a new network!!

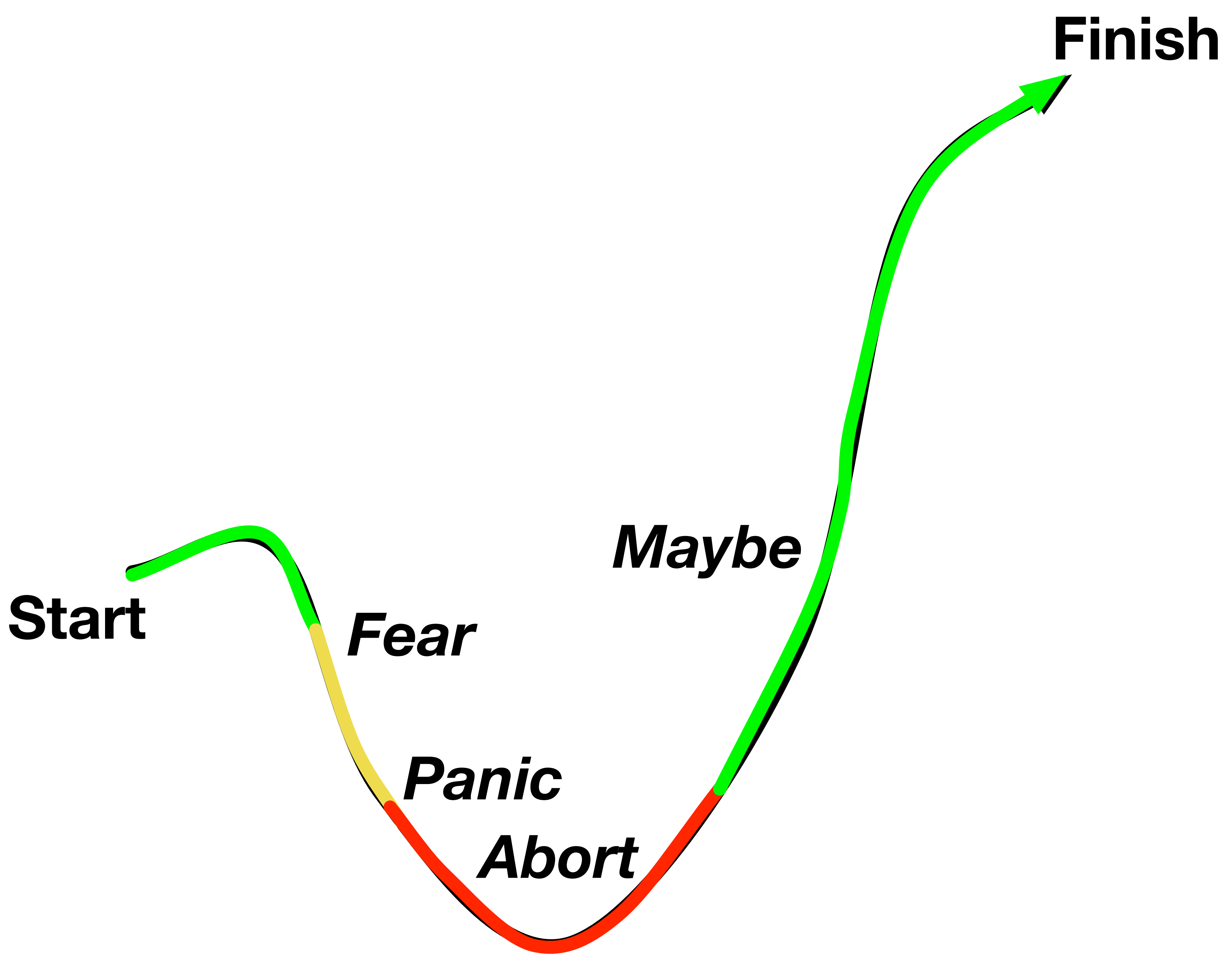
Engineers

“Little J’s” instead of “Big J”

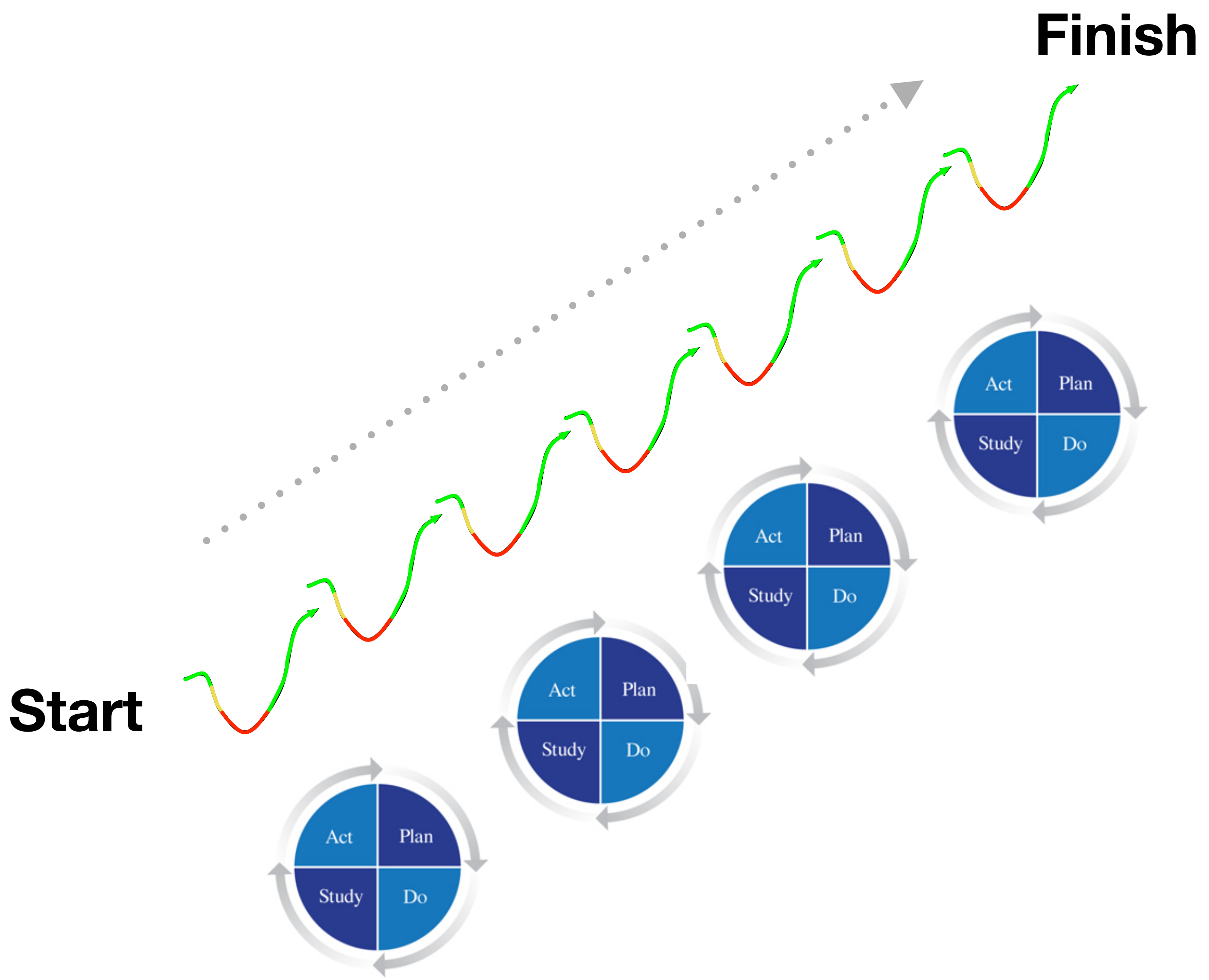


“Big Bang”

“Little J’s” instead of “Big J”

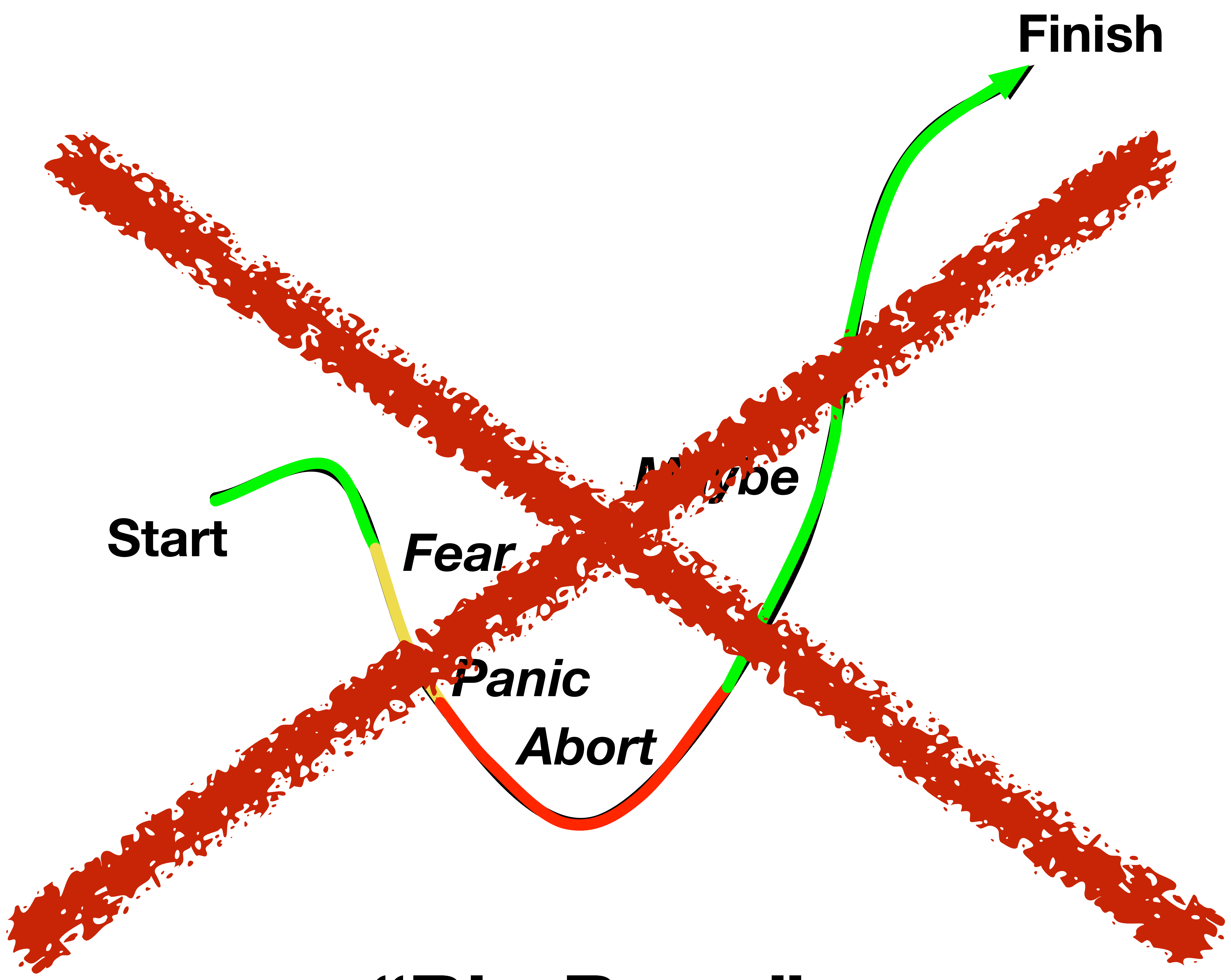


“Big Bang”

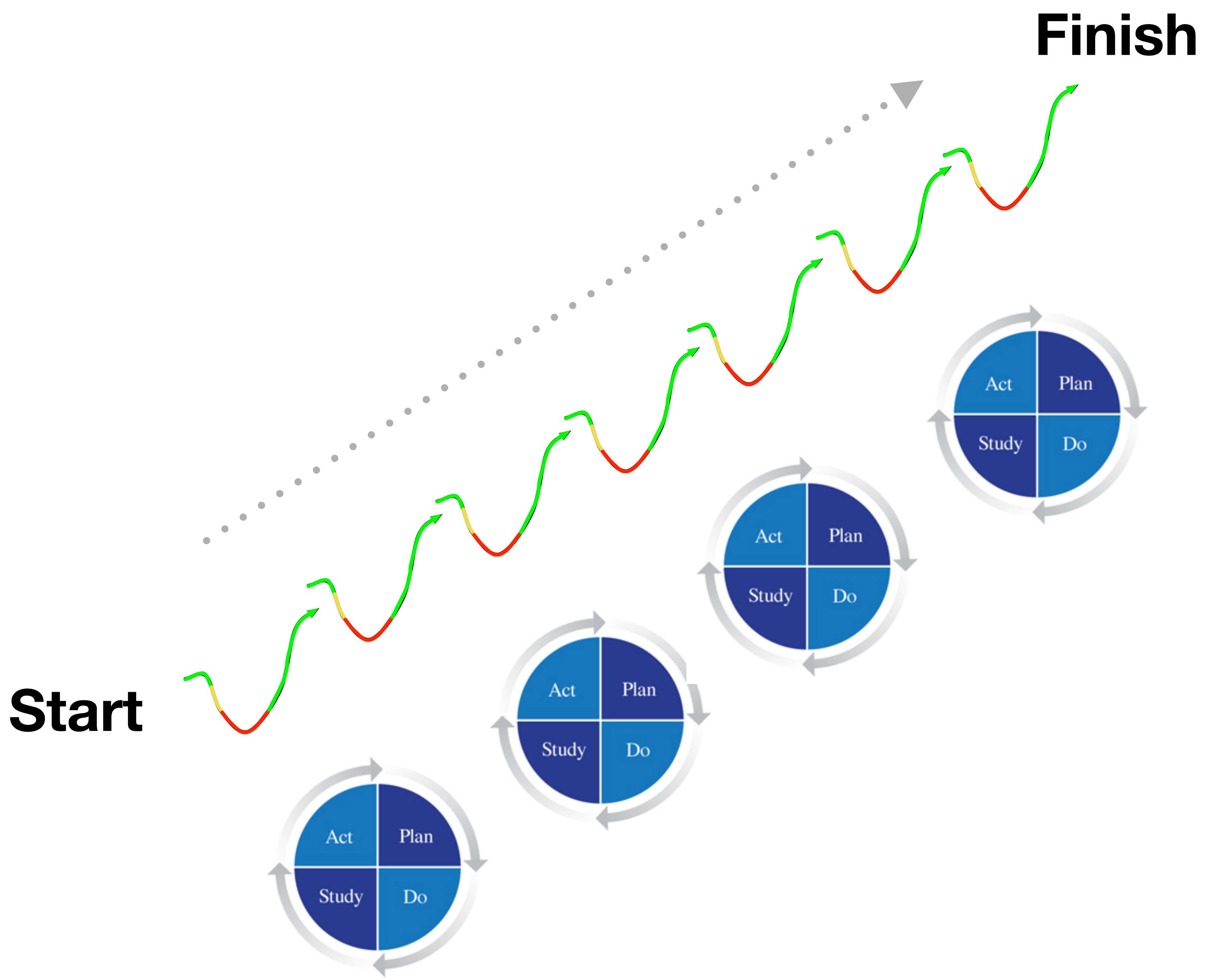


Continuous Improvement

“Little J’s” instead of “Big J”



“Big Bang”



Continuous Improvement

Turn Continuous Improvement into an enterprise program

You are going to have to…

Turn Continuous Improvement into an enterprise program

You are going to have to…

- **Keep improvement efforts aligned**

Turn Continuous Improvement into an enterprise program

You are going to have to…

- **Keep improvement efforts aligned**
- **Scale quickly**

Turn Continuous Improvement into an enterprise program

You are going to have to…

- **Keep improvement efforts aligned**
- **Scale quickly**
- **Span multiple organizational boundaries**

RUNDECK

Turn Continuous Improvement into an enterprise program

You are going to have to…

- **Keep improvement efforts aligned**
- **Scale quickly**
- **Span multiple organizational boundaries**
- **Work with substantial numbers of legacy technologies**

Turn Continuous Improvement into an enterprise program

You are going to have to…

- **Keep improvement efforts aligned**
- **Scale quickly**
- **Span multiple organizational boundaries**
- **Work with substantial numbers of legacy technologies**
- **Develop your existing staff in mass**

Turn Continuous Improvement into an enterprise program

You are going to have to…

- Keep improvement efforts aligned
- Scale quickly
- Span multiple organizational boundaries
- Work with substantial numbers of legacy technologies
- Develop your existing staff in mass
- Be self-funding after initial seed investment

But how do you do that when…

1. The work isn’t visible

2. People are working out of context

3. Inertia is pulling your org out of alignment

You need a systemic way to teach an organization to **find and fix what is getting in its own way.**

“Kaizen”

“Kaizen”

- **Kaizen: Japanese word for improvement**

“Kaizen”

- Kaizen: Japanese word for improvement
- Modern business context:
    - Continuous improvement
    - Systematic, scientific-method approach
    - Total engagement of the workforce
    - Valuing small changes as much as large changes (outcome is what matters)

“Kaizen”

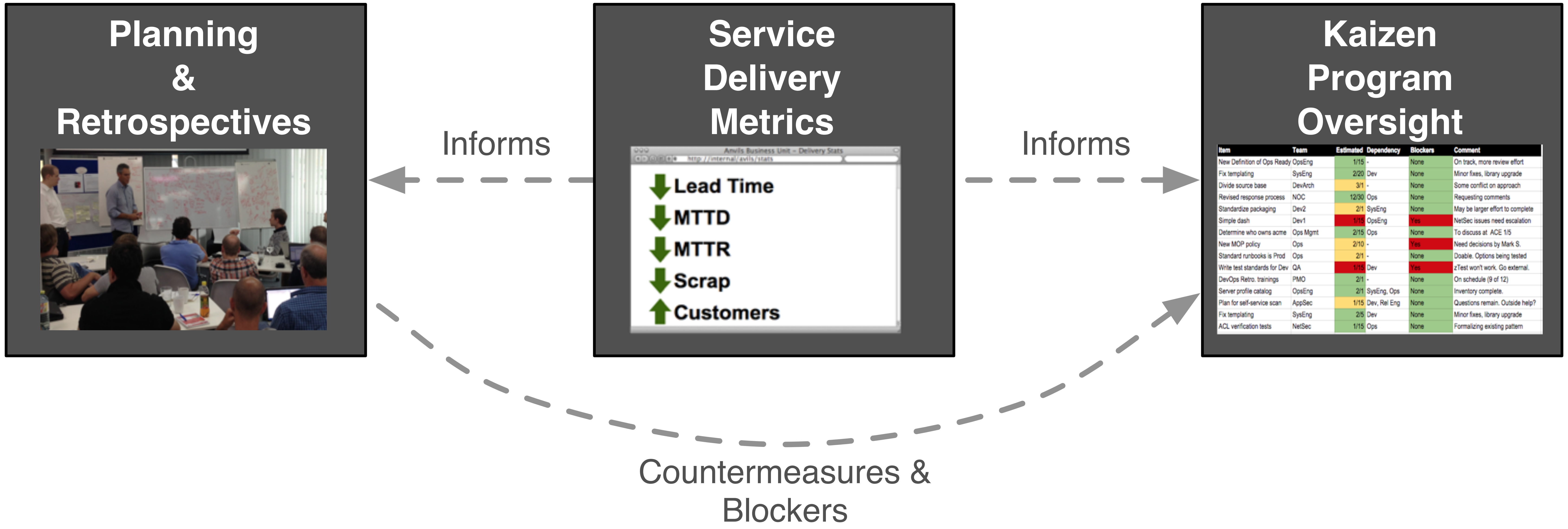
- **Kaizen: Japanese word for improvement**
- **Modern business context:**
    - **Continuous improvement**
    - **Systematic, scientific-method approach**
    - **Total engagement of the workforce**
    - **Valuing small changes as much as large changes (outcome is what matters)**
- **Kaizen in a DevOps context:**
    - **Continuously improve the flow of work through the full value stream in order to improve customer outcomes**

“DevOps Kaizen”

Proven Lean Techniques
+
DevOps Context

> *“If I have seen further, it is by standing on the shoulders of giants.”*
> *-Sir Isaac Newton*

Elements of a DevOps Kaizen Program



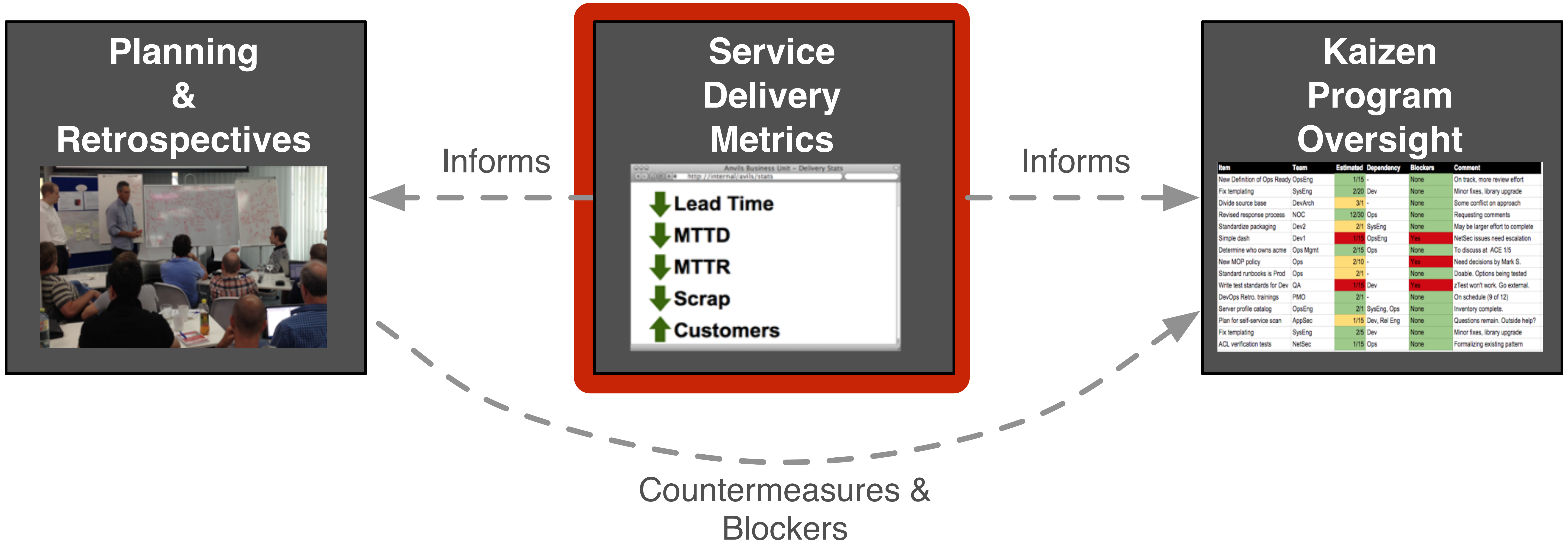
Anvils Business Unit – Delivery Stats

http://internal/avils/stats

- ⬇ Lead Time
- ⬇ MTTD
- ⬇ MTTR
- ⬇ Scrap
- ⬆ Customers

Item	Team	Estimated	Dependency	Blockers	Comment
New Definition of Ops Ready	OpsEng	1/15	-	None	On track, more review effort
Fix templating	SysEng	2/20	Dev	None	Minor fixes, library upgrade
Divide source base	DevArch	3/1	-	None	Some conflict on approach
Revised response process	NOC	12/30	Ops	None	Requesting comments
Standardize packaging	Dev2	2/1	SysEng	None	May be larger effort to complete
Simple dash	Dev1	1/15	OpsEng	Yes	NetSec issues need escalation
Determine who owns acme	Ops Mgmt	2/15	Ops	None	To discuss at ACE 1/5
New MOP policy	Ops	2/10	-	Yes	Need decisions by Mark S.
Standard runbooks is Prod	Ops	2/1	-	None	Doable. Options being tested
Write test standards for Dev	QA	1/15	Dev	Yes	zTest won't work. Go external.
DevOps Retro. trainings	PMO	2/1	-	None	On schedule (9 of 12)
Server profile catalog	OpsEng	2/1	SysEng, Ops	None	Inventory complete.
Plan for self-service scan	AppSec	1/15	Dev, Rel Eng	None	Questions remain. Outside help?
Fix templating	SysEng	2/5	Dev	None	Minor fixes, library upgrade
ACL verification tests	NetSec	1/15	Ops	None	Formalizing existing pattern

Elements of a DevOps Kaizen Program



Anvils Business Unit – Delivery Stats

http://internal/anvils/stats

⬇ Lead Time

⬇ MTTD

⬇ MTTR

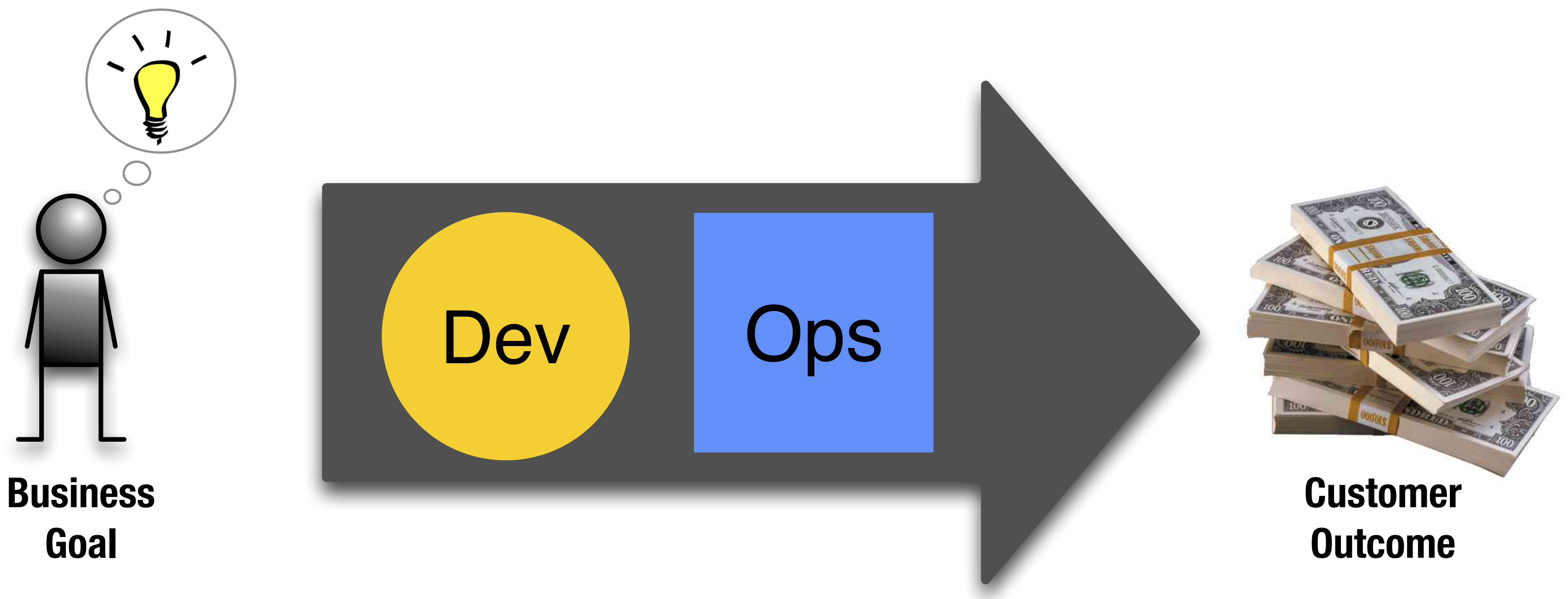
⬇ Scrap

⬆ Customers

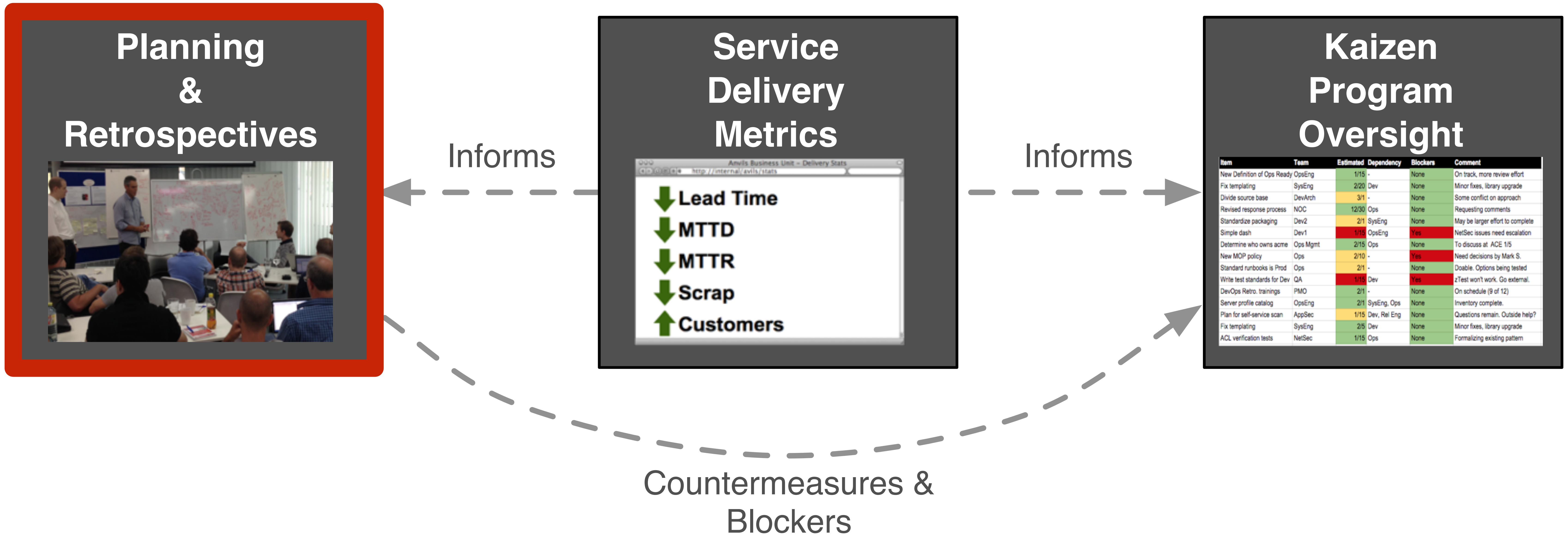
Item	Team	Estimated	Dependency	Blockers	Comment
New Definition of Ops Ready	OpsEng	1/15	-	None	On track, more review effort
Fix templating	SysEng	2/20	Dev	None	Minor fixes, library upgrade
Divide source base	DevArch	3/1	-	None	Some conflict on approach
Revised response process	NOC	12/30	Ops	None	Requesting comments
Standardize packaging	Dev2	2/1	SysEng	None	May be larger effort to complete
Simple dash	Dev1	1/15	OpsEng	Yes	NetSec issues need escalation
Determine who owns acme	Ops Mgmt	2/15	Ops	None	To discuss at ACE 1/5
New MOP policy	Ops	2/10	-	Yes	Need decisions by Mark S.
Standard runbooks is Prod	Ops	2/1	-	None	Doable. Options being tested
Write test standards for Dev	QA	1/15	Dev	Yes	zTest won't work. Go external.
DevOps Retro. trainings	PMO	2/1	-	None	On schedule (9 of 12)
Server profile catalog	OpsEng	2/1	SysEng, Ops	None	Inventory complete.
Plan for self-service scan	AppSec	1/15	Dev, Rel Eng	None	Questions remain. Outside help?
Fix templating	SysEng	2/5	Dev	None	Minor fixes, library upgrade
ACL verification tests	NetSec	1/15	Ops	None	Formalizing existing pattern

Organization-wide focus on service delivery metrics

- **Lead Time** (Duration and Predictability)
- **MTTD** (Mean Time To Detect)
- **MTTR** (Mean Time to Repair, Mean Time to Fix)
- **Quality at the Source** (Scrap/Rework)



Elements of a DevOps Kaizen Program



Planning & Retrospectives

Service Delivery Metrics

Anvils Business Unit – Delivery Stats

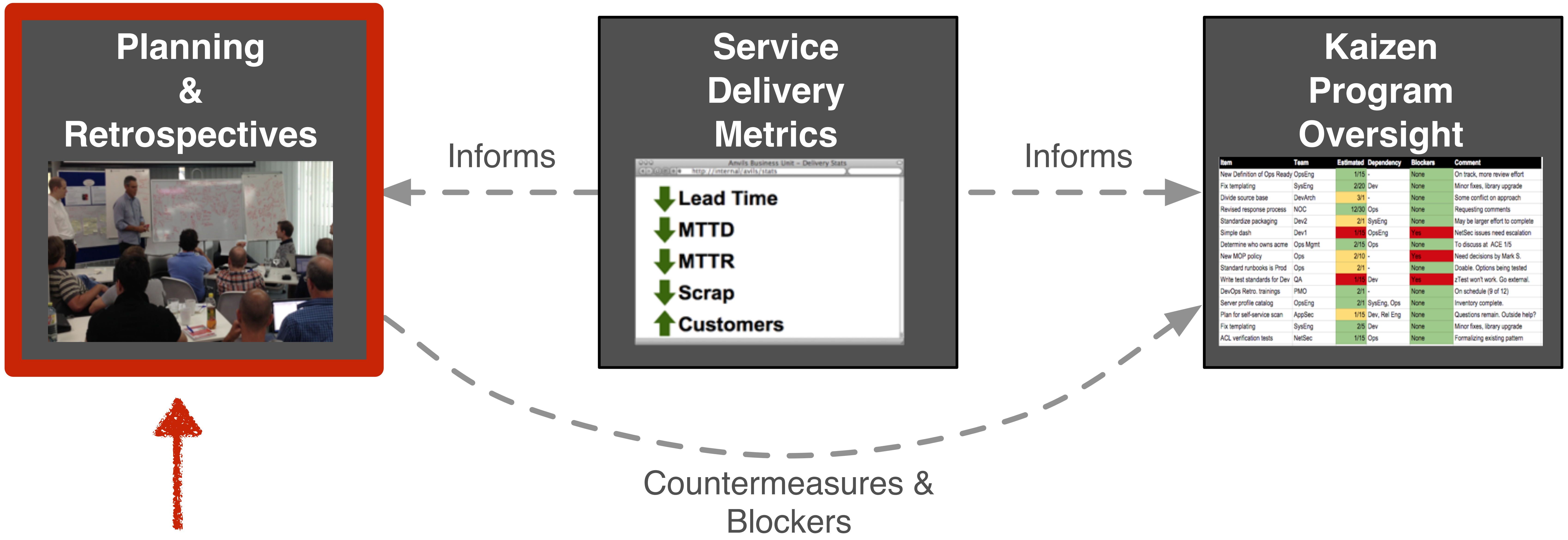
http://internal/anvils/stats

- ↓ Lead Time
- ↓ MTTD
- ↓ MTTR
- ↓ Scrap
- ↑ Customers

Kaizen Program Oversight

Item	Team	Estimated	Dependency	Blockers	Comment
New Definition of Ops Ready	OpsEng	1/15	-	None	On track, more review effort
Fix templating	SysEng	2/20	Dev	None	Minor fixes, library upgrade
Divide source base	DevArch	3/1	-	None	Some conflict on approach
Revised response process	NOC	12/30	Ops	None	Requesting comments
Standardize packaging	Dev2	2/1	SysEng	None	May be larger effort to complete
Simple dash	Dev1	1/15	OpsEng	Yes	NetSec issues need escalation
Determine who owns acme	Ops Mgmt	2/15	Ops	None	To discuss at ACE 1/5
New MOP policy	Ops	2/10	-	Yes	Need decisions by Mark S.
Standard runbooks is Prod	Ops	2/1	-	None	Doable. Options being tested
Write test standards for Dev	QA	1/15	Dev	Yes	zTest won't work. Go external.
DevOps Retro. trainings	PMO	2/1	-	None	On schedule (9 of 12)
Server profile catalog	OpsEng	2/1	SysEng, Ops	None	Inventory complete.
Plan for self-service scan	AppSec	1/15	Dev, Rel Eng	None	Questions remain. Outside help?
Fix templating	SysEng	2/5	Dev	None	Minor fixes, library upgrade
ACL verification tests	NetSec	1/15	Ops	None	Formalizing existing pattern

Elements of a DevOps Kaizen Program



Planning & Retrospectives

Informs

Service Delivery Metrics

Anvils Business Unit – Delivery Stats

http://internal/avils/stats

- ⬇ Lead Time
- ⬇ MTTD
- ⬇ MTTR
- ⬇ Scrap
- ⬆ Customers

Informs

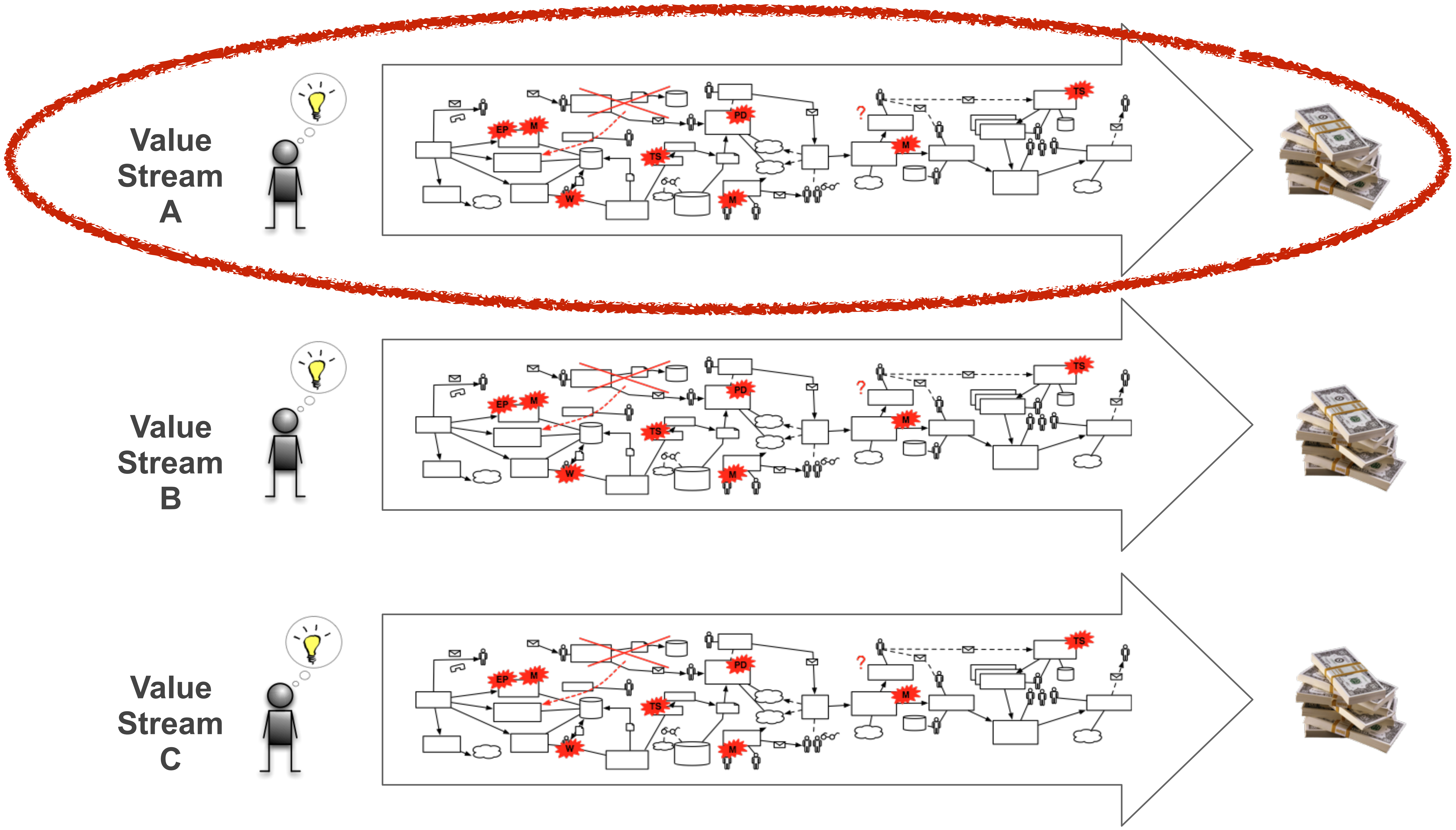
Kaizen Program Oversight

Item	Team	Estimated	Dependency	Blockers	Comment
New Definition of Ops Ready	OpsEng	1/15	-	None	On track, more review effort
Fix templating	SysEng	2/20	Dev	None	Minor fixes, library upgrade
Divide source base	DevArch	3/1	-	None	Some conflict on approach
Revised response process	NOC	12/30	Ops	None	Requesting comments
Standardize packaging	Dev2	2/1	SysEng	None	May be larger effort to complete
Simple dash	Dev1	1/15	OpsEng	Yes	NetSec issues need escalation
Determine who owns acme	Ops Mgmt	2/15	Ops	None	To discuss at ACE 1/5
New MOP policy	Ops	2/10	-	Yes	Need decisions by Mark S.
Standard runbooks is Prod	Ops	2/1	-	None	Doable. Options being tested
Write test standards for Dev	QA	1/15	Dev	Yes	zTest won't work. Go external.
DevOps Retro. trainings	PMO	2/1	-	None	On schedule (9 of 12)
Server profile catalog	OpsEng	2/1	SysEng, Ops	None	Inventory complete.
Plan for self-service scan	AppSec	1/15	Dev, Rel Eng	None	Questions remain. Outside help?
Fix templating	SysEng	2/5	Dev	None	Minor fixes, library upgrade
ACL verification tests	NetSec	1/15	Ops	None	Formalizing existing pattern

Countermeasures & Blockers

This is where the work becomes visible!

Retrospectives are a per value stream tool



Value Stream A

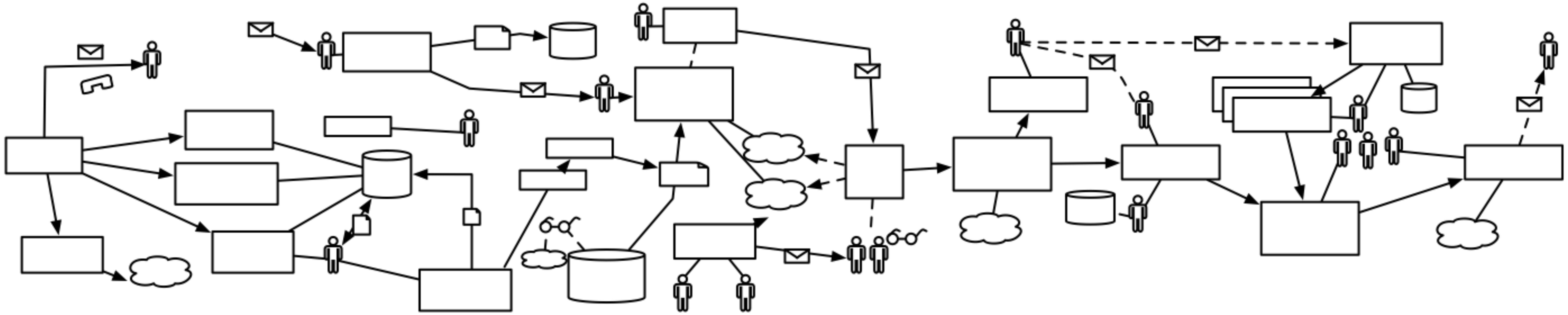
Value Stream B

Value Stream C

Key:
"horizontal
thinking"

DevOps Kaizen: Retrospective Technique

1 **Map end-to-end process**

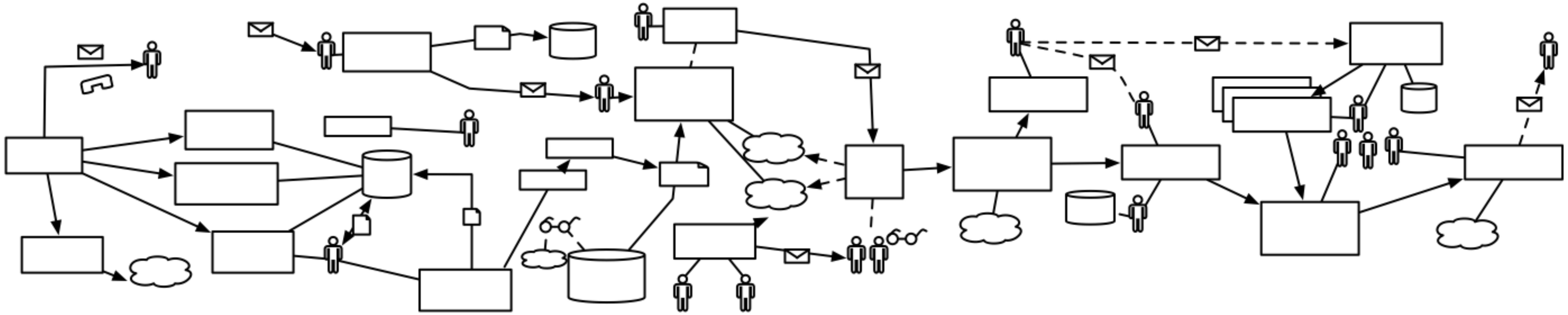


Include key process metrics:

Lead Time
Processing Time
Scrap Rate
Head Count

DevOps Kaizen: Retrospective Technique

1 **Map end-to-end process**



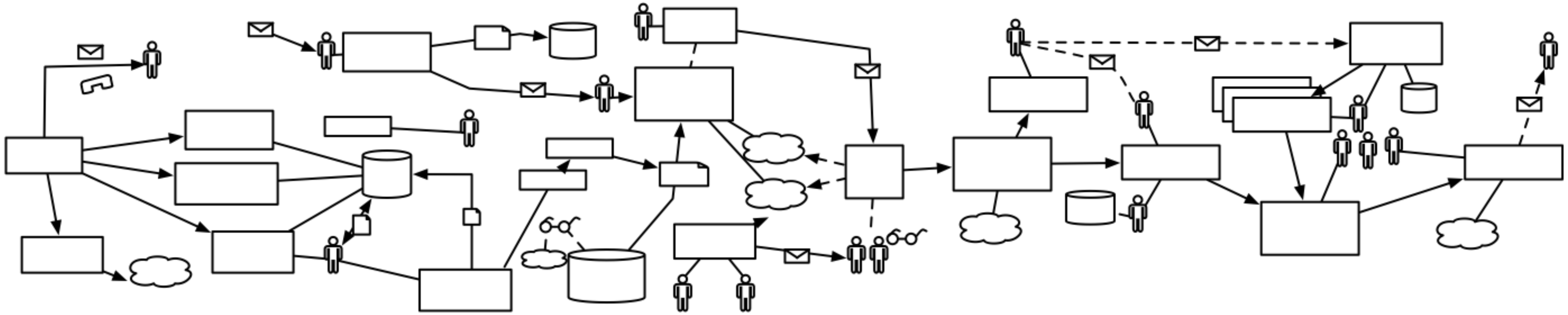
Note: "going to the gemba" requires making it visible together

Include key process metrics:

Lead Time
Processing Time
Scrap Rate
Head Count

DevOps Kaizen: Retrospective Technique

1 **Map end-to-end process**



Note: "going to the gemba" requires making it visible together

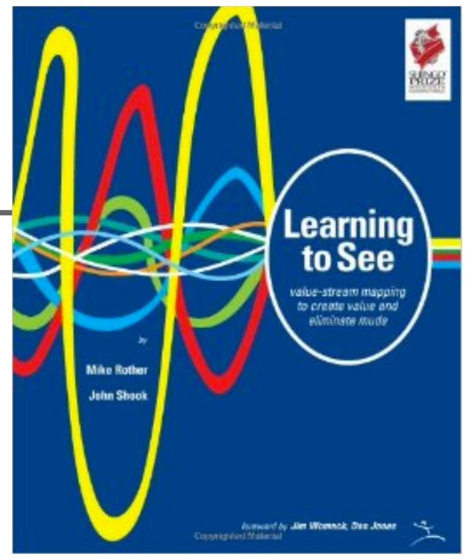
Include key process metrics:

Lead Time
Processing Time
Scrap Rate
Head Count

Key: graphical facilitation above all else!

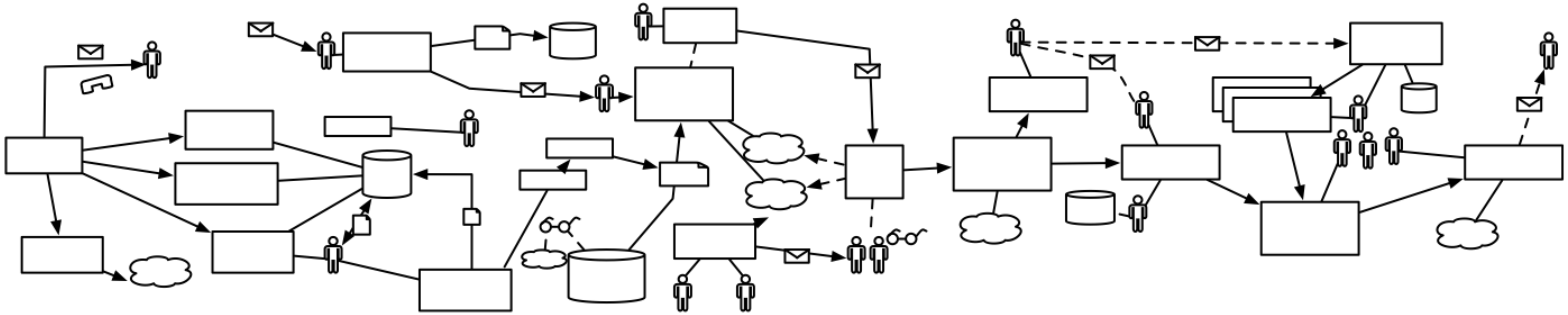
RUNDECK

DevOps Kaizen: Retrospective Technique



Inspiration: value stream mapping

1 **Map end-to-end process**



Note: "going to the gemba" requires making it visible together

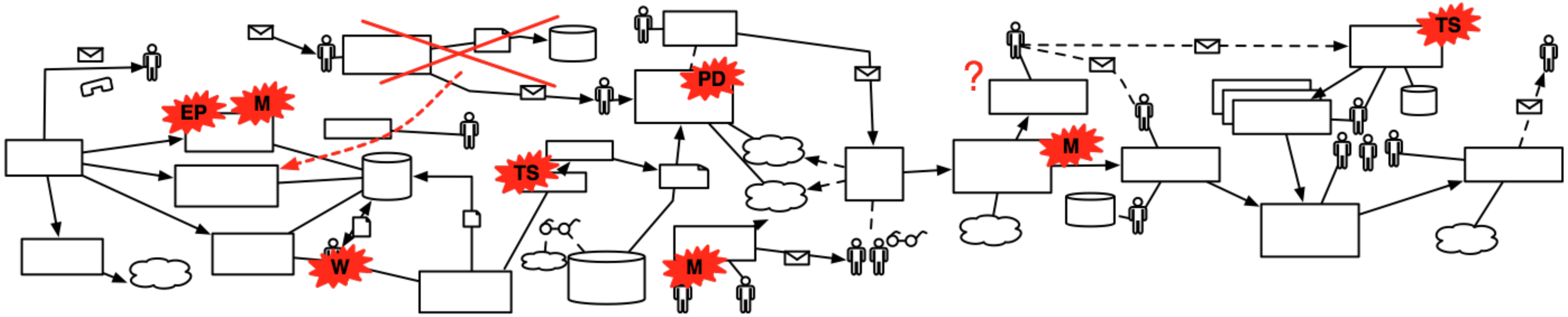
Include key process metrics:

Lead Time
Processing Time
Scrap Rate
Head Count

Key: graphical facilitation above all else!

DevOps Kaizen: Retrospective Technique

2 **Identify wastes, inefficiencies, bottlenecks**

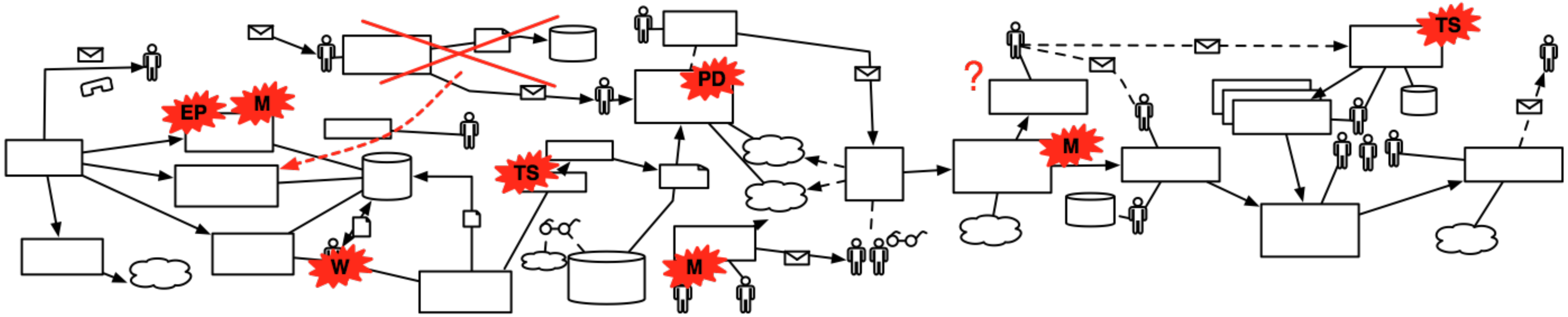


Structured approach building on DevOps adaptation of “7 deadly wastes” from Lean / Agile:

PD - Partially Done	***D - Defects***
TS - Task Switching	***EP - Extra Process***
W - Waiting	***EF - Extra Features***
M - Motion / Manual	***HB - Heroics***

DevOps Kaizen: Retrospective Technique

2 **Identify wastes, inefficiencies, bottlenecks**



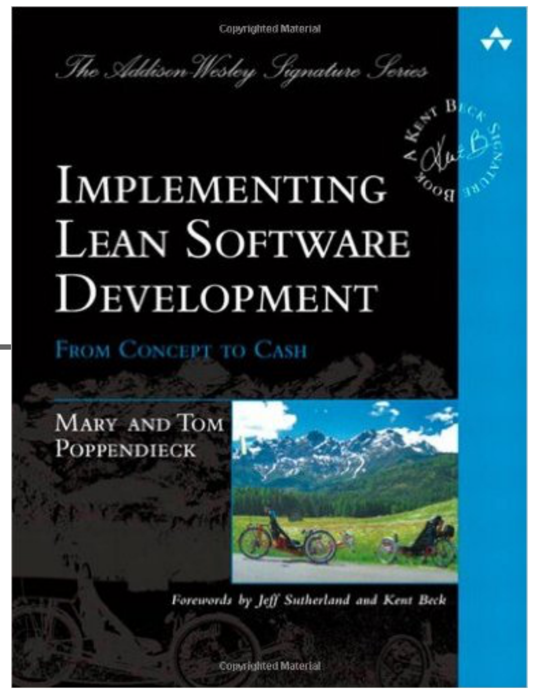
Structured approach building on DevOps adaptation of “7 deadly wastes” from Lean / Agile:

PD - Partially Done	*D - Defects*
TS - Task Switching	*EP - Extra Process*
W - Waiting	*EF - Extra Features*
M - Motion / Manual	*HB - Heroics*

Key: focus on flow of value... not gripes

RUNDECK

DevOps Kaizen: Retrospective Technique



The Addison-Wesley Signature Series

IMPLEMENTING LEAN SOFTWARE DEVELOPMENT

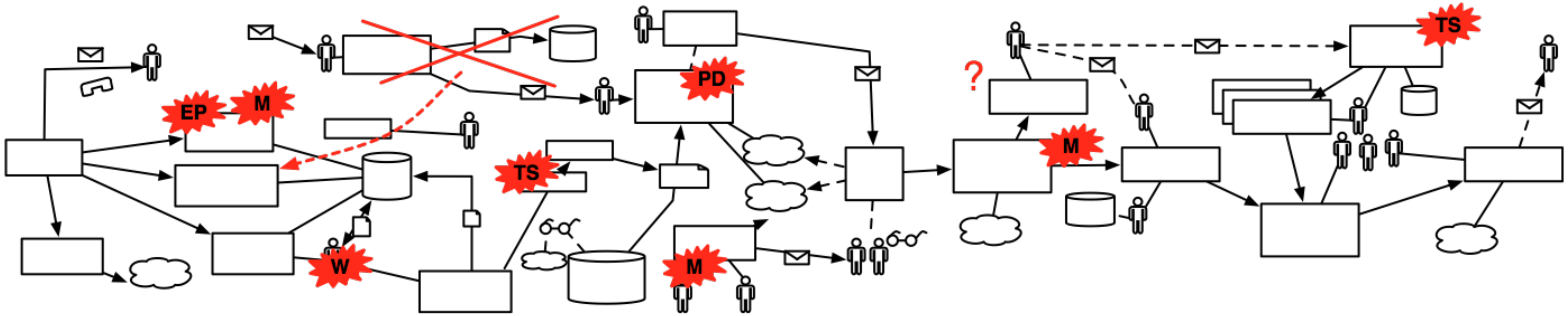
FROM CONCEPT TO CASH

MARY AND TOM POPPENDIECK

Forewords by Jeff Sutherland and Kent Beck

Inspiration: 7 Wastes of Software Development

2 Identify wastes, inefficiencies, bottlenecks



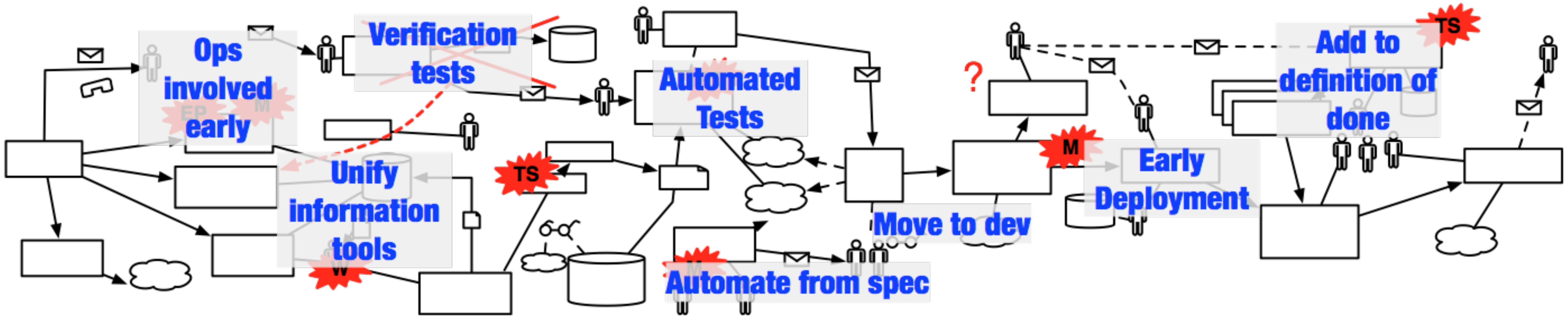
Structured approach building on DevOps adaptation of “7 deadly wastes” from Lean / Agile:

PD - Partially Done	*D - Defects*
TS - Task Switching	*EP - Extra Process*
W - Waiting	*EF - Extra Features*
M - Motion / Manual	*HB - Heroics*

Key: focus on flow of value... not gripes

DevOps Kaizen: Retrospective Technique

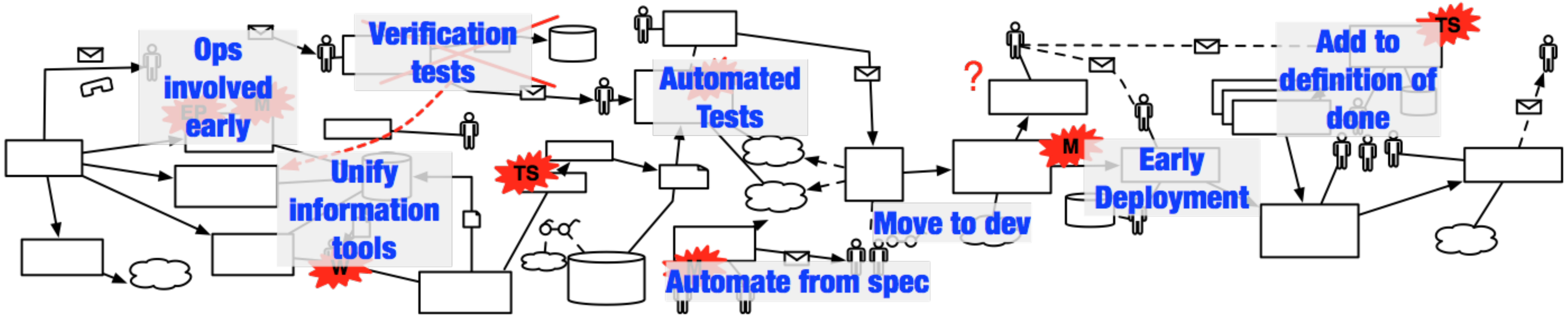
3 **Identify countermeasures**



Countermeasures must be actionable, backlog ready. Focus on short-term "baby steps". Note broader, strategic recommendations.

DevOps Kaizen: Retrospective Technique

3 Identify countermeasures

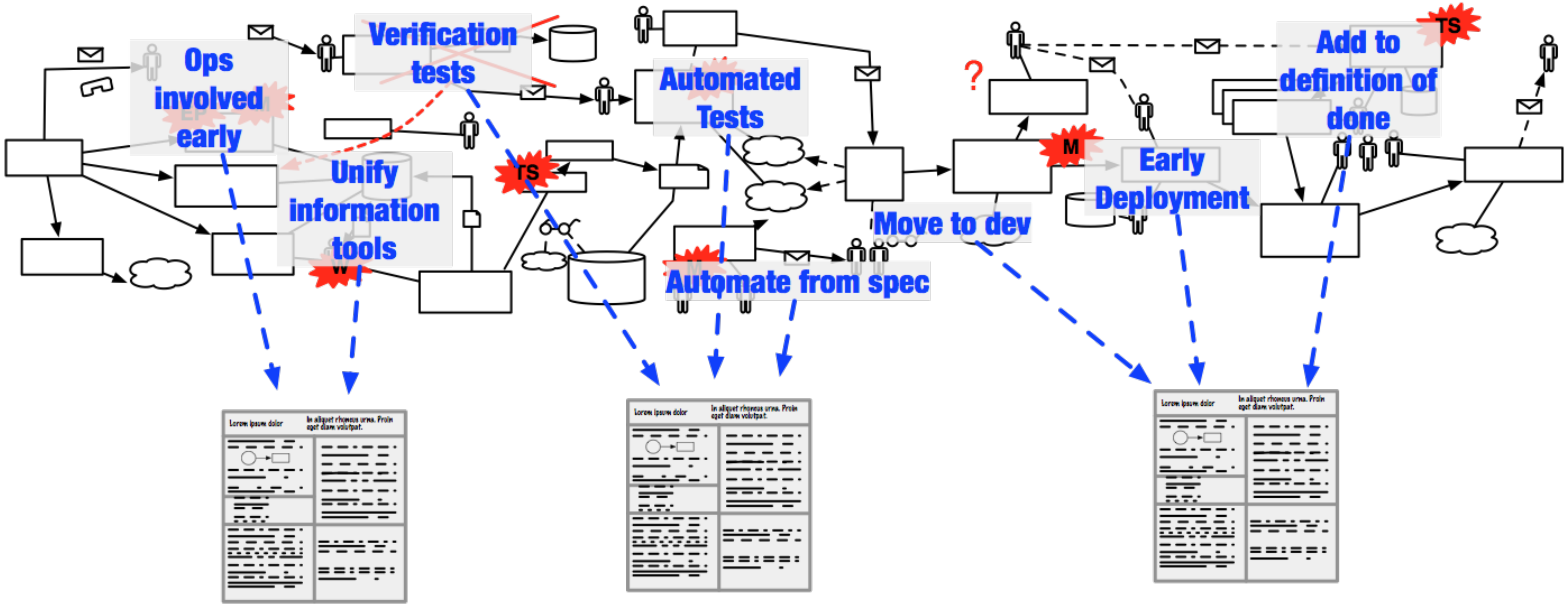


Key: "small j's, not big j's"

Countermeasures must be actionable, backlog ready. Focus on short-term "baby steps". Note broader, strategic recommendations.

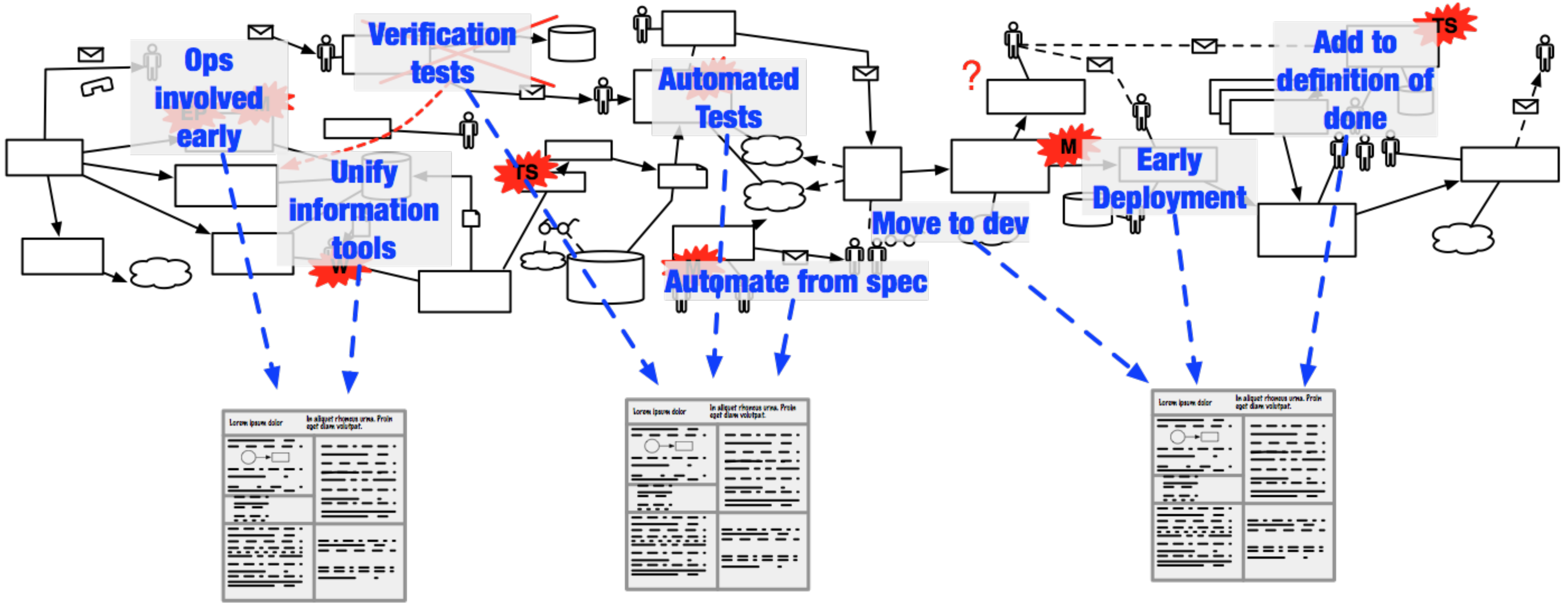
DevOps Kaizen: Retrospective Technique

4 Create Improvement Storyboards (Kata Style)



DevOps Kaizen: Retrospective Technique

4 Create Improvement Storyboards (Kata Style)



Key: actionable short-term "baby steps"...
"what are we going to do next?"

DevOps Kaizen: Retrospective Technique

Major Feature for Partner Markets

Customers and Clients

Upcoming Market Event

Chief Strategy Officer

1.

Strategize Product Roadmap

EP

D

TS

Ad-Hoc

Weekly Exec Meeting

"John"

P.M

6. Service Aligned Team Concept?

Product Requirements Defined

S/R - 71%

ARB Queue

"Susan"

Program

Track Ticket Dependencies

Jira "Stories"

Confluence Pages

design docs

Chief Architect "Linda"

2.

Architecture Review Board

Ops (sometimes)

?

Working Group

EP

M

Assign Requirements

more detail for their team

Team Leads and PMs

4. Write end-to-end customer func. tests

Development

Development Sprint 2 week c/t

Existing Dev Environments

Devs, PM, Engr, QA

Service Verification test writing: shift left to Dev (test early)

5. Resolve interface to legacy

3.

D

Test Data Configuration Manager "Jennifer"

Data Setup

TS

PD

M

W

Acquire / Prepare needed data

DBA in TechOps

S/R - 90%

1. Test data setup automation

Scrum Dev/QA

Dev, QA

S/R - 55%

4.

Deploy to Integration

M

Integration & Regression Testing focused on service

5.

Scrum Dev/QA

Test Link

TS

D

Integ03

Sprint Review

Remove Bottleneck and Environment Contention (test more)

Product Owners (Using own criteria)

6.

Release to Prod

TS

Jira

or

Scrum Team (if new arch)

Ops Team (if legacy)

D

Stage

Push Deployment to Stage

S/R - 32%

3. Dev to Prod for legacy

2. Unify change management tools

Create CAB ticket

Legacy

NewArch

Jira

Service Now

Build VMs

Ops

Email Notification

QA Lead PMs QAs

End to end testing in Prod

Team Leads

Go-No Go decision meeting

7.

S/R - 43%

PD

M

Prod Env

Prd DB

Jira

Ops

8.

7. Tool

"Remove Feature Flag"

By Cluster

M

Data Setup 6 weeks H/C: 6

Integration Testing 3 weeks H/C: 8

Requirements / Planning 17 weeks

Dev / Test 16 weeks

Staging Release 4 weeks H/C:8

Prod Release 3 weeks H/C: 14

Business Trigger

40 weeks total

Major Feature for Partner Markets

Customers and Clients
Upcoming Market Event
1.
Chief Strategy Officer
Strategize Product Roadmap
Weekly Exec Meeting
6. Service Aligned Team Concept?
EP
Ad-Hoc
D
TS
"John"
P.M
Product Requirements Defined
S/R - 71%
ARB Queue
"Susan"
Program
Track Ticket Dependencies
Jira "Stories"
Confluence Pages
design docs
2.
Chief Architect "Linda"
Ops (sometimes)
?
Architecture Review Board
Working Group
EP
M
Assign Requirements
more detail for their team
Team Leads and PMs
4. Write end-to-end customer func. tests
Development
Development Sprint 2 week c/t
Existing Dev Environments
Devs, PM, Engr, QA
3.
D
Test Data Configuration Manager "Jennifer"
Data Setup
TS
PD
M
W
Acquire / Prepare needed data
S/R - 90%
DBA in TechOps
5. Resolve interface to legacy
Service Verification test writing: shi
(test early)
1. Test data setup automation
Scrum Dev/QA
D
D
In
fe
Remove Bottleneck and Environme
(test more)
Data Setup
6 weeks H/C: 6
Inte
3 v
Requirements / Planning
17 weeks
Dev / Test
16 weeks
Business Trigger

Major Feature for Partner Markets

Customers and Clients

Upcoming Market Event

Chief Strategy Officer

1\.

Strategize Product Roadmap

EP

D

TS

Ad-Hoc

Weekly Exec Meeting

"John"

P.M

6. Service Aligned Team Concept?

Product Requirements Defined

S/R - 71%

"Susan"

Program

Track Ticket Dependencies

ARB Queue

Jira "Stories"

Confluence Pages

design docs

Chief Architect "Linda"

2\.

Architecture Review Board

Ops (sometimes)

?

Working Group

EP

M

Assign Requirements

more detail for their team

Team Leads and PMs

4. Write end-to-end customer func. tests

Development

Development Sprint 2 week c/t

Existing Dev Environments

Devs, PM, Engr, QA

DBA in TechOps

Acquire / Prepare needed data

3\.

D

Test Data Configuration Manager "Jennifer"

Data Setup

TS

PD

M

W

S/R - 90%

Service Verification test writing: shift left to Dev (test early)

5. Resolve interface to legacy

1. Test data setup automation

Remove Bottleneck and Environment Contention (test more)

S/R - 55%

Dev, QA

4\.

Deploy to Integration

M

Scrum Dev/QA

Integration & Regression Testing focused on service

5\.

TS

D

Integ03

Scrum Dev/QA

Test Link

Sprint Review

6\.

Product Owners (Using own criteria)

Release to Prod

TS

Jira

or

Scrum Team (if new arch)

Ops Team (if legacy)

D

Stage

Push Deployment to Stage

S/R - 32%

3. Dev to Prod for legacy

2. Unify change management tools

Create CAB ticket

Legacy

NewArch

Jira

Service Now

Build VMs

Ops

Email Notification

QA Lead PMs QAs

End to end testing in Prod

Team Leads

Go-No Go decision meeting

7\.

S/R - 43%

PD

M

Prod Env

Prd DB

Jira

Ops

8\.

"Remove Feature Flag" By Cluster

M

7. Tool

Data Setup

6 weeks H/C: 6

Integration Testing

3 weeks H/C: 8

Requirements / Planning

17 weeks

Dev / Test

16 weeks

Staging Release

4 weeks H/C:8

Prod Release

3 weeks H/C: 14

Business Trigger

40 weeks total

Service Verification test writing: shift left to Dev (test early)

hief
hitect
"Linda"

Ops
(sometimes)

?

e
rd

Working Group

EP

M

gn
ments

etail for their team

Leads
PMs

4. Write end-to-end customer func. tests

Development

Development Sprint
2 week c/t

Existing Dev Environments

Devs, PM, Engr, QA

3.

D

Test Data Configuration Manager
"Jennifer"

Data Setup

TS

PD

M

W

Acquire / Prepare needed data

DBA in TechOps

S/R - 90%

5. Resolve interface to legacy

1. Test data setup automation

Dev, QA

S/R - 55%

4.

Deploy to Integration

M

Scrum Dev/QA

Integration & Regression Testing
focused on service

Scrum Dev/QA

Test Link

5.

TS

D

Integ03

Sprint Review

Product Owners
(Using own criteria)

6.

Release to Prod

TS

Jira

or

Scrum Team
(if new arch)

Ops Team
(if legacy)

D

Stage

Push Deployment to Stage

S/R - 32%

3. Dev to Prod for legacy

C

Remove Bottleneck and Environment Contention (test more)

Data Setup
6 weeks H/C: 6

Integration Testing
3 weeks H/C: 8

Dev / Test
16 weeks

Staging
4 weeks

Major Feature for Partner Markets

Service Verification test writing: shift left to Dev (test early)

Remove Bottleneck and Environment Contention (test more)

6. Service Aligned Team Concept?

4. Write end-to-end customer func. tests

5. Resolve interface to legacy

1. Test data setup automation

3. Dev to Prod for legacy

2. Unify change management tools

7. Tool

Upcoming Market Event
Customers and Clients
Chief Strategy Officer
1.
Strategize Product Roadmap
EP
D
TS
Ad-Hoc
"John"
P.M
Weekly Exec Meeting
Product Requirements Defined
S/R - 71%
"Susan"
Program
Track Ticket Dependencies
ARB Queue
Jira "Stories"
Confluence Pages
design docs
Chief Architect "Linda"
2.
Architecture Review Board
Ops (sometimes)
?
Working Group
EP
M
Assign Requirements
more detail for their team
Team Leads and PMs
Development
Development Sprint 2 week c/t
Existing Dev Environments
Devs, PM, Engr, QA
3.
D
Test Data Configuration Manager "Jennifer"
Data Setup
TS
PD
M
W
Acquire / Prepare needed data
DBA in TechOps
S/R - 90%
Dev, QA
S/R - 55%
4.
Deploy to Integration
M
Scrum Dev/QA
Integration & Regression Testing focused on service
5.
TS
D
Integ03
Scrum Dev/QA
Test Link
Sprint Review
6.
Product Owners (Using own criteria)
Release to Prod
TS
Jira
or
Scrum Team (if new arch)
Ops Team (if legacy)
D
Stage
Push Deployment to Stage
S/R - 32%
Email Notification
Create CAB ticket
Legacy
NewArch
Jira
Service Now
Build VMs
Ops
QA Lead PMs QAs
End to end testing in Prod
Team Leads
Go-No Go decision meeting
7.
S/R - 43%
PD
M
Prod Env
Prd DB
Jira
Ops
8.
"Remove Feature Flag"
By Cluster
M
Data Setup 6 weeks H/C: 6
Integration Testing 3 weeks H/C: 8
Requirements / Planning 17 weeks
Dev / Test 16 weeks
Staging Release 4 weeks H/C:8
Prod Release 3 weeks H/C: 14
Business Trigger
40 weeks total

vice Verification test writing: shift left to Dev
(test early)

5. Resolve interface to legacy

Test Data Configuration Manager "Jennifer"

3.

D

Data Setup

TS

PD

M

W

ire / needed a

S/R - 90%

1. Test data setup automation

Dev, QA

S/R - 55%

4.

Deploy to Integration

M

Scrum Dev/QA

Integration & Regression Testing focused on service

Test Link

5.

Scrum Dev/QA

TS

D

Integ03

Sprint Review

6.

Product Owners
(Using own criteria)

Release to Prod

TS

Jira

or

Scrum Team
(if new arch)

Ops Team
(if legacy)

D

Stage

Push Deployment to Stage

S/R - 32%

Email Notification

2. Unify change management tools

Create CAB ticket

Legacy

NewArch

Jira

Service Now

Build VMs

Ops

3. Dev to Prod for legacy

QA Lead
PMs
QAs

End to end testing in Prod

Team Leads

Go-No Go decision meeting

7.

S/R - 43%

PD

M

Prod Env

Prd DB

Jira

Ops

8.

"Remove Feature Flag"

M

By Cluster

7. Tool

move Bottleneck and Environment Contention
(test more)

Data Setup
weeks H/C: 6

Integration Testing
3 weeks H/C: 8

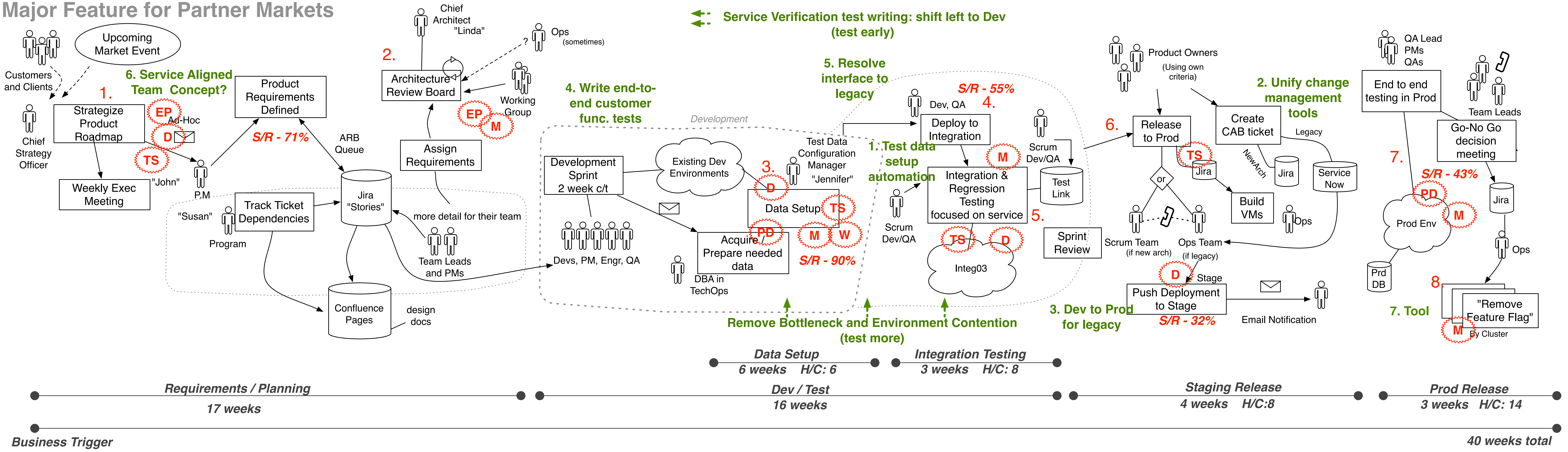
Dev / Test
16 weeks

Staging Release
4 weeks H/C:8

Prod Release
3 weeks H/C: 14

40 weeks total

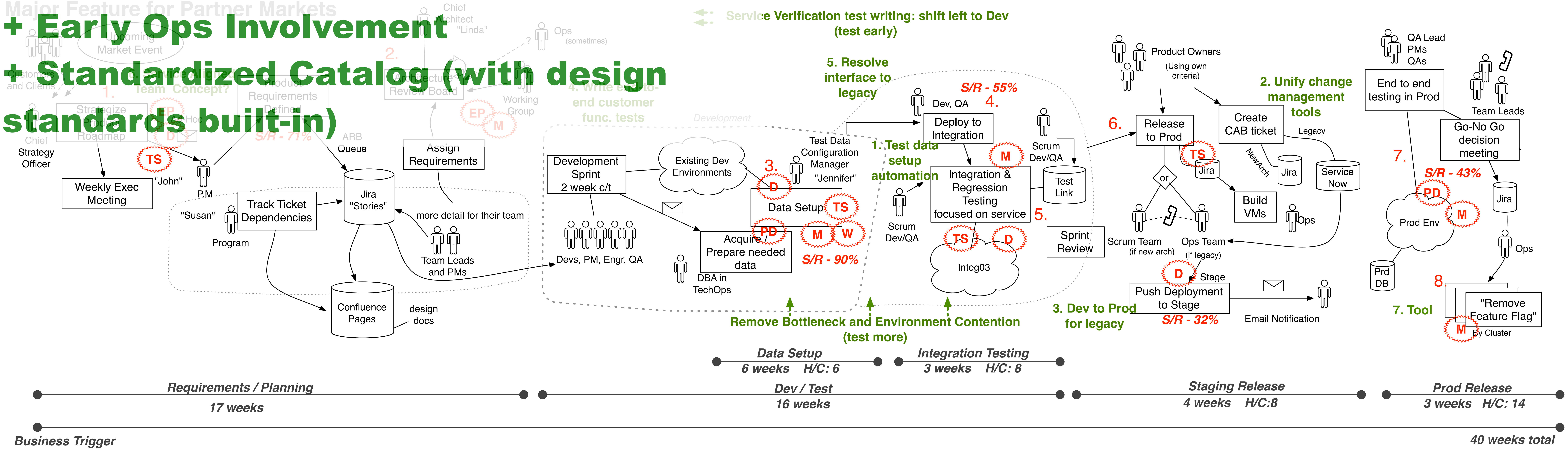
Major Feature for Partner Markets



+ Work in small batches

+ Early Ops Involvement

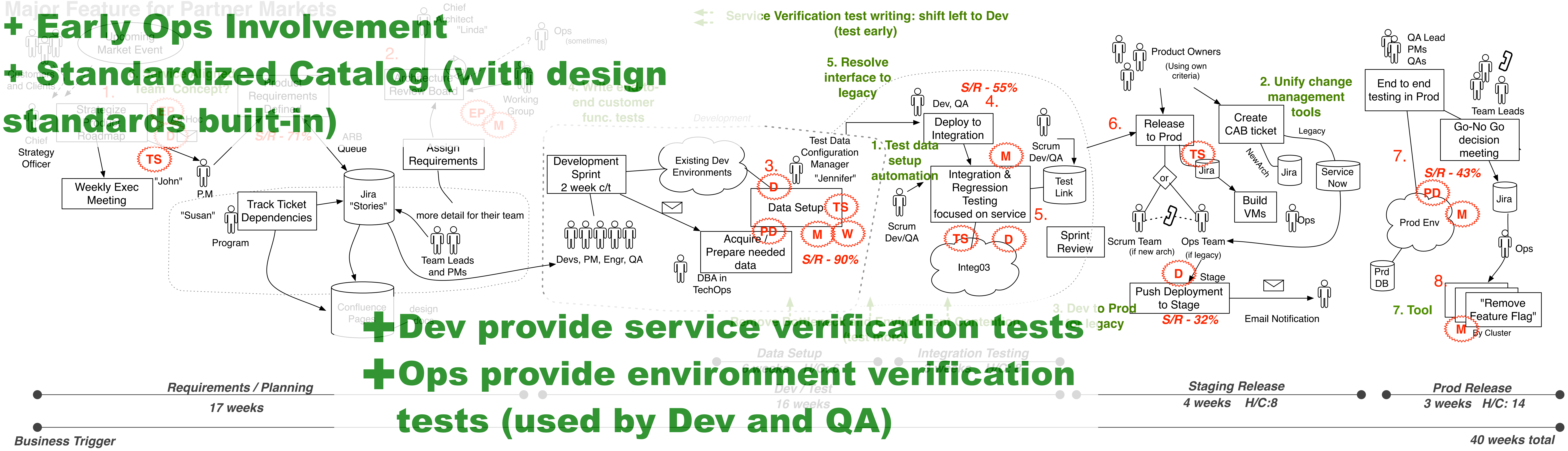
+ Standardized Catalog (with design standards built-in)



+ Work in small batches

+ Early Ops Involvement

+ Standardized Catalog (with design standards built-in)



✚Dev provide service verification tests

✚Ops provide environment verification tests (used by Dev and QA)

✚Service service test data setup (including mainframe)

+ Work in small batches

+ Early Ops Involvement

+ Standardized Catalog (with design standards built-in)

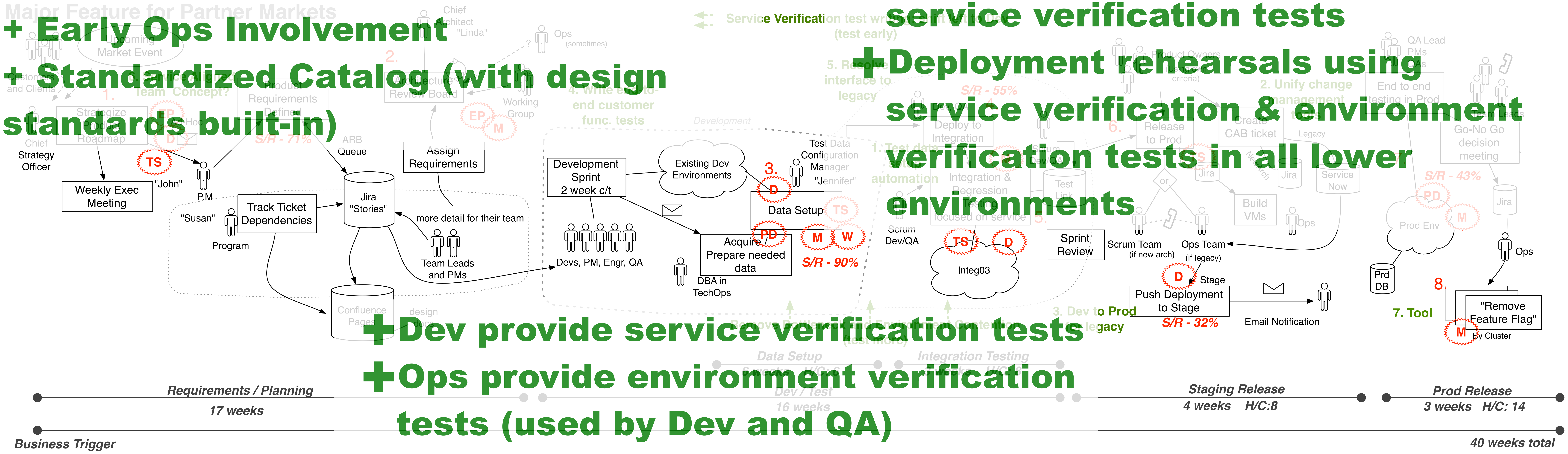
+All deploys use Dev provided service verification tests

+Deployment rehearsals using service verification & environment verification tests in all lower environments

+Dev provide service verification tests

+Ops provide environment verification tests (used by Dev and QA)

+Service service test data setup (including mainframe)



Key: "What can we do next?" NOT "what is nirvana?"

+ Work in small batches

+ Early Ops Involvement

+ Standardized Catalog (with design standards built-in)

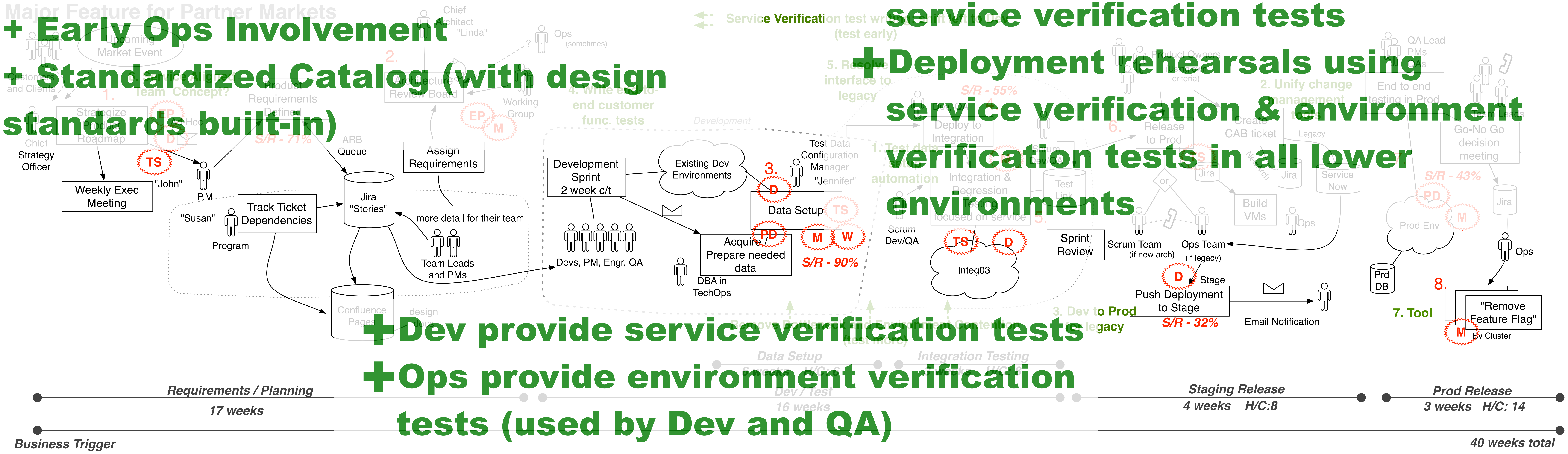
+All deploys use Dev provided service verification tests

+Deployment rehearsals using service verification & environment verification tests in all lower environments

+Dev provide service verification tests

+Ops provide environment verification tests (used by Dev and QA)

+Service service test data setup (including mainframe)



"MyAccount Perf Incident"

History:
12/20/15 - Started seeing problems
12/29/15 - Swarm of problems then everyone on vacation
9am Monday 1/4/16 - everyone back to work and problems spike

!!!
Customer
Call Contact Center
Try to recreate errors
Contact Service Desk
Contact Center
Many calls
"stuff isn't working!"
Tix
Many tickets
Service Now Tickets
Call Service Desk
Try to recreate errors
Check Monitoring
PD
BMC
"All green" (up/down monitoring)
Service Desk
9 AM
2 hours
11 AM
1/5/16
"Terri" Dir. Account Reps
Request
Incident Commander NOC
TS
Start Bridge Call
Assemble War Room
call someone
call someone else
Call Vendor Support
Start Biz Management Bridge
TS
Send Biz Communication Emails
every 4 hours
Update BRM Bridge Call
Update
Update
Update
Update
Update
Update
Update
Incident Commander
TS
M
HB
PD
Web Dev Fix
Test
Services Fix
Test
Portal Ops Fix
Test
DBA Ops Fix
Test
Network Fix
Test
Vendor Partners Fix
Test
Investigate DDOS (Security)
Test
Problem Subsides for Day
~2pm
Problem Spikes!!
~5am
1/6/16
PD
M
TS
H
Repeat War Room Cycles
"Scribes"
Notes .doc
Chat
TS
Problem Subsides for Day
~2pm
Problem Spikes!!
~5am
1/7/16
PD
M
H
TS
Repeat War Room Cycles
"Focused on Stored Procedures but not confident why"
M-F Every Day!
PD
M
H
TS
Repeat War Room Cycles
"Vendor consultant provides config changes"
"Stored Procedure s used in way never seen before!"
1/13/16
QA Run Load Tests to Confirm Fix (After Hours)
Prod
Additional Fix from Vendor Consultant
1/13/16-1/15/16

- Vendor Consultant onsite for 3 days
- QA tried to simulate load
- Tried adding capacity (3 app servers and web)
- Tried disabling

Permanent Fix Options (assumptions):

- Upgrade to latest version (~Q3 2016 go live?)
- Rewrite DB layer of App to remove stored procedures

No user experience monitoring in place
No standard application performance tools available early and often
Limited database diagnostic tools

Vendor reps joining calls aren't correct people

Too much noise on bridge. Engineers can't focus. Open up side bridges because of difficulty of noise / task switching on main bridge

Lots of teams spending many cycles to "prove it wasn't them"

No service owner to manage biz communication and determine if resolved

Experimenting directly in production

Documentation not up to date

Inability to rollback recent changes with any confidence ("probably impossible?"). "Fight forward" only.

Lots of people coming in and out.

Can't trace user transactions through the stack

No controlled "prod-like" stage env for troubleshooting

Vendor provided "fix" is really just a workaround

Resolution loop was never really closed. "Bandaid fix" is still in place!

1. Service performance monitoring, error detection, and automated health checks starting Dev

2. Improved vendor SLA management

3. Improved code review

5. "Prod-Like" Pre-Prod environments (with load testing)

6. Platform App Dev leading new deployment strategy to improve traceability and ease rollback

7. Improved / Streamlined Communication in War Room / Bridge

8. Follow through on resolution

Also In-flight:

- *Peer review for every check-in*
- *Automation tooling and SDLC for DB changes (no more adhoc scripts)*

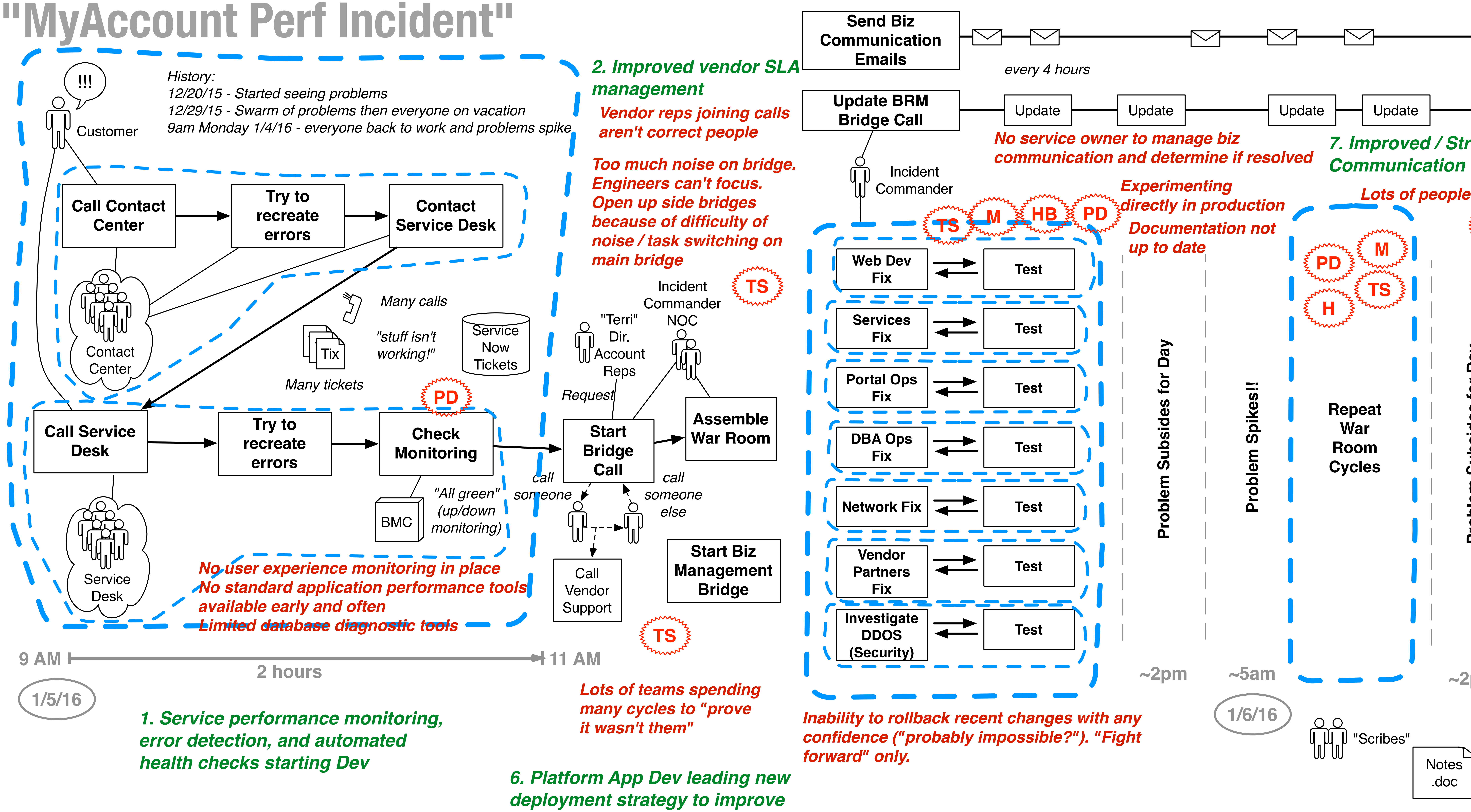
War Room and Bridge Cost (direct labor):
M-F: 35 h/c per day for 7 days = $195,000
S,S: 6 h/c per day for 2 days = $11,400
Total: $206,400

Impact on Call Centers:
1500 Agents 30% degraded 8 hrs for 7 days = $806,000

Cost of impact/delay on other inflight projects:
Unknown ("feels large", estimated at 20% - 100%)
Cost of impact/delay on compliance issues:
Unknown
Cost of brand damage:
Unknown

TOTAL COST OF INCIDENT:
$1,012,000 ++

"MyAccount Perf Incident"



History:
12/20/15 - Started seeing problems
12/29/15 - Swarm of problems then everyone on vacation
9am Monday 1/4/16 - everyone back to work and problems spike

No user experience monitoring in place
No standard application performance tools available early and often
~~Limited database diagnostic tools~~

1. Service performance monitoring, error detection, and automated health checks starting Dev

2. Improved vendor SLA management

Vendor reps joining calls aren't correct people

Too much noise on bridge. Engineers can't focus. Open up side bridges because of difficulty of noise / task switching on main bridge

Lots of teams spending many cycles to "prove it wasn't them"

6. Platform App Dev leading new deployment strategy to improve traceability and ease rollback

No service owner to manage biz communication and determine if resolved

Experimenting directly in production

Documentation not up to date

Inability to rollback recent changes with any confidence ("probably impossible?"). "Fight forward" only.

7. Improved / Str
Communication

Lots of people

War Room and Bridge Cost (direct labor):
M-F: 35 h/c per day for 7 days = $195,000
S,S: 6 h/c per day for 2 days = $11,400
Total: $206,400

"MyAccount Perf Incident"

!!!
Customer
Call Contact Center
Try to recreate errors
Contact Service Desk
Contact Center
Many calls
"stuff isn't working!"
Tix
Many tickets
Service Now Tickets
Call Service Desk
Try to recreate errors
Check Monitoring
PD
BMC
"All green" (up/down monitoring)
Service Desk
9 AM
2 hours
11 AM
1/5/16
"Terri" Dir. Account Reps
Incident Commander NOC
TS
Request
Start Bridge Call
Assemble War Room
call someone
call someone else
Call Vendor Support
Start Biz Management Bridge
TS
Send Biz Communication Emails
every 4 hours
Update BRM Bridge Call
Update
Update
Update
Update
Update
Update
Update
Incident Commander
TS
M
HB
PD
Web Dev Fix
Test
Services Fix
Test
Portal Ops Fix
Test
DBA Ops Fix
Test
Network Fix
Test
Vendor Partners Fix
Test
Investigate DDOS (Security)
Test
Problem Subsides for Day
~2pm
Problem Spikes!!
~5am
1/6/16
PD
M
TS
H
Repeat War Room Cycles
"Scribes"
Notes .doc
Chat
TS
Problem Subsides for Day
~2pm
Problem Spikes!!
~5am
1/7/16
PD
M
TS
H
Repeat War Room Cycles
M-F Every Day!
PD
M
TS
H
Repeat War Room Cycles
1/13/16
QA Run Load Tests to Confirm Fix (After Hours)
Prod
Additional Fix from Vendor Consultant
1/13/16-1/15/16

History:
12/20/15 - Started seeing problems
12/29/15 - Swarm of problems then everyone on vacation
9am Monday 1/4/16 - everyone back to work and problems spike

No user experience monitoring in place
No standard application performance tools available early and often
~~Limited database diagnostic tools~~

1. Service performance monitoring, error detection, and automated health checks starting Dev

2. Improved vendor SLA management

Vendor reps joining calls aren't correct people

Too much noise on bridge. Engineers can't focus. Open up side bridges because of difficulty of noise / task switching on main bridge

Lots of teams spending many cycles to "prove it wasn't them"

6. Platform App Dev leading new deployment strategy to improve traceability and ease rollback

No service owner to manage biz communication and determine if resolved

Experimenting directly in production

Documentation not up to date

Inability to rollback recent changes with any confidence ("probably impossible?"). "Fight forward" only.

7. Improved / Streamlined Communication in War Room / Bridge

Lots of people coming in and out.

Can't trace user transactions through the stack

No controlled "prod-like" stage env for troubleshooting

"Focused on Stored Procedures but not confident why"

- Vendor Consultant onsite for 3 days
- QA tried to simulate load
- Tried adding capacity (3 app servers and web)
- Tried disabling

"Vendor consultant provides config changes"

"Stored Procedure s used in way never seen before!"

5. "Prod-Like" Pre-Prod environments (with load testing)

Vendor provided "fix" is really just a workaround

3. Improved code review

8. Follow through on resolution

Resolution loop was never really closed. "Bandaid fix" is still in place!

Permanent Fix Options (assumptions):

- Upgrade to latest version (~Q3 2016 go live?)
- Rewrite DB layer of App to remove stored procedures

Also In-flight:

- Peer review for every check-in
- Automation tooling and SDLC for DB changes (no more adhoc scripts)

War Room and Bridge Cost (direct labor):
M-F: 35 h/c per day for 7 days = $195,000
S,S: 6 h/c per day for 2 days = $11,400
Total: $206,400

Impact on Call Centers:
1500 Agents 30% degraded 8 hrs for 7 days = $806,000

Cost of impact/delay on other inflight projects:
Unknown ("feels large", estimated at 20% - 100%)
Cost of impact/delay on compliance issues:
Unknown
Cost of brand damage:
Unknown

TOTAL COST OF INCIDENT:
$1,012,000 ++

Send Biz Communication Emails

every 4 hours

Update BRM Bridge Call

Update

Update

Update

Update

Update

Update

Update

Incident Commander

No service owner to manage biz communication and determine if resolved

7. Improved / Streamlined Communication in War Room / Bridge

Can't trace user transactions through the stack

No controlled "prod-like" stage env for troubleshooting

Experimenting directly in production

Documentation not up to date

Lots of people coming in and out.

TS M HB PD

Web Dev Fix — Test

Services Fix — Test

Portal Ops Fix — Test

DBA Ops Fix — Test

Network Fix — Test

Vendor Partners Fix — Test

Investigate DDOS (Security) — Test

Problem Subsides for Day

~2pm

Problem Spikes!!

~5am

1/6/16

PD M TS H

Repeat War Room Cycles

"Scribes"

Notes .doc

Chat

TS

Problem Subsides for Day

~2pm

Problem Spikes!!

~5am

1/7/16

PD M TS H

Repeat War Room Cycles

"Focused on Stored Procedures but not confident why"

M-F Every Day!

PD M TS H

Repeat War Room Cycles

"Vendor consultant provides config changes"

"Stored Procedure s used in way never seen before!"

1/13/16

5. "Prod-Like" Pre-Prod environments (with load testing)

Vendor provided "fix" is really just a workaround

8. Follow through on resolution

Resolution loop was never really closed. "Bandaid fix" is still in place!

QA Run Load Tests to Confirm Fix (After Hours)

Prod

Additional Fix from Vendor Consultant

Permanent Fix Options (assumptions):

- Upgrade to latest version (~Q3 2016 go live?)
- Rewrite DB layer of App to remove stored procedures

3. Improved code review

Also In-flight:

- Peer review for every check-in
- Automation tooling and SDLC for DB changes (no more adhoc scripts)

1/13/16-1/15/16

nability to rollback recent changes with any confidence ("probably impossible?"). "Fight forward" only.

- Vendor Consultant onsite for 3 days
- QA tried to simulate load
- Tried adding capacity (3 app servers and web)
- Tried disabling

War Room and Bridge Cost (direct labor):
M-F: 35 h/c per day for 7 days = $195,000
S,S: 6 h/c per day for 2 days = $11,400
Total: $206,400

Impact on Call Centers:
1500 Agents 30% degraded 8 hrs for 7 days = $806,000

Cost of impact/delay on other inflight projects:
Unknown ("feels large", estimated at 20% - 100%)
Cost of impact/delay on compliance issues:
Unknown
Cost of brand damage:
Unknown

TOTAL COST OF INCIDENT:
$1,012,000 ++

"MyAccount Perf Incident"

History:
12/20/15 - Started seeing problems
12/29/15 - Swarm of problems then everyone on vacation
9am Monday 1/4/16 - everyone back to work and problems spike

!!!
Customer
Call Contact Center
Try to recreate errors
Contact Service Desk
Contact Center
Many calls
"stuff isn't working!"
Tix
Many tickets
Service Now Tickets
Call Service Desk
Try to recreate errors
PD
Check Monitoring
BMC
"All green" (up/down monitoring)
Service Desk
9 AM
2 hours
11 AM
1/5/16

No user experience monitoring in place
No standard application performance tools available early and often
Limited database diagnostic tools

1. Service performance monitoring, error detection, and automated health checks starting Dev

2. Improved vendor SLA management

Vendor reps joining calls aren't correct people

Too much noise on bridge. Engineers can't focus. Open up side bridges because of difficulty of noise / task switching on main bridge

Lots of teams spending many cycles to "prove it wasn't them"

6. Platform App Dev leading new deployment strategy to improve traceability and ease rollback

No service owner to manage biz communication and determine if resolved

Experimenting directly in production

Documentation not up to date

Inability to rollback recent changes with any confidence ("probably impossible?"). "Fight forward" only.

7. Improved / Streamlined Communication in War Room / Bridge

Lots of people coming in and out.

Can't trace user transactions through the stack

No controlled "prod-like" stage env for troubleshooting

- Vendor Consultant onsite for 3 days
- QA tried to simulate load
- Tried adding capacity (3 app servers and web)
- Tried disabling

5. "Prod-Like" Pre-Prod environments (with load testing)

Vendor provided "fix" is really just a workaround

3. Improved code review

8. Follow through on resolution

Resolution loop was never really closed. "Bandaid fix" is still in place!

Permanent Fix Options (assumptions):
- Upgrade to latest version (~Q3 2016 go live?)
- Rewrite DB layer of App to remove stored procedures

Also In-flight:
- Peer review for every check-in
- Automation tooling and SDLC for DB changes (no more adhoc scripts)

War Room and Bridge Cost (direct labor):
M-F: 35 h/c per day for 7 days = $195,000
S,S: 6 h/c per day for 2 days = $11,400
Total: $206,400

Impact on Call Centers:
1500 Agents 30% degraded 8 hrs for 7 days = $806,000

Cost of impact/delay on other inflight projects:
Unknown ("feels large", estimated at 20% - 100%)
Cost of impact/delay on compliance issues:
Unknown
Cost of brand damage:
Unknown

TOTAL COST OF INCIDENT:
$1,012,000 ++

place!

Problem Subside

Problem Spike

Repeat War Room Cycles

Problem Subsides

Problem Spike

Room Cycles

"Focused on Stored Procedures but not confident why"

M-F Every Day!

Cycles

"Vendor consultant provides config changes"

"Stored Procedure s used in way never seen before!"

QA Run Load Tests to Confirm Fix (After Hours)

Additional Fix from Vendor Consultant

Prod

Permanent
(assum
- Upgrade
(~Q3 201
- Rewrite D
to remove
procedur

~2pm

~5am

~2pm

~5am

3. Improved code review

Also In-flight:
- *Peer review for every chec*
- *Automation tooling and S*
DB changes (no more adh

with any
"). "Fight

1/6/16

"Scribes"

Notes .doc

Chat

1/7/16

- *Vendor Consultant onsite for 3 days*
- *QA tried to simulate load*
- *Tried adding capacity (3 app servers and web)*
- *Tried disabling*

1/13/16

1/13/16-1/15/16

War Room and Bridge Cost (direct labor):
M-F: 35 h/c per day for 7 days = $195,000
S,S: 6 h/c per day for 2 days = $11,400
Total: $206,400

Impact on Call Centers:
1500 Agents 30% degraded 8 hrs for 7 days = $806,000

Cost of impact/delay on other inflight projects:
Unknown ("feels large", estimated at 20% - 100%)
Cost of impact/delay on compliance issues:
Unknown
Cost of brand damage:
Unknown

TOTAL COST OF I
$1,012,000 ++

Problem Subside

Problem Spike

Repeat War Room Cycles

Problem Subsides

Problem Spike

Room Cycles

"Focused on Stored Procedures but not confident why"

M-F Every Day!

Cycles

"Vendor consultant provides config changes"

"Stored Procedure s used in way never seen before!"

QA Run Load Tests to Confirm Fix (After Hours)

Prod

Additional Fix from Vendor Consultant

place!

Permanent
(assum

- **Upgrade**
(~Q3 201
- **Rewrite D**
to remov
procedur

3. Improved code review

Also In-flight:

- ***Peer review for every chec***
- ***Automation tooling and S***
DB changes (no more adh

~2pm ~5am ~2pm ~5am

1/6/16

1/7/16

1/13/16

1/13/16-1/15/16

with any
"). "Fight

"Scribes"

Notes .doc

Chat

- *Vendor Consultant onsite for 3 days*
- *QA tried to simulate load*
- *Tried adding capacity (3 app servers and web)*
- *Tried disabling*

War Room and Bridge Cost (direct labor):
M-F: 35 h/c per day for 7 days = $195,000
S,S: 6 h/c per day for 2 days = $11,400
Total: $206,400

Impact on Call Centers:
1500 Agents 30% degraded 8 hrs for 7 days = $806,000

Cost of impact/delay on other inflight projects:
Unknown ("feels large", estimated at 20% - 100%)
Cost of impact/delay on compliance issues:
Unknown
Cost of brand damage:
Unknown

TOTAL COST OF I
$1,012,000 ++

"MyAccount Perf Incident"

History:
12/20/15 - Started seeing problems
12/29/15 - Swarm of problems then everyone on vacation
9am Monday 1/4/16 - everyone back to work and problems spike

!!!
Customer

Call Contact Center → Try to recreate errors → Contact Service Desk

Contact Center

Many calls

"stuff isn't working!"

Tix

Many tickets

Service Now Tickets

Call Service Desk → Try to recreate errors → Check Monitoring → Start Bridge Call → Assemble War Room

Service Desk

PD

BMC

"All green" (up/down monitoring)

No user experience monitoring in place
No standard application performance tools available early and often
Limited database diagnostic tools

9 AM — 2 hours — 11 AM

1/5/16

1. Service performance monitoring, error detection, and automated health checks starting Dev

2. Improved vendor SLA management

Vendor reps joining calls aren't correct people

Too much noise on bridge. Engineers can't focus. Open up side bridges because of difficulty of noise / task switching on main bridge

"Terri" Dir. Account Reps

Request

Incident Commander NOC

TS

call someone

call someone else

Call Vendor Support

Start Biz Management Bridge

TS

Lots of teams spending many cycles to "prove it wasn't them"

6. Platform App Dev leading new deployment strategy to improve traceability and ease rollback

Send Biz Communication Emails

every 4 hours

Update BRM Bridge Call → Update → Update → Update → Update → Update → Update → Update

No service owner to manage biz communication and determine if resolved

Incident Commander

TS M HB PD

Web Dev Fix	Test
Services Fix	Test
Portal Ops Fix	Test
DBA Ops Fix	Test
Network Fix	Test
Vendor Partners Fix	Test
Investigate DDOS (Security)	Test

Experimenting directly in production

Documentation not up to date

Inability to rollback recent changes with any confidence ("probably impossible?"). "Fight forward" only.

Problem Subsides for Day — ~2pm

Problem Spikes!! — ~5am

1/6/16

7. Improved / Streamlined Communication in War Room / Bridge

Lots of people coming in and out.

PD M TS H

Repeat War Room Cycles

"Scribes"

Notes .doc

Chat

TS

Problem Subsides for Day — ~2pm

Problem Spikes!! — ~5am

1/7/16

PD M TS H

Repeat War Room Cycles

"Focused on Stored Procedures but not confident why"

- *Vendor Consultant onsite for 3 days*
- *QA tried to simulate load*
- *Tried adding capacity (3 app servers and web)*
- *Tried disabling*

Can't trace user transactions through the stack

No controlled "prod-like" stage env for troubleshooting

M-F Every Day!

PD M TS H

Repeat War Room Cycles

"Vendor consultant provides config changes"

"Stored Procedure s used in way never seen before!"

1/13/16

3. Improved code review

5. "Prod-Like" Pre-Prod environments (with load testing)

Vendor provided "fix" is really just a workaround

QA Run Load Tests to Confirm Fix (After Hours) → Additional Fix from Vendor Consultant

Prod

1/13/16-1/15/16

Permanent Fix Options (assumptions):

- **Upgrade to latest version (~Q3 2016 go live?)**
- **Rewrite DB layer of App to remove stored procedures**

8. Follow through on resolution

Resolution loop was never really closed. "Bandaid fix" is still in place!

Also In-flight:

- ***Peer review for every check-in***
- ***Automation tooling and SDLC for DB changes (no more adhoc scripts)***

War Room and Bridge Cost (direct labor):
M-F: 35 h/c per day for 7 days = $195,000
S,S: 6 h/c per day for 2 days = $11,400
Total: $206,400

Impact on Call Centers:
1500 Agents 30% degraded 8 hrs for 7 days = $806,000

Cost of impact/delay on other inflight projects:
Unknown ("feels large", estimated at 20% - 100%)
Cost of impact/delay on compliance issues:
Unknown
Cost of brand damage:
Unknown

TOTAL COST OF INCIDENT:
$1,012,000 ++

"MyAccount Perf Incident"

+ Automated service verification and performance health checks starting in Dev

+Ops added to code peer reviews

+Streamlined war rom and bridge call communication

History:
12/20/15 - Started seeing problems
12/29/15 - Swarm of problems then everyone on vacation
9am Monday 1/4/16 - everyone back to work and problems spike

!!!
Customer
Call Contact Center
Try to recreate errors
Contact Service Desk
Contact Center
Many calls
"stuff isn't working!"
Tix
Service Now Tickets
Many tickets
Call Service Desk
Try to recreate errors
Check Monitoring
"All green" (up/down monitoring)
BMC
Service Desk
Start Bridge Call
call someone
call someone else
Request
"Terri" Dir. Account Reps
Incident Commander
NOC
Assemble War Room
Start Biz Management Bridge
Call Vendor Support
TS
PD
9 AM
2 hours
11 AM
1/5/16
Send Biz Communication Emails
every 4 hours
Update BRM Bridge Call
Update
Update
Update
Update
Update
Update
Update
Incident Commander
TS
M
HB
PD
Web Dev Fix
Services Fix
Portal Ops Fix
DBA Ops Fix
Network Fix
Vendor Partners Fix
Investigate DDOS (Security)
Test
Test
Test
Test
Test
Test
Test
Problem Subsides for Day
~2pm
Problem Spikes!!
~5am
1/6/16
PD
M
TS
H
Repeat War Room Cycles
TS
Problem Subsides for Day
~2pm
Problem Spikes!!
~5am
1/7/16
PD
M
TS
H
Repeat War Room Cycles
M-F Every Day!
PD
M
TS
H
Repeat War Room Cycles
1/13/16
"Scribes"
Notes .doc
Chat
QA Run Load Tests to Confirm Fix (After Hours)
Prod
Additional Fix from Vendor Consultant
1/13/16-1/15/16

No user experience monitoring in place
No standard application performance tools available early and often
Limited database diagnostic tools

1. Service performance monitoring, error detection, and automated health checks starting Dev

2. Improved vendor SLA management

Vendor reps joining calls aren't correct people

Too much noise on bridge. Engineers can't focus. Open up side bridges because of difficulty of noise / task switching on main bridge

Lots of teams spending many cycles to "prove it wasn't them"

6. Platform App Dev leading new deployment strategy to improve traceability and ease rollback

No service owner to manage biz communication and determine if resolved

Experimenting directly in production

Documentation not up to date

Inability to rollback recent changes with any confidence ("probably impossible?"). "Fight forward" only.

7. Improved / Streamlined Communication in War Room / Bridge

Lots of people coming in and out.

Can't trace user transactions through the stack

No controlled "prod-like" stage env for troubleshooting

"Focused on Stored Procedures but not confident why"

"Vendor consultant provides config changes"

"Stored Procedure s used in way never seen before!"

- Vendor Consultant onsite for 3 days
- QA tried to simulate load
- Tried adding capacity (3 app servers and web)
- Tried disabling

5. "Prod-Like" Pre-Prod environments (with load testing)

Vendor provided "fix" is really just a workaround

3. Improved code review

8. Follow through on resolution

Resolution loop was never really closed. "Bandaid fix" is still in place!

Permanent Fix Options (assumptions):

- Upgrade to latest version (~Q3 2016 go live?)
- Rewrite DB layer of App to remove stored procedures

Also In-flight:

- Peer review for every check-in
- Automation tooling and SDLC for DB changes (no more adhoc scripts)

War Room and Bridge Cost (direct labor):
M-F: 35 h/c per day for 7 days = $195,000
S,S: 6 h/c per day for 2 days = $11,400
Total: $206,400

Impact on Call Centers:
1500 Agents 30% degraded 8 hrs for 7 days = $806,000

Cost of impact/delay on other inflight projects:
Unknown ("feels large", estimated at 20% - 100%)
Cost of impact/delay on compliance issues:
Unknown
Cost of brand damage:
Unknown

TOTAL COST OF INCIDENT:
$1,012,000 ++

"MyAccount Perf Incident"

+ Automated service verification and performance health checks starting in Dev

+Ops added to code peer reviews

+Streamlined war rom and bridge call communication

+ “Prod-like” testing environments (with load testing)

+Automation and SDLC for database changes

+Follow through on resolution to achieve permanent fixes

History:
12/20/15 - Started seeing problems
12/29/15 - Swarm of problems then everyone on vacation
9am Monday 1/4/16 - everyone back to work and problems spike

1. Service performance monitoring, error detection, and automated health checks starting Dev

2. Improved vendor SLA management

Vendor reps joining calls aren't correct people

Too much noise on bridge. Engineers can't focus. Open up side bridges because of difficulty of noise / task switching on main bridge

5. "Prod-Like" Pre-Prod environments (with load testing)

6. Platform App Dev leading new deployment strategy to improve traceability and ease rollback

7. Improved / Streamlined Communication in War Room / Bridge

8. Follow through on resolution

No service owner to manage biz communication and determine if resolved

Experimenting directly in production

Documentation not up to date

Lots of people coming in and out.

Can't trace user transactions through the stack

No controlled "prod-like" stage env for troubleshooting

No user experience monitoring in place

No standard application performance tools available early and often

Limited database diagnostic tools

Lots of teams spending many cycles to "prove it wasn't them"

Inability to rollback recent changes with any confidence ("probably impossible?"). "Fight forward" only.

Vendor provided "fix" is really just a workaround

Resolution loop was never really closed. "Bandaid fix" is still in place!

Permanent Fix Options (assumptions):
- Upgrade to latest version (~Q3 2016 go live?)
- Rewrite DB layer of App to remove stored procedures

Also In-flight:
- Peer review for every check-in
- Automation tooling and SDLC for DB changes (no more adhoc scripts)

- Vendor Consultant onsite for 3 days
- QA tried to simulate load
- Tried adding capacity (3 app servers and web)
- Tried disabling

War Room and Bridge Cost (direct labor):
M-F: 35 h/c per day for 7 days = $195,000
S,S: 6 h/c per day for 2 days = $11,400
Total: $206,400

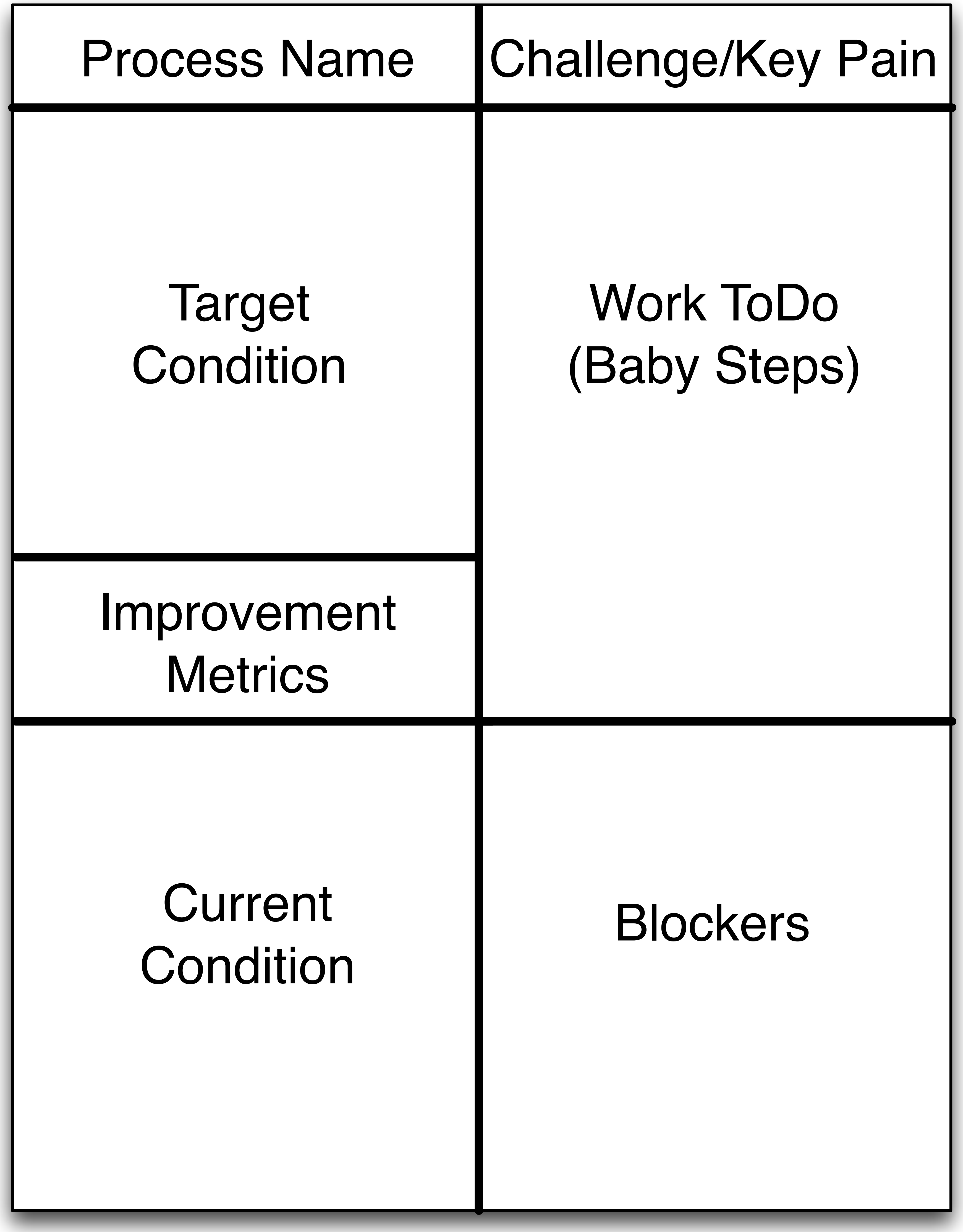
Impact on Call Centers:
1500 Agents 30% degraded 8 hrs for 7 days = $806,000

Cost of impact/delay on other inflight projects:
Unknown ("feels large", estimated at 20% - 100%)
Cost of impact/delay on compliance issues:
Unknown
Cost of brand damage:
Unknown

TOTAL COST OF INCIDENT:
$1,012,000 ++

Improvement Storyboards

Template



Process Name

Challenge/Key Pain

Target Condition

Work ToDo (Baby Steps)

Improvement Metrics

Current Condition

Blockers

Example

Process

GTM/LTM (Traffic manager configuration process)

Challenge/Key Pain

Changes are being introduced / tested in production the first time causing delays, rework, outages

Target Condition

- GM/TLM functionality across all SDLC environments (capex request needed)
- change window reduction for non-prod environments (turn those around instantaneously less than 13 days)
- Provide read-only to all F5 consoles
- Standardize GM pattern

Improvement Metrics

- Lead Time (post-dev to prod)
- Scrap Rate

Work ToDo (Baby Steps)

- Acquire the F5 hardware or software to support envs throughout SDLC
- Make these changes L3 or 5 change requests
- Write automation scripts
- provide read only access to all environments… can include API access to facilitate automation script writing
- Create design template with customer pattern

Current Condition

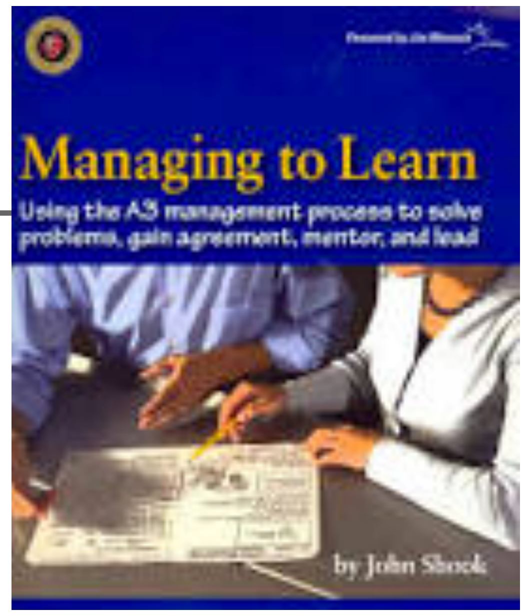
- Apps are not developed in production-like environments (not testing F5 behavior)
- Ops teams cannot practice or learn
- App teams have no visibility into constraints
- No remediation capabilities for app support teams
- No repeatable pattern for GM health activity
- 80% S/R with 2-3 rework cycles
- 50% cause outages

Blockers

- Financial approval (Jennifer)
- Segregation between environments (Mark)
- Non-standard request types (Susan)
- Two network teams with different rules (Mark)

Improvement Storyboards

Inspiration: A3 management process


Managing to Learn

Using the A3 management process to solve problems, gain agreement, mentor, and lead

by John Shook

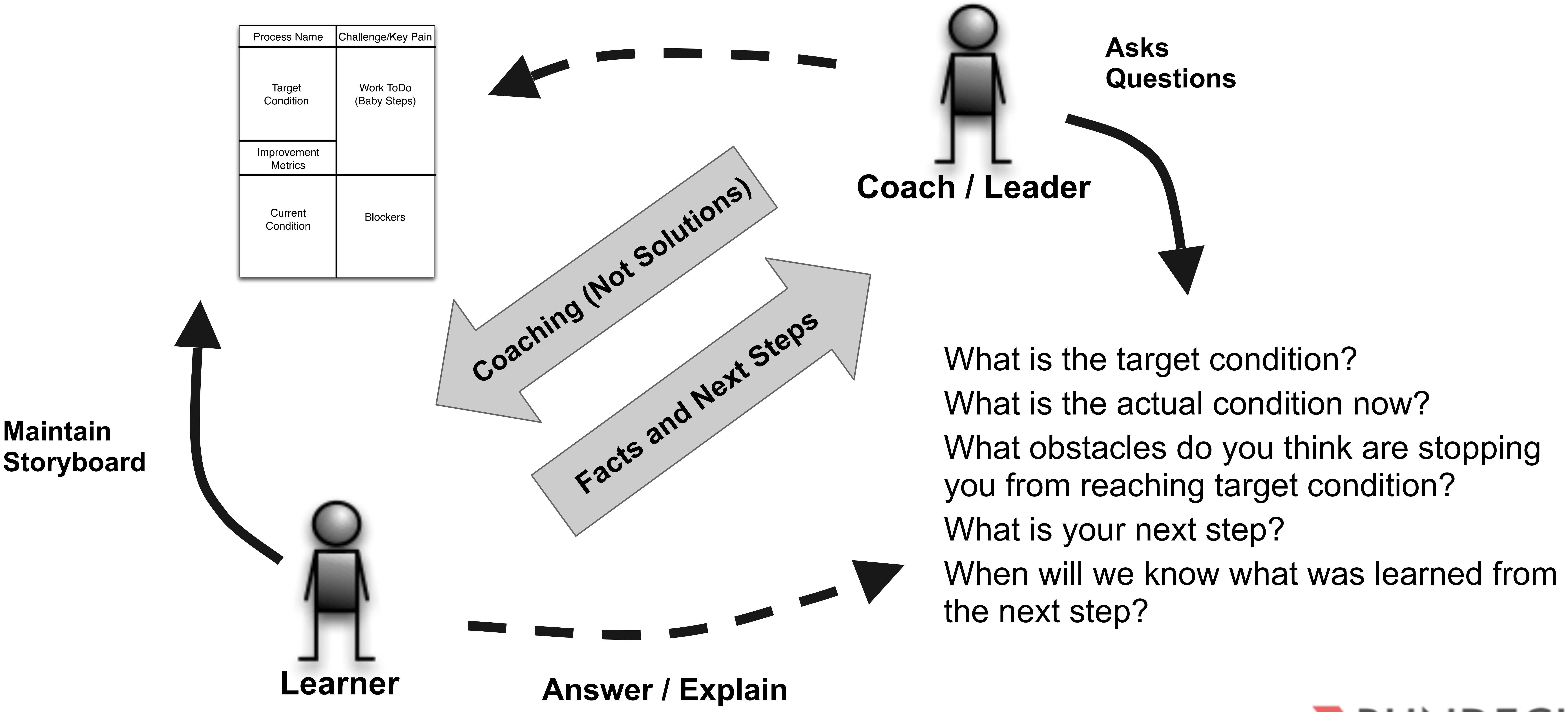
Template

Process Name	Challenge/Key Pain
Target Condition	Work ToDo (Baby Steps)
Improvement Metrics	
Current Condition	Blockers

Example

Process GTM/LTM (Traffic manager configuration process)	**Challenge/Key Pain** Changes are being introduced / tested in production the first time causing delays, rework, outages
Target Condition • GM/TLM functionality across all SDLC environments (capex request needed) • change window reduction for non-prod environments (turn those around instantaneously less than 13 days) • Provide read-only to all F5 consoles • Standardize GM pattern	**Work ToDo (Baby Steps)** • Acquire the F5 hardware or software to support envs throughout SDLC • Make these changes L3 or 5 change requests • Write automation scripts • provide read only access to all environments… can include API access to facilitate automation script writing • Create design template with customer pattern
Improvement Metrics • Lead Time (post-dev to prod) • Scrap Rate	
Current Condition • Apps are not developed in production-like environments (not testing F5 behavior) • Ops teams cannot practice or learn • App teams have no visibility into constraints • No remediation capabilities for app support teams • No repeatable pattern for GM health activity • 80% S/R with 2-3 rework cycles • 50% cause outages	**Blockers** ◦ Financial approval (Jennifer) ◦ Segregation between environments (Mark) ◦ Non-standard request types (Susan) ◦ Two network teams with different rules (Mark)

Using Storyboards: Part "Sales", Part Coaching



Process Name	Challenge/Key Pain
Target Condition	Work ToDo (Baby Steps)
Improvement Metrics	
Current Condition	Blockers

What is the target condition?

What is the actual condition now?

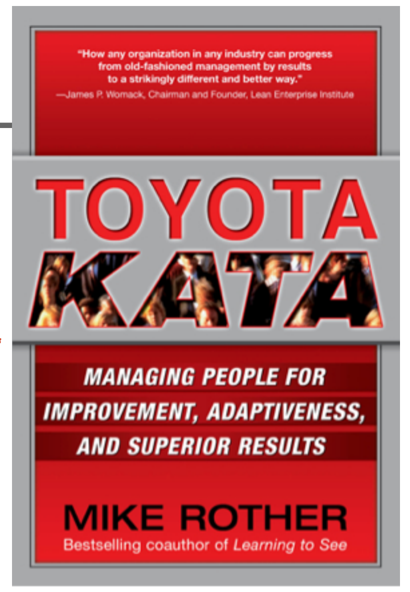
What obstacles do you think are stopping you from reaching target condition?

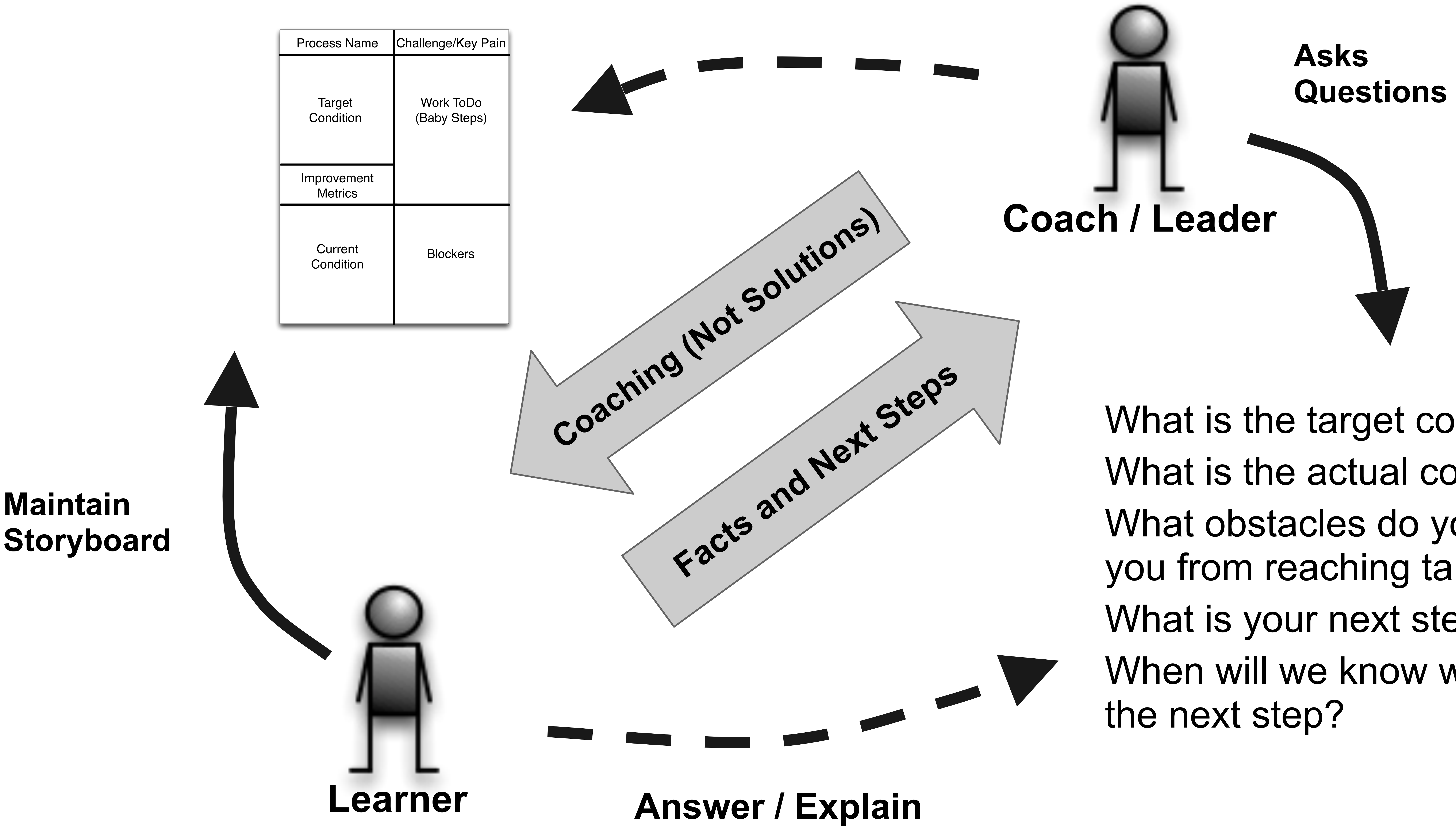
What is your next step?

When will we know what was learned from the next step?

Using Storyboards: Part "Sales", Part Coaching

Inspiration: Toyota Kata


TOYOTA
KATA
MANAGING PEOPLE FOR IMPROVEMENT, ADAPTIVENESS, AND SUPERIOR RESULTS
MIKE ROTHER



Process Name	Challenge/Key Pain
Target Condition	Work ToDo (Baby Steps)
Improvement Metrics	
Current Condition	Blockers

What is the target condition?

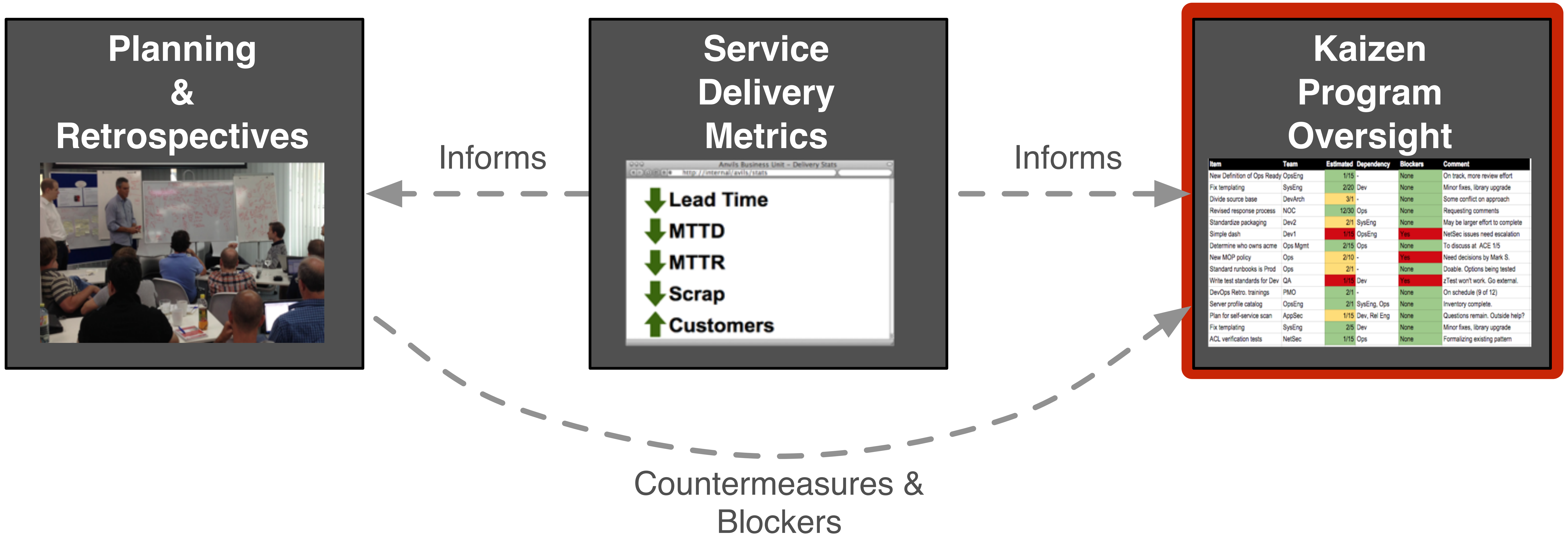
What is the actual condition now?

What obstacles do you think are stopping you from reaching target condition?

What is your next step?

When will we know what was learned from the next step?

Elements of a DevOps Kaizen Program



Planning & Retrospectives

Informs

Service Delivery Metrics

Anvils Business Unit – Delivery Stats

http://internal/avils/stats

- ↓ Lead Time
- ↓ MTTD
- ↓ MTTR
- ↓ Scrap
- ↑ Customers

Informs

Kaizen Program Oversight

Item	Team	Estimated	Dependency	Blockers	Comment
New Definition of Ops Ready	OpsEng	1/15	-	None	On track, more review effort
Fix templating	SysEng	2/20	Dev	None	Minor fixes, library upgrade
Divide source base	DevArch	3/1	-	None	Some conflict on approach
Revised response process	NOC	12/30	Ops	None	Requesting comments
Standardize packaging	Dev2	2/1	SysEng	None	May be larger effort to complete
Simple dash	Dev1	1/15	OpsEng	Yes	NetSec issues need escalation
Determine who owns acme	Ops Mgmt	2/15	Ops	None	To discuss at ACE 1/5
New MOP policy	Ops	2/10	-	Yes	Need decisions by Mark S.
Standard runbooks is Prod	Ops	2/1	-	None	Doable. Options being tested
Write test standards for Dev	QA	1/15	Dev	Yes	zTest won't work. Go external.
DevOps Retro. trainings	PMO	2/1	-	None	On schedule (9 of 12)
Server profile catalog	OpsEng	2/1	SysEng, Ops	None	Inventory complete.
Plan for self-service scan	AppSec	1/15	Dev, Rel Eng	None	Questions remain. Outside help?
Fix templating	SysEng	2/5	Dev	None	Minor fixes, library upgrade
ACL verification tests	NetSec	1/15	Ops	None	Formalizing existing pattern

Countermeasures & Blockers

Kaizen Program Oversight

1. The will to make change happen
2. The resources to make change happen
3. Drive follow-through / clear obstacles

RUNDECK

Kaizen Program Oversight

1. The will to make change happen
2. The resources to make change happen
3. Drive follow-through / clear obstacles

This (and only this) is what the Kaizen Program Oversight Group does!

RUNDECK

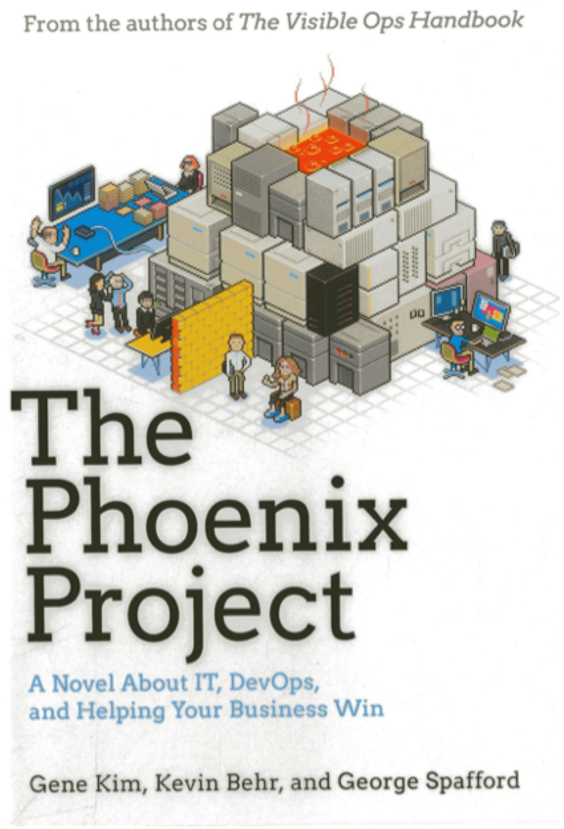
Kaizen Program Oversight

1. The will to make change happen
2. The resources to make change happen
3. Drive follow-through / clear obstacles

Kaizen Program Oversight

1. The will to make change happen
2. The resources to make change happen
3. Drive follow-through / clear obstacles

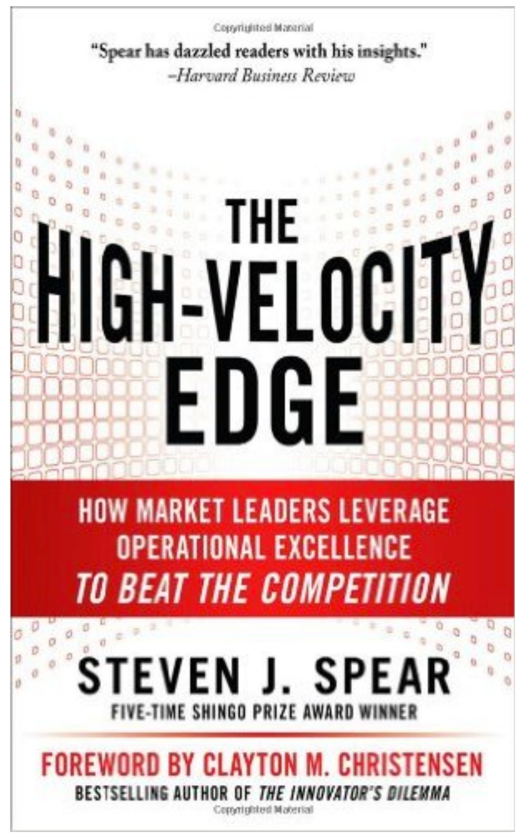
Inspire Executives with:


From the authors of *The Visible Ops Handbook*

The Phoenix Project

A Novel About IT, DevOps, and Helping Your Business Win

Gene Kim, Kevin Behr, and George Spafford


Copyrighted Material

"Spear has dazzled readers with his insights."
–Harvard Business Review

THE HIGH-VELOCITY EDGE

HOW MARKET LEADERS LEVERAGE OPERATIONAL EXCELLENCE *TO BEAT THE COMPETITION*

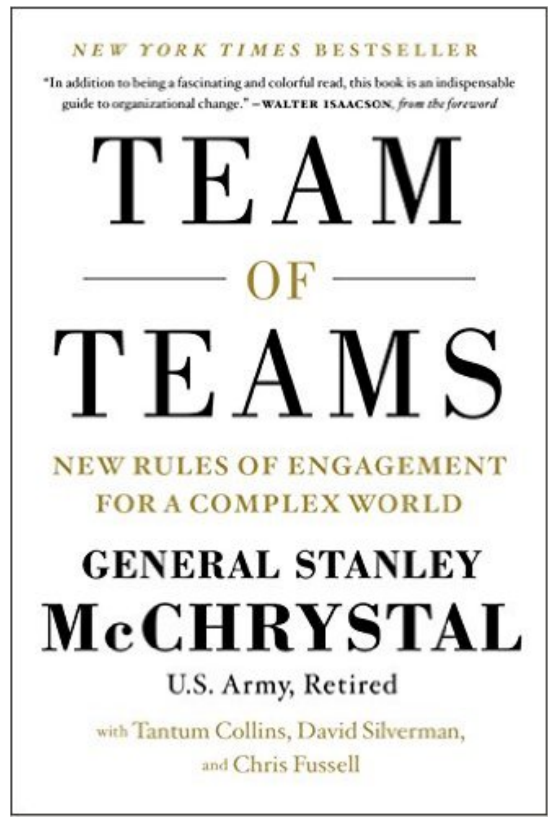
STEVEN J. SPEAR

FIVE-TIME SHINGO PRIZE AWARD WINNER

FOREWORD BY CLAYTON M. CHRISTENSEN

BESTSELLING AUTHOR OF *THE INNOVATOR'S DILEMMA*

Copyrighted Material


NEW YORK TIMES BESTSELLER

"In addition to being a fascinating and colorful read, this book is an indispensable guide to organizational change." –WALTER ISAACSON, *from the foreword*

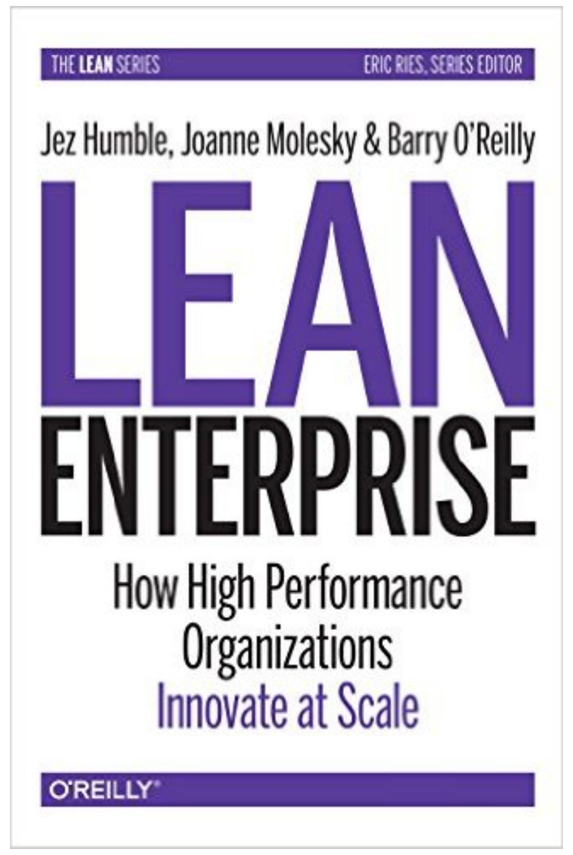
TEAM OF TEAMS

NEW RULES OF ENGAGEMENT FOR A COMPLEX WORLD

GENERAL STANLEY McCHRYSTAL

U.S. Army, Retired

with Tantum Collins, David Silverman, and Chris Fussell

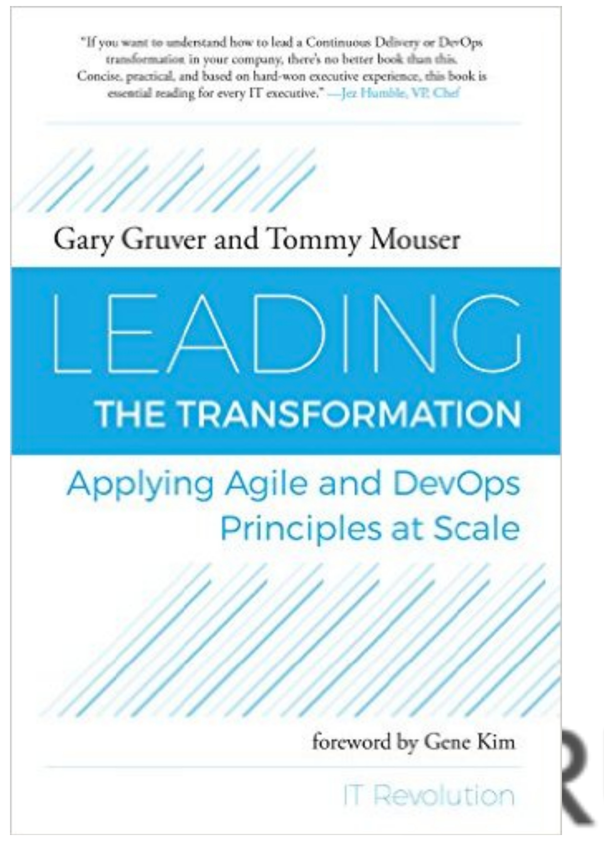

THE LEAN SERIES ERIC RIES, SERIES EDITOR

Jez Humble, Joanne Molesky & Barry O'Reilly

LEAN ENTERPRISE

How High Performance Organizations Innovate at Scale

O'REILLY


"If you want to understand how to lead a Continuous Delivery or DevOps transformation in your company, there's no better book than this. Concise, practical, and based on hard-won executive experience, this book is essential reading for every IT executive." —Jez Humble, VP, Chef

Gary Gruver and Tommy Mouser

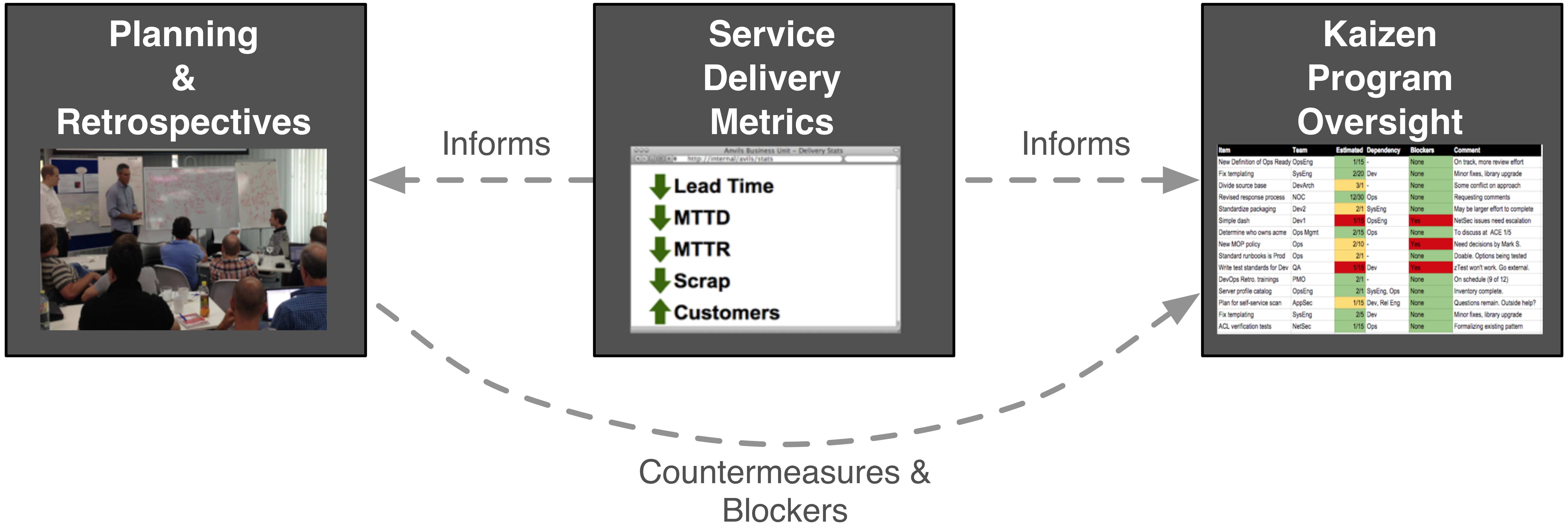
LEADING THE TRANSFORMATION

Applying Agile and DevOps Principles at Scale

foreword by Gene Kim

IT Revolution

Elements of a DevOps Kaizen Program



Planning & Retrospectives

Service Delivery Metrics

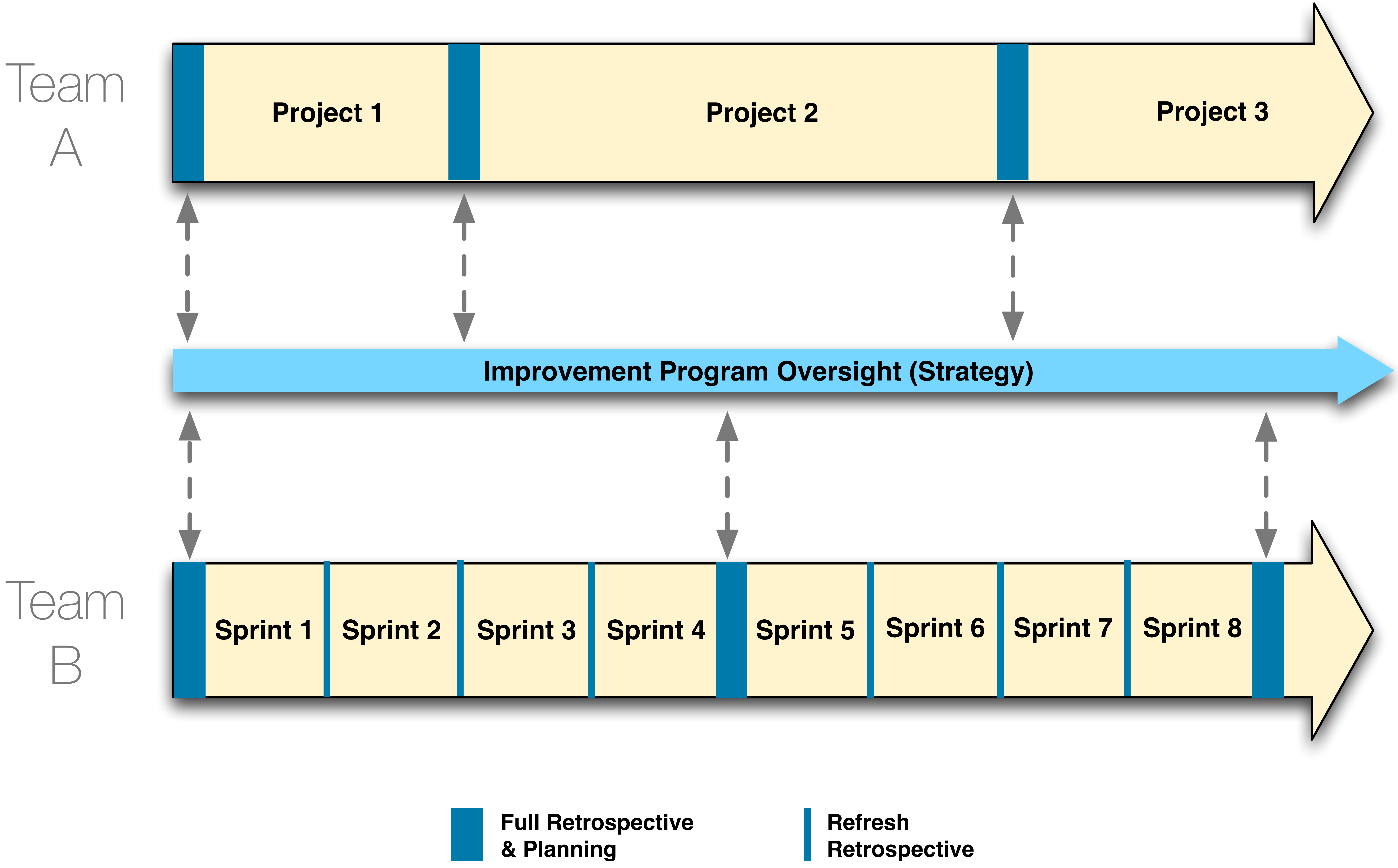
Anvils Business Unit – Delivery Stats
http://internal/avils/stats

- ⬇ Lead Time
- ⬇ MTTD
- ⬇ MTTR
- ⬇ Scrap
- ⬆ Customers

Kaizen Program Oversight

Item	Team	Estimated	Dependency	Blockers	Comment
New Definition of Ops Ready	OpsEng	1/15	-	None	On track, more review effort
Fix templating	SysEng	2/20	Dev	None	Minor fixes, library upgrade
Divide source base	DevArch	3/1	-	None	Some conflict on approach
Revised response process	NOC	12/30	Ops	None	Requesting comments
Standardize packaging	Dev2	2/1	SysEng	None	May be larger effort to complete
Simple dash	Dev1	1/15	OpsEng	Yes	NetSec issues need escalation
Determine who owns acme	Ops Mgmt	2/15	Ops	None	To discuss at ACE 1/5
New MOP policy	Ops	2/10	-	Yes	Need decisions by Mark S.
Standard runbooks is Prod	Ops	2/1	-	None	Doable. Options being tested
Write test standards for Dev	QA	1/15	Dev	Yes	zTest won't work. Go external.
DevOps Retro. trainings	PMO	2/1	-	None	On schedule (9 of 12)
Server profile catalog	OpsEng	2/1	SysEng, Ops	None	Inventory complete.
Plan for self-service scan	AppSec	1/15	Dev, Rel Eng	None	Questions remain. Outside help?
Fix templating	SysEng	2/5	Dev	None	Minor fixes, library upgrade
ACL verification tests	NetSec	1/15	Ops	None	Formalizing existing pattern

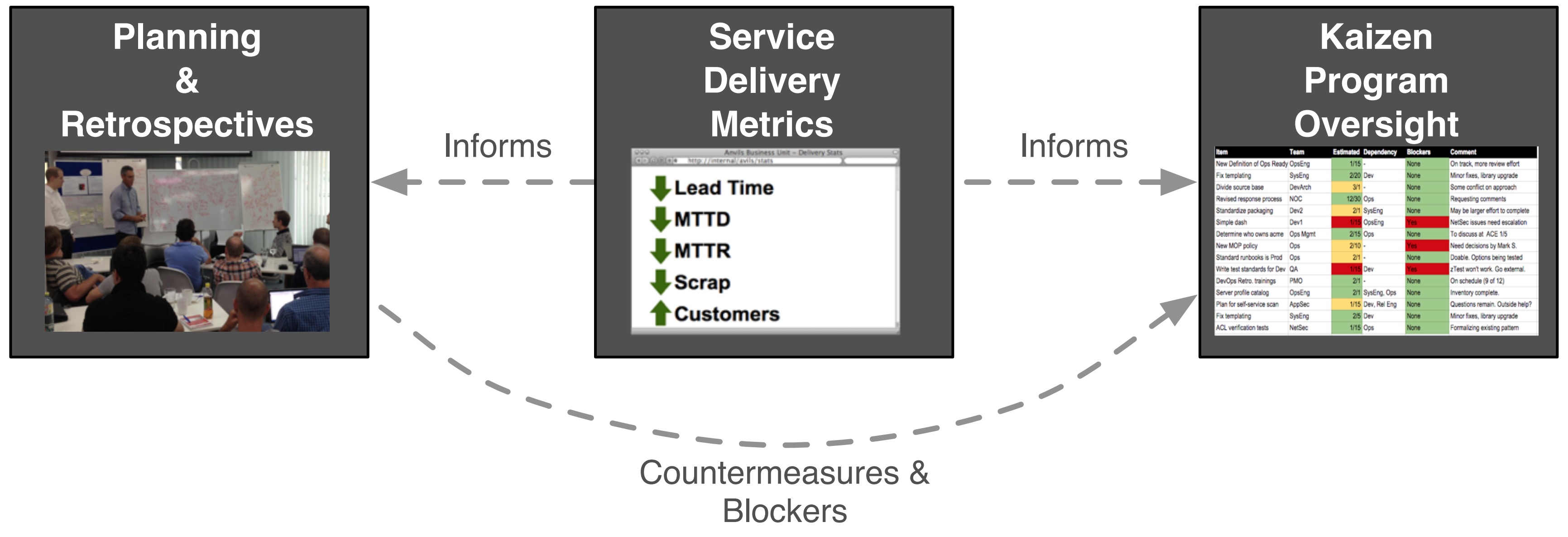
DevOps Kaizen Program is an overlay for any delivery methodology



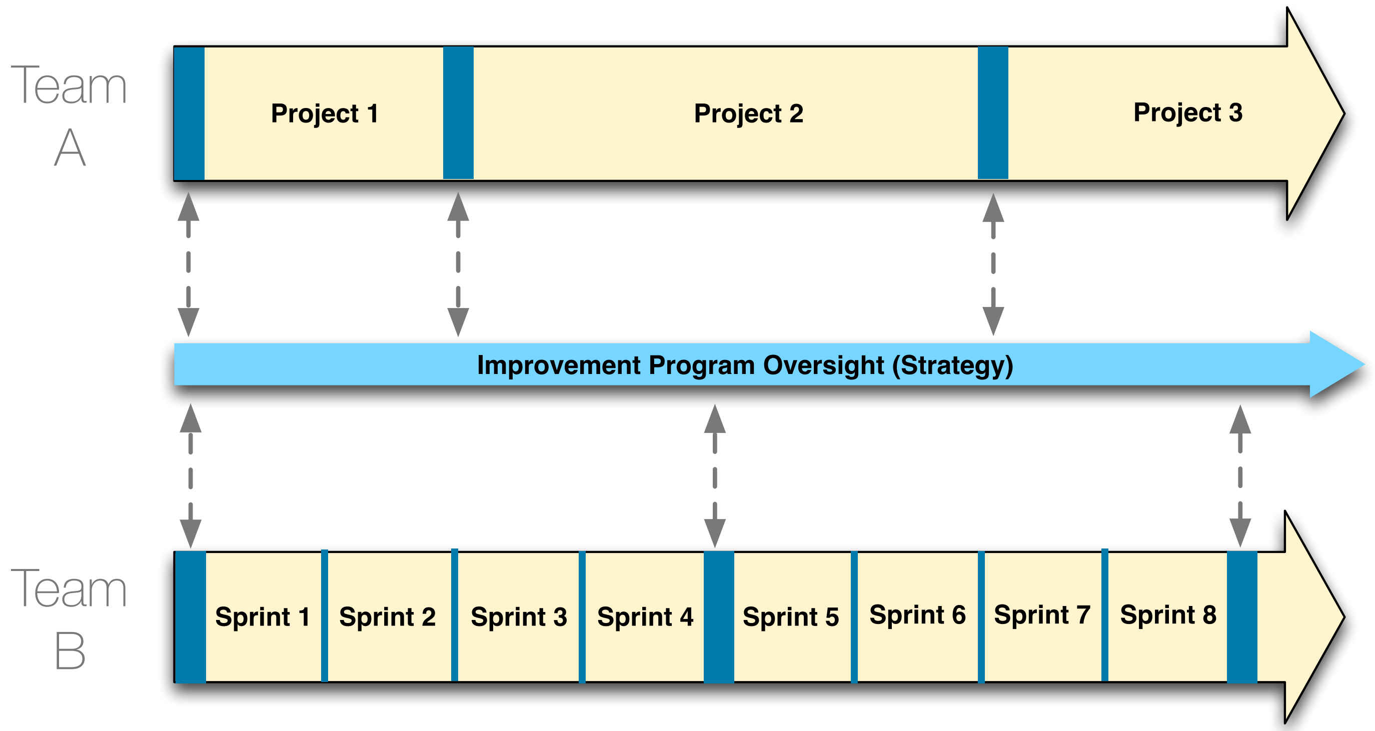
RUNDECK

DevOps Kaizen: *Let’s Recap!*

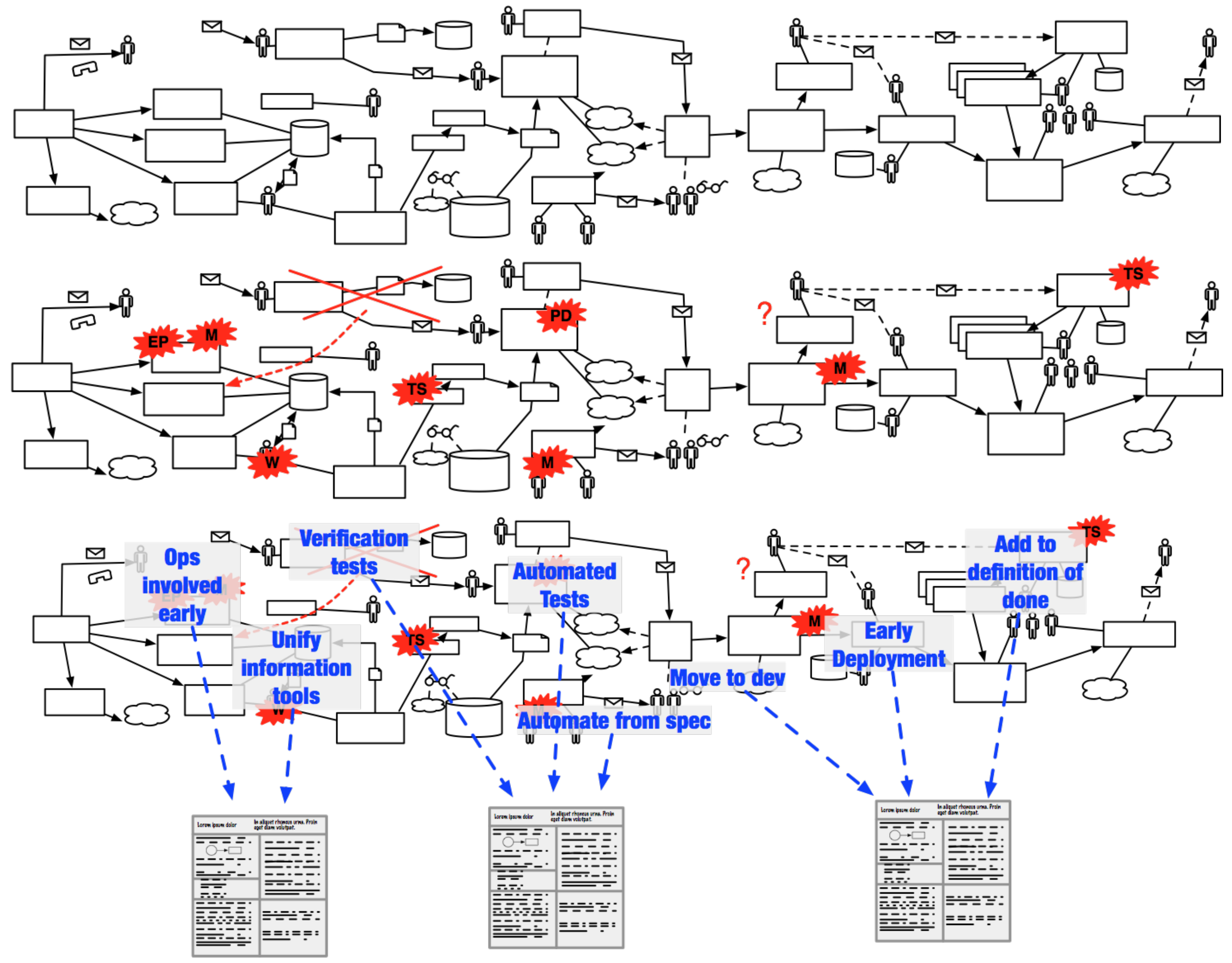
Establish program elements



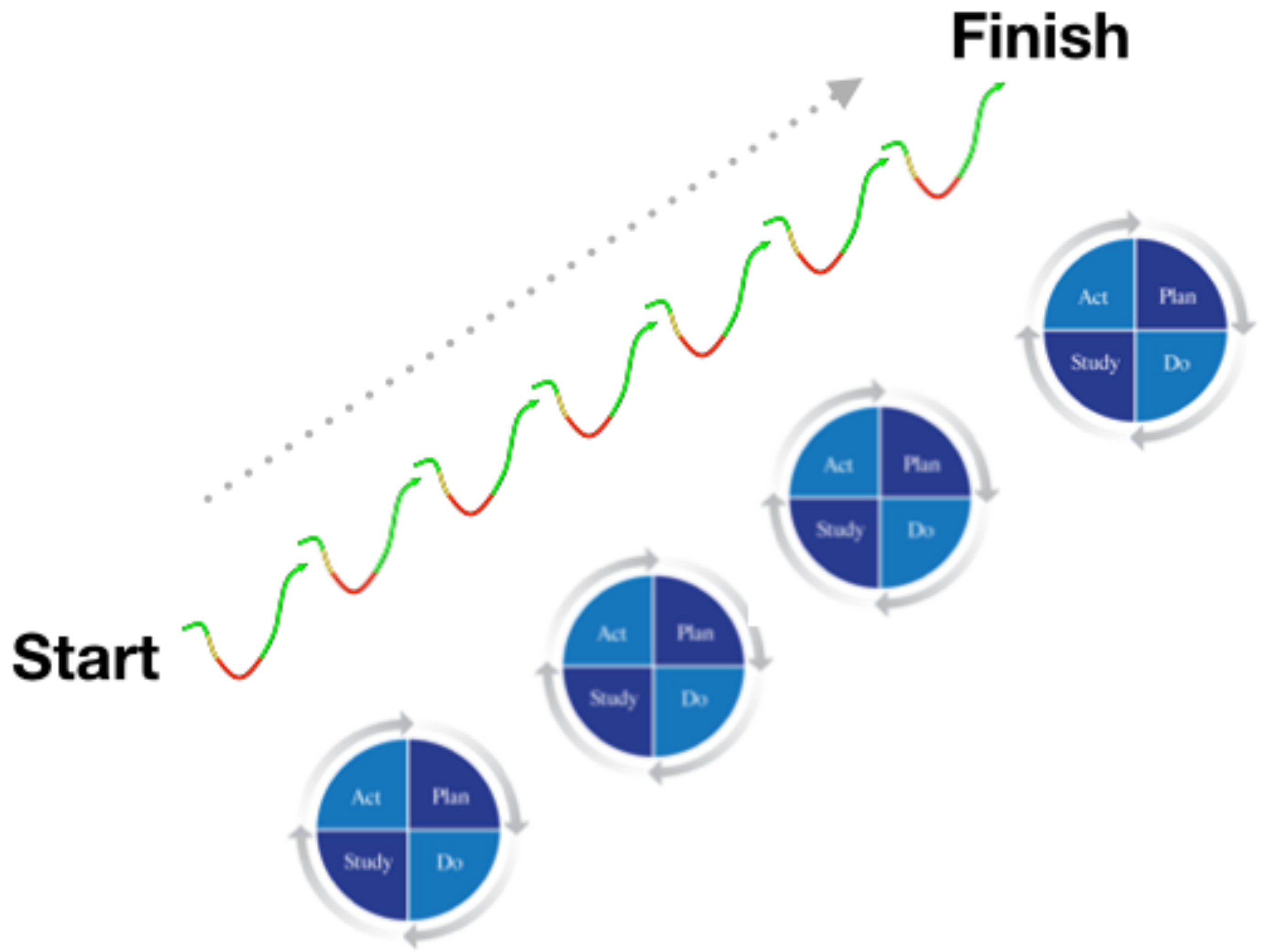
Build into your operating model



Make the work visible



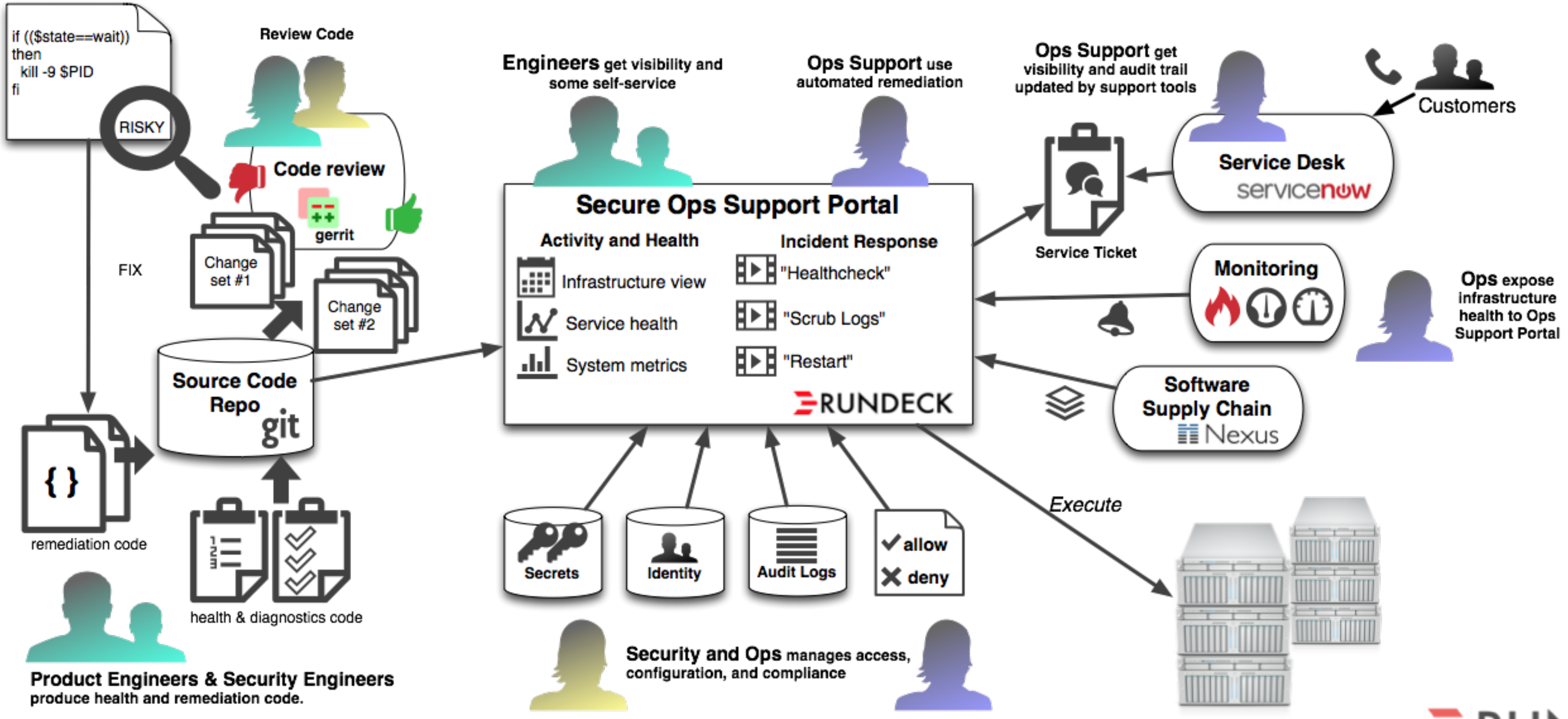
Focus on Continuous Improvement



RUNDECK

Join me tomorrow! 10:15 in Victoria Suite

Helping Ops Help You: Development's Role in Enabling Self-Service Operations


if (($state==wait))
then
kill -9 $PID
fi
RISKY
FIX
Review Code
Code review
gerrit
Change set #1
Change set #2
Source Code Repo
git
remediation code
health & diagnostics code
Product Engineers & Security Engineers produce health and remediation code.
Engineers get visibility and some self-service
Ops Support use automated remediation
Ops Support get visibility and audit trail updated by support tools
Customers
Secure Ops Support Portal
Activity and Health
Infrastructure view
Service health
System metrics
Incident Response
"Healthcheck"
"Scrub Logs"
"Restart"
RUNDECK
Service Ticket
Service Desk
servicenow
Monitoring
Ops expose infrastructure health to Ops Support Portal
Software Supply Chain
Nexus
Execute
Secrets
Identity
Audit Logs
✓ allow
✗ deny
Security and Ops manages access, configuration, and compliance

RUNDECK

Damon Edwards

@damonedwards

damon@rundeck.com

RUNDECK