



# **DevOps Kaizen:**

## **Practical Steps to Start & Sustain an Organization's Transformation**

**Damon Edwards**  
**@damonedwards**



**dto**  
SOLUTIONS

DevOps Consulting  
Operational Improvement

#SimplifyOps

RUN  
DECK

Tools

Symantec.

emdeon

EQUIFAX

salesforce

GUARDIAN

apple

at&t

omgeo

Adobe

barclaycard

ticketmaster®  
intuit.

TD

WILLIAMS-SONOMA

CapitalOne

zynga

U.S. Citizenship  
and Immigration  
Services

LOYAL3

D&B

RESTORATION  
HARDWARE

bwin.party  
digital  
entertainment

R M S

LinkedIn

BT

CCP

vmware

digby

Cisco  
webex

TransUnion.

RueLala

backcountry

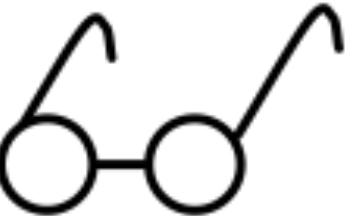
ING

Nextdoor

eTRADE  
FINANCIAL

Fidelity  
INVESTMENTS

Gene Kim



- - - - →  
← - - - - - Practices & Behaviors



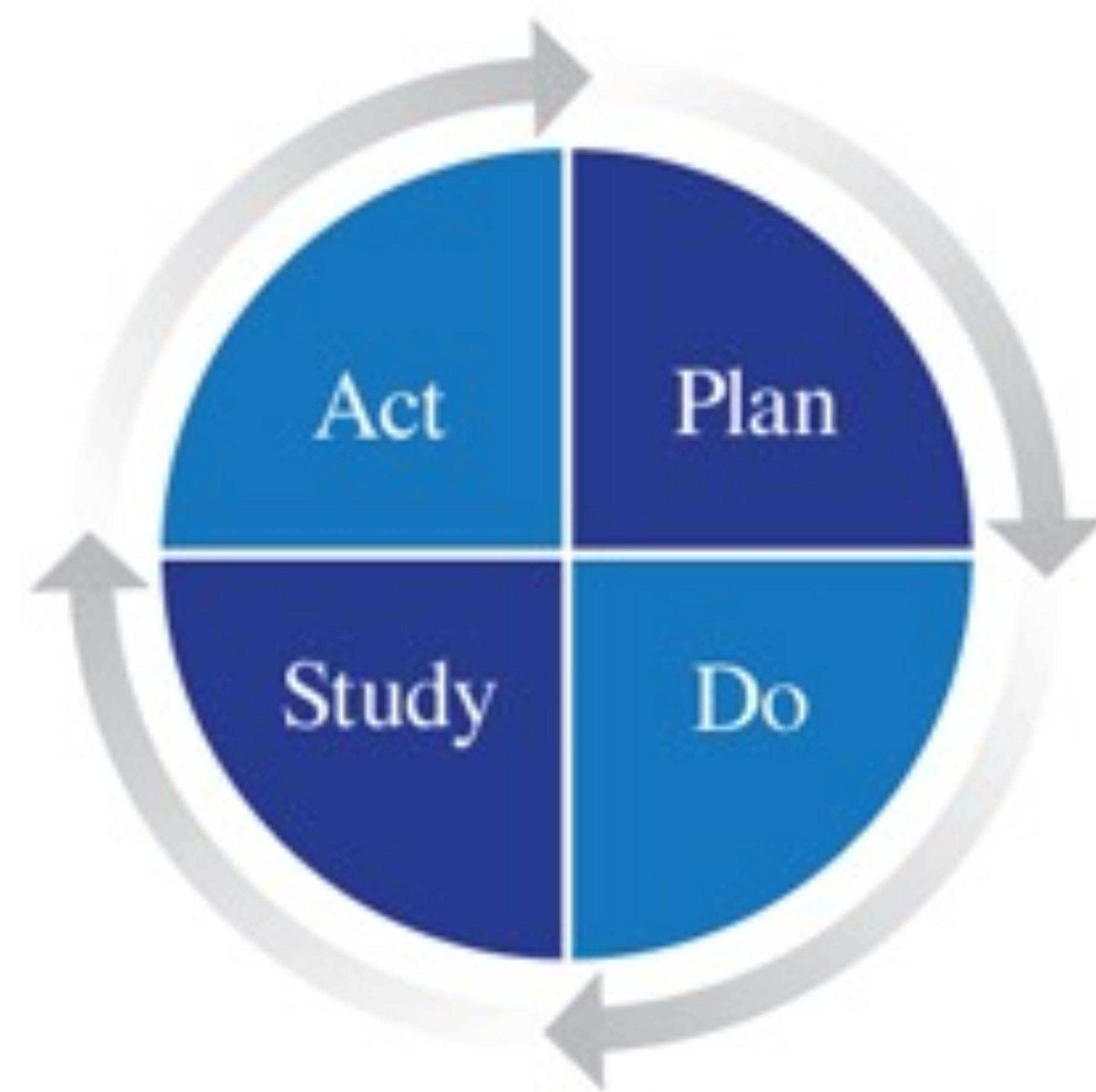
... but WHY are  
they different?



**The ability to improve.**

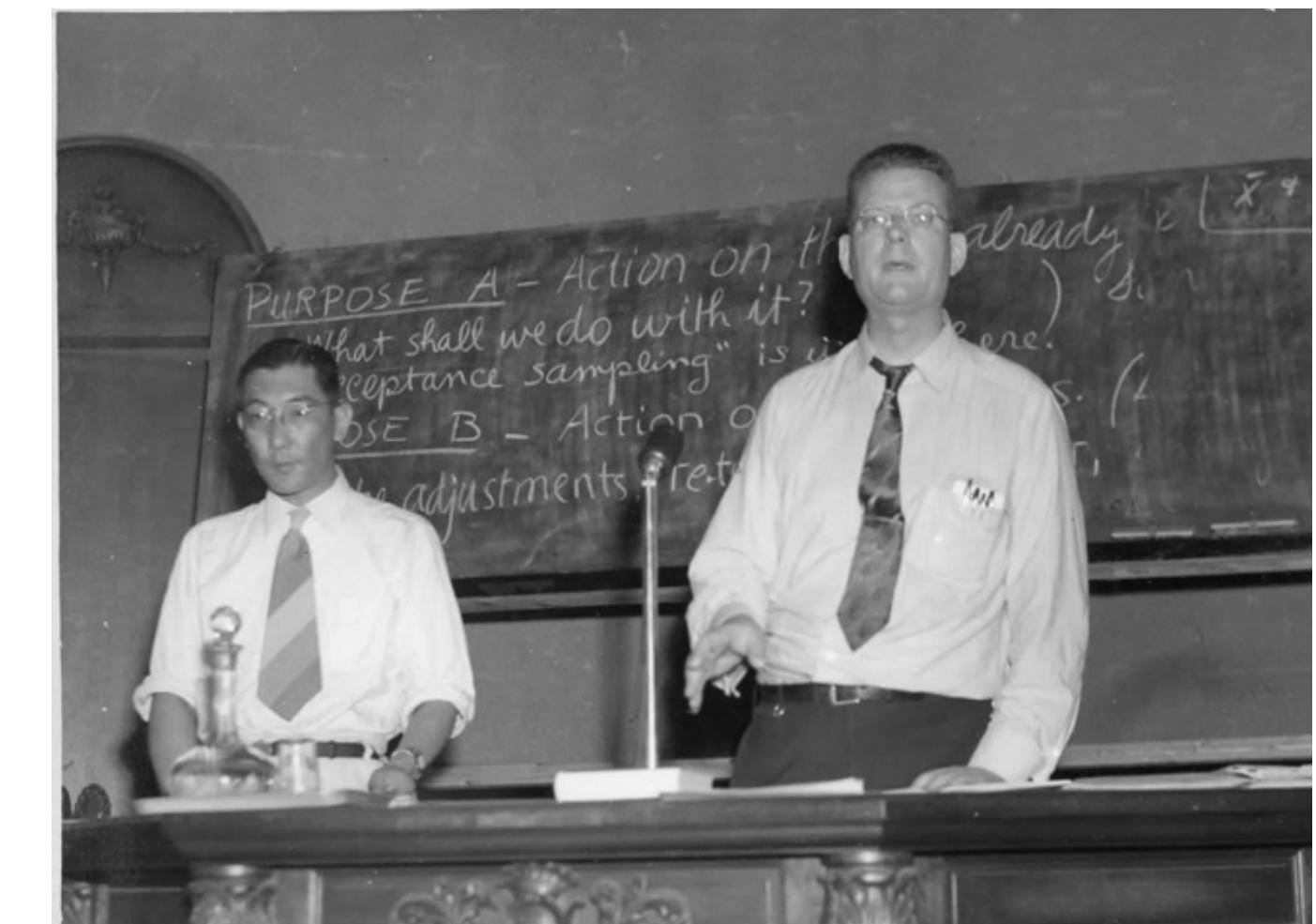
**The unique trait of high-performing companies is  
that they are good at learning fast.**

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



*Other variants:*  
PDCA  
OODA

**W. Edwards Deming - 1950**



# 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**

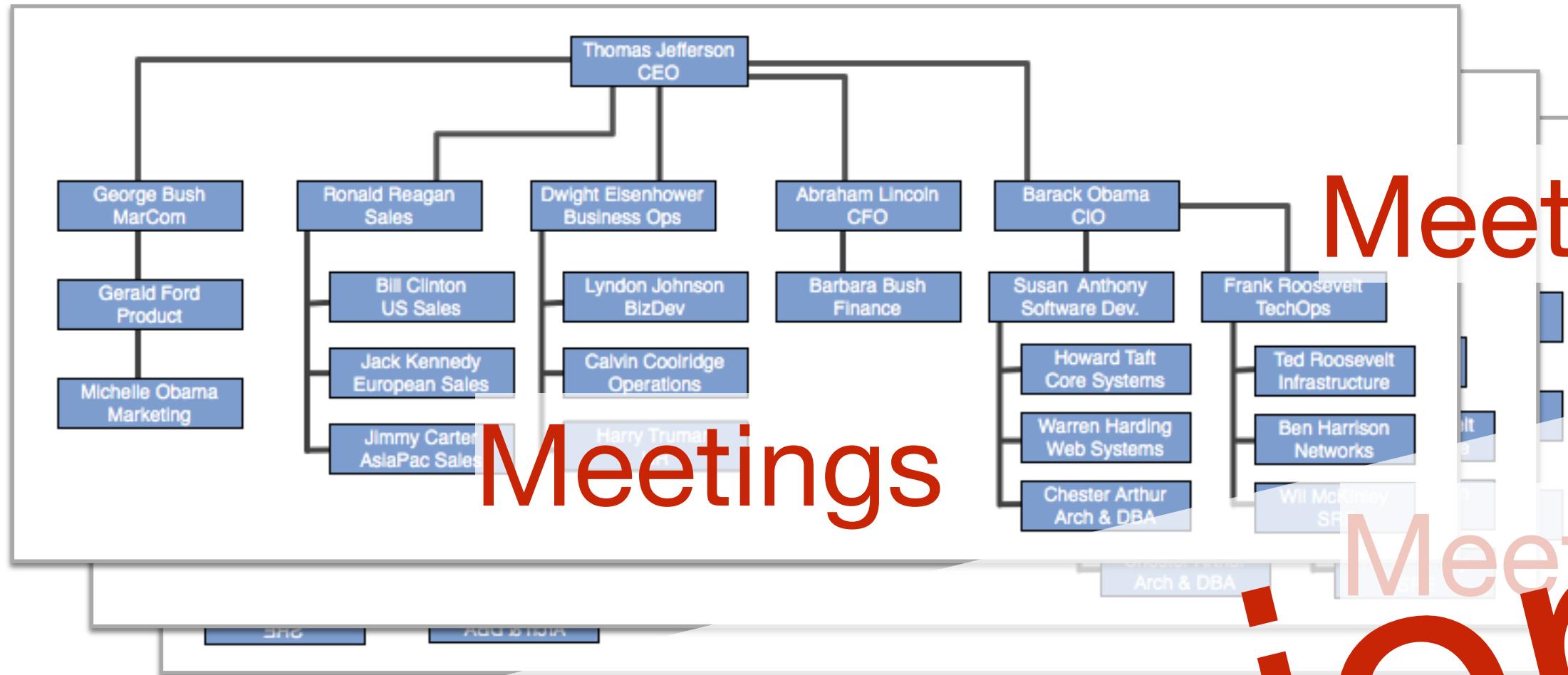


**(Silo Effects)**



# Traditional “Visibility” For Technology Managers

# Org Charts



# Meetings

# Meetings

# Proj

Finance	Project Gamma	31-Jun-09	26-Jun-01
	Upgrade Windows 95	10-Dec-09	27-Dec-99
IT	<b>Windows 2000 Project</b>	<b>08-May-01</b>	<b>07-Dec-01</b>
IT	Windows Windows 2000 (R&D)	50-Jun-01	03-Jun-00
IT	Windows 2000 Management Strategy (Develop)	24-Apr-00	21-Jul-00
IT	Windows 2000 Management Strategy (Implement)	24-Jul-00	20-Oct-00
IT	Windows 2000 Management Strategy (Test)	25-Aug-00	15-Dec-00
IT	<b>Windows 2000 Management Strategy (Deploy)</b>	<b>10-Dec-00</b>	<b>07-Dec-01</b>
IT	Windows 2000 Upgrade for Desktops	05-Feb-01	03-Oct-01
IT	Windows 2000 Upgrade for Laptops	15-Mar-01	03-Nov-01
IT	Windows 2000 Upgrade for Servers	16-Dec-00	23-Mar-01
IT	Windows 2000 Complete	07-Dec-01	07-Dec-01
Finance	<b>Mega Project B</b>	<b>31-Feb-08</b>	<b>24-Nov-00</b>
Finance	Project B.1	20	21-Feb-00

# gs

# Meetings Control!

# Meetings

# Meetings

# Meetings

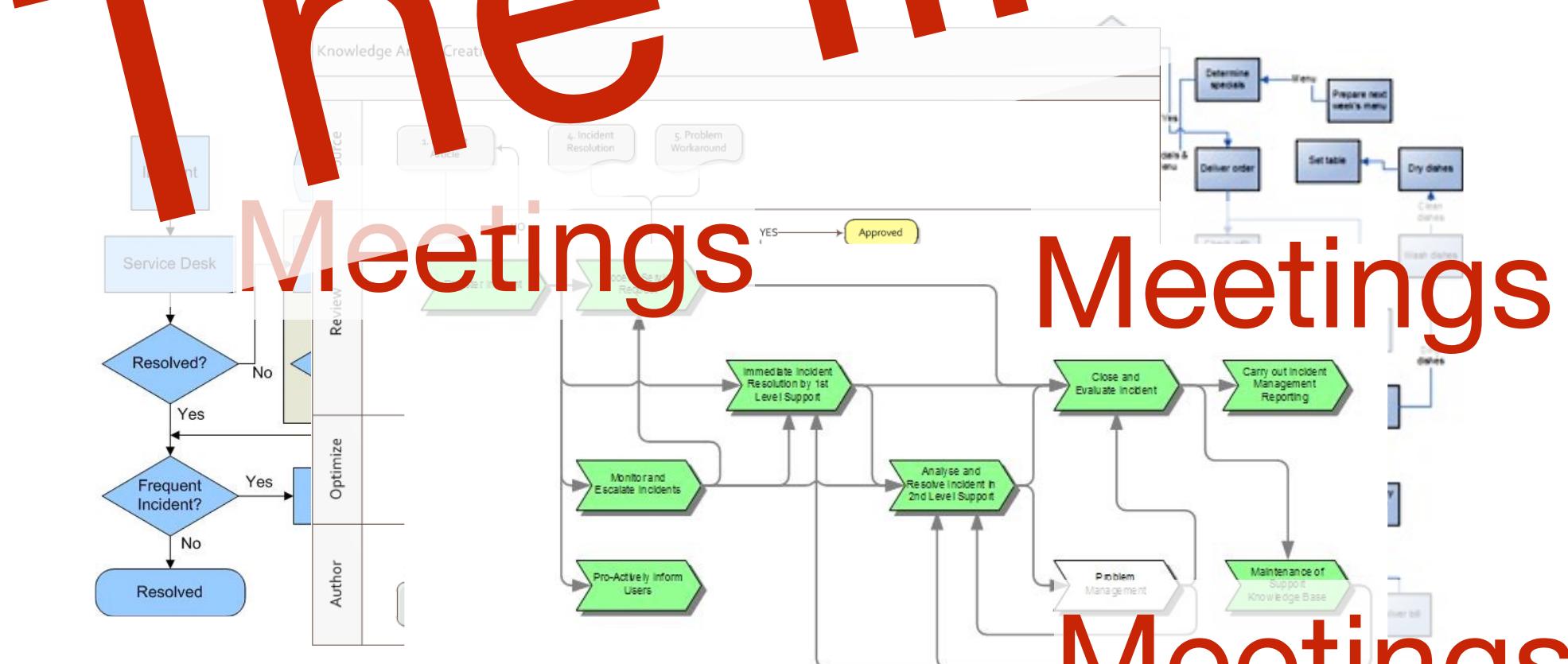
# Re|ease Trains

12.2(50)SQ Maintenance Train

3.2.1SQ 3.2.2SQ

12.2(50) 12.2(50) 12.2(50) 12.2(50) 12.2(50)

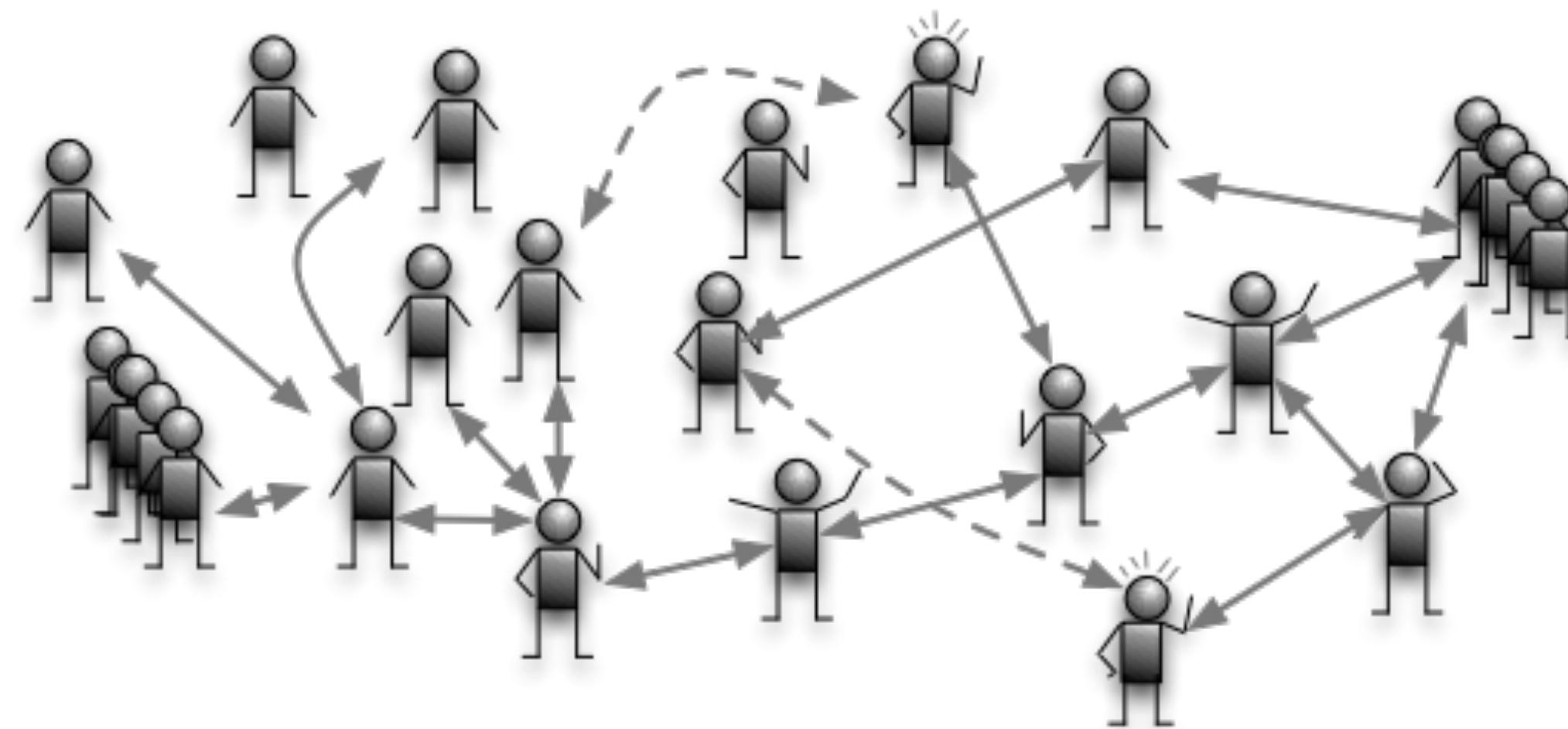
SO3 SO3 SO4 SO5



# It's a complex system

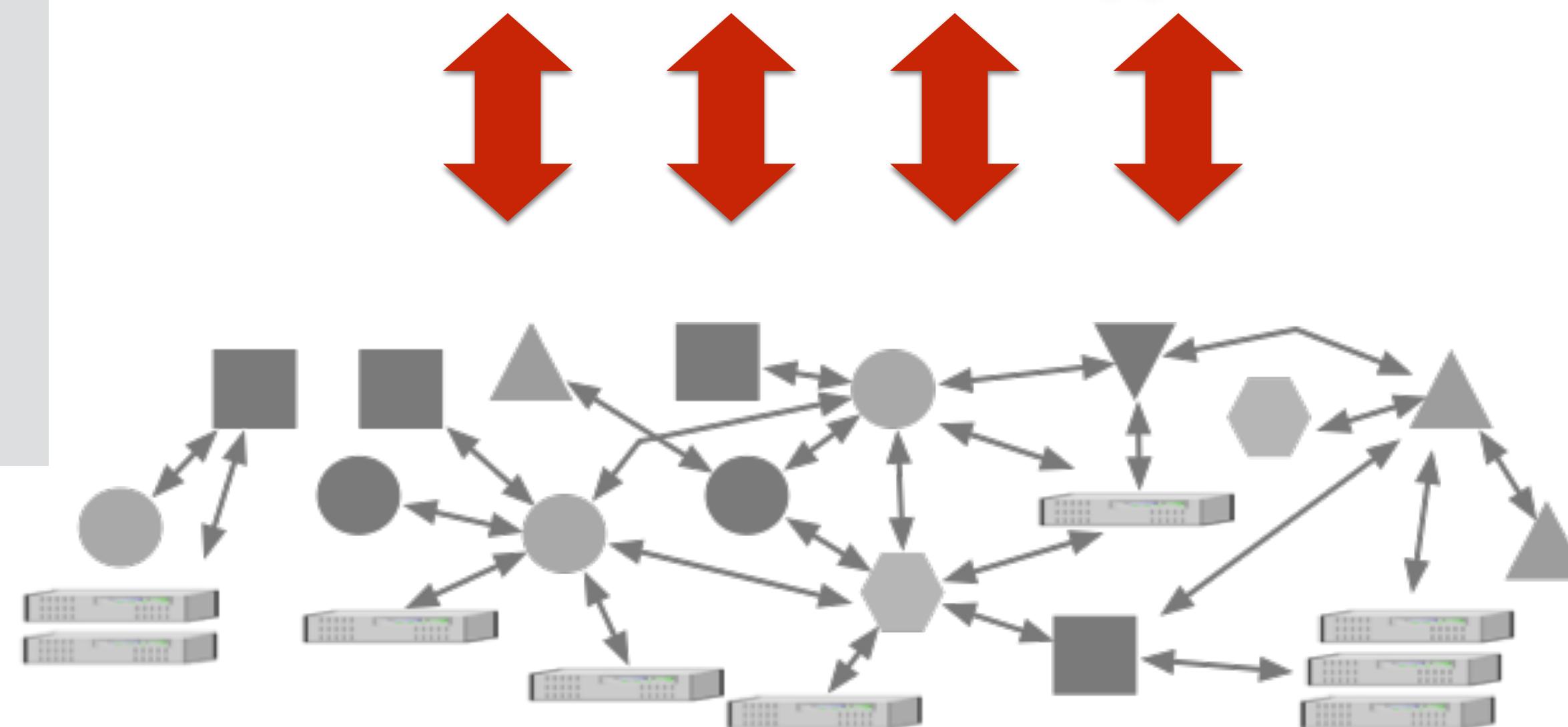
No wonder...

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



Complex System

interacting with



Complex System

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

***"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!

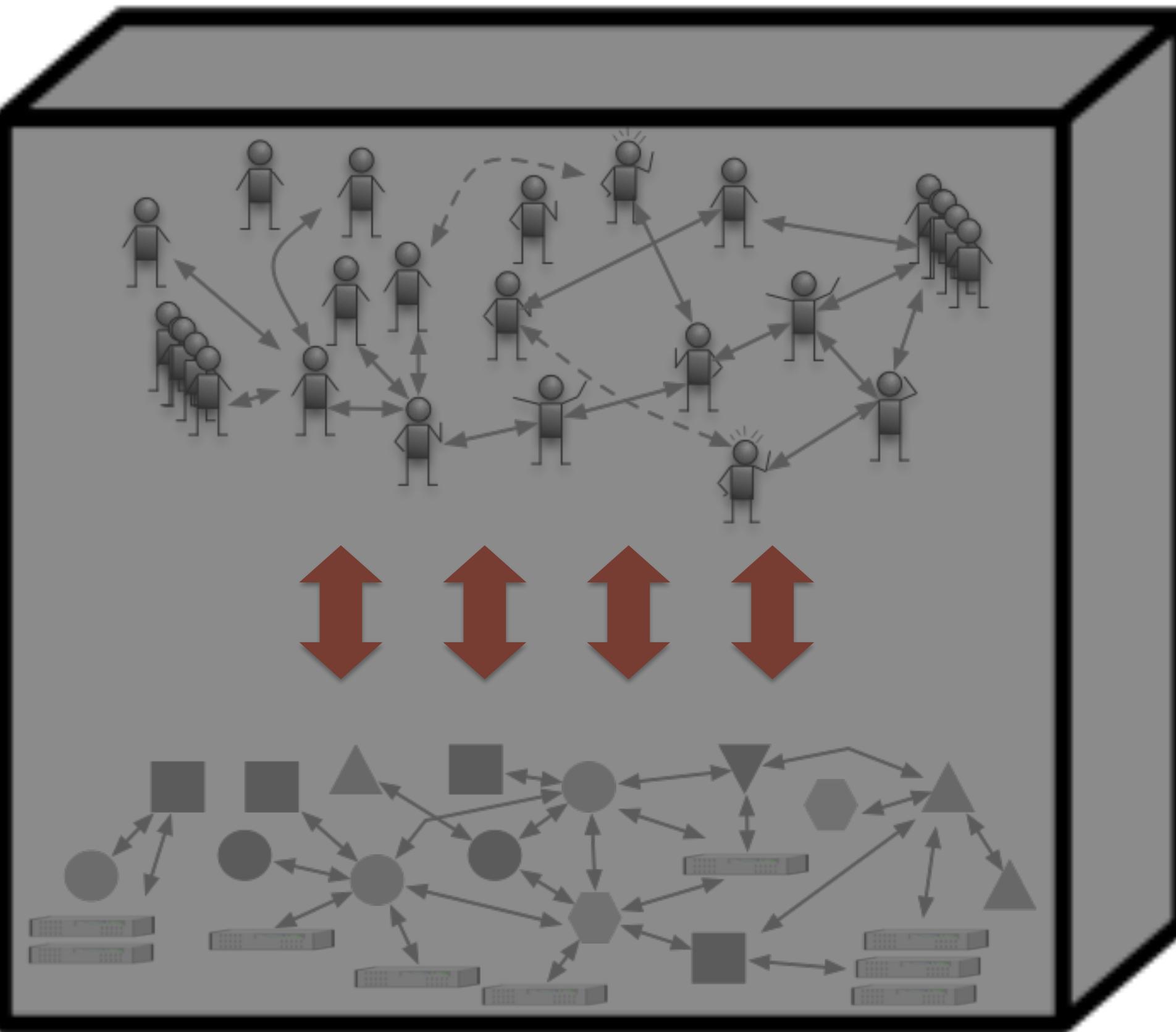


Executive  
Committee

Need better tools...  
new automation and a  
new network!

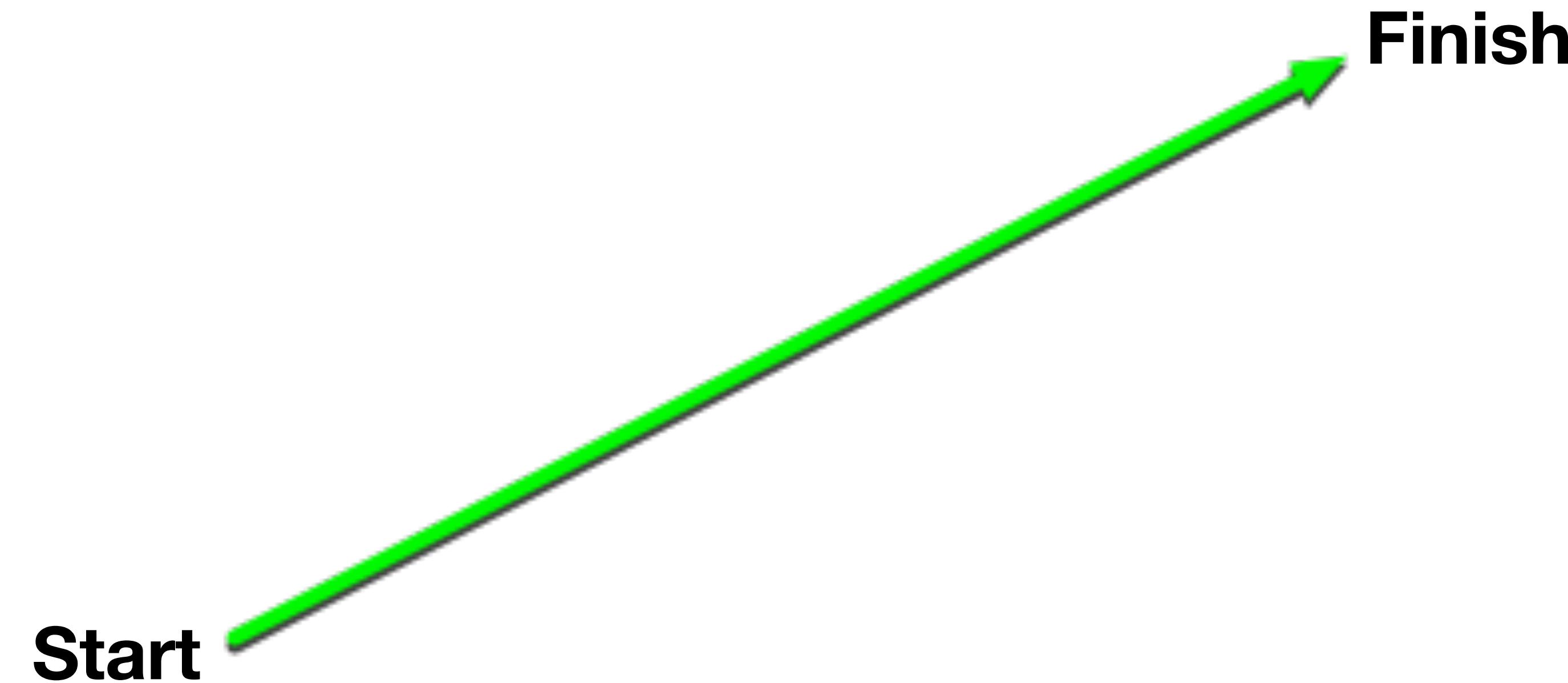


Engineers



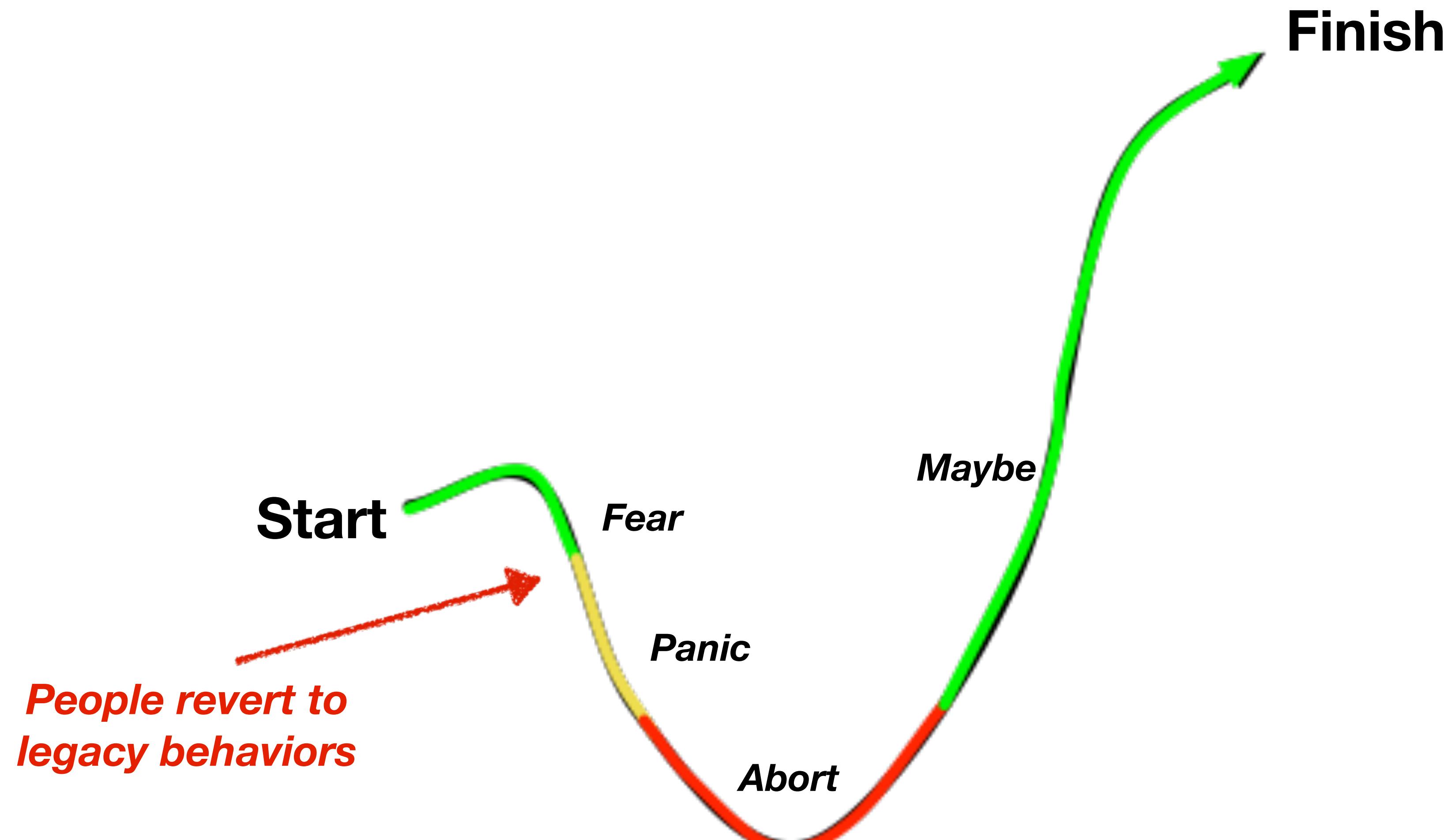
# The “Big Bang” Transformation Dream

---



# The “Big Bang” Transformation Reality

---



Too costly...  
outsource more!



Finance

Need Results...  
Re-Org!



Executive Committee

More discipline...  
tighter process and  
more approvals



Change  
Management

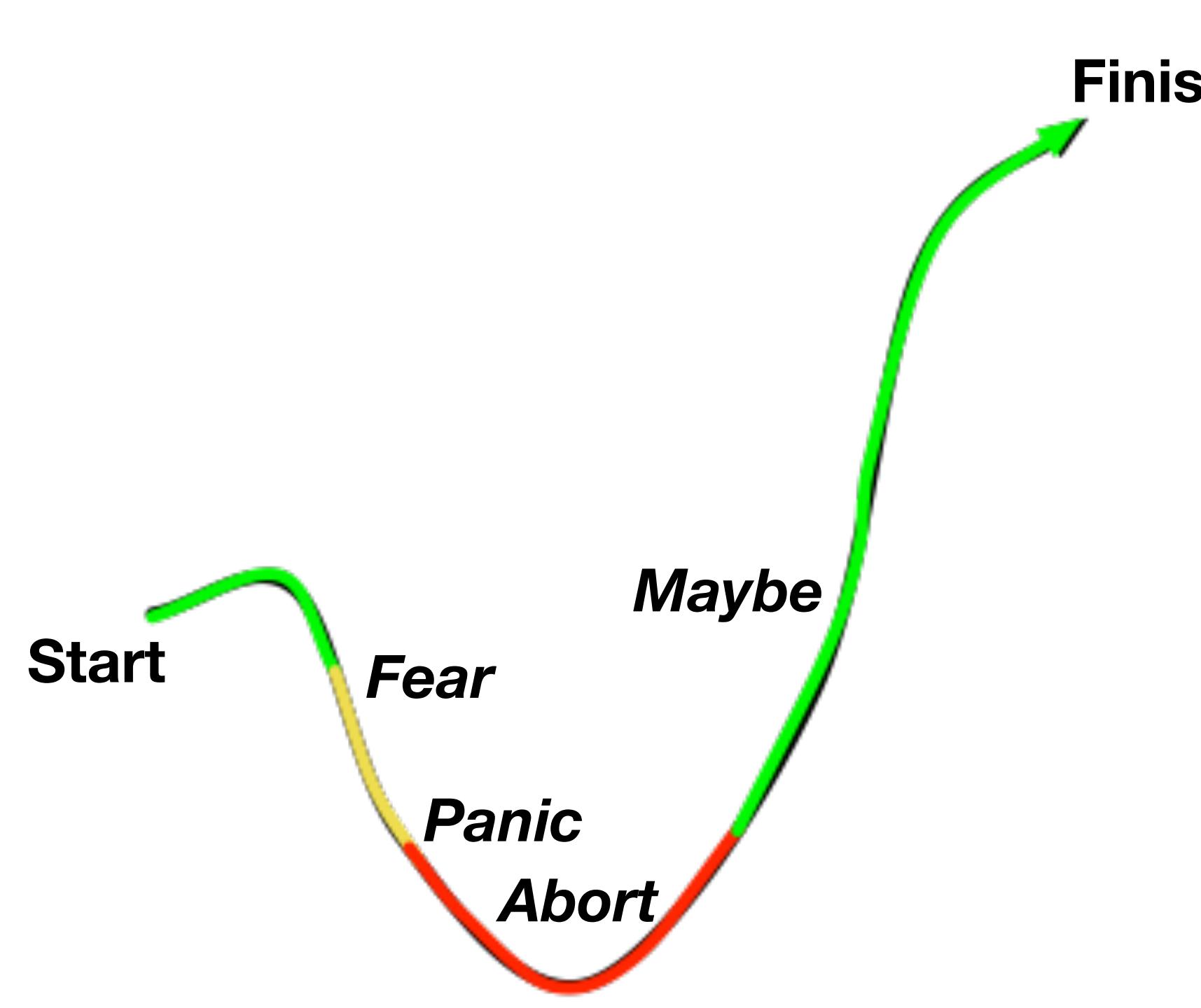
Need better tools...  
cool automation and a  
new network!!



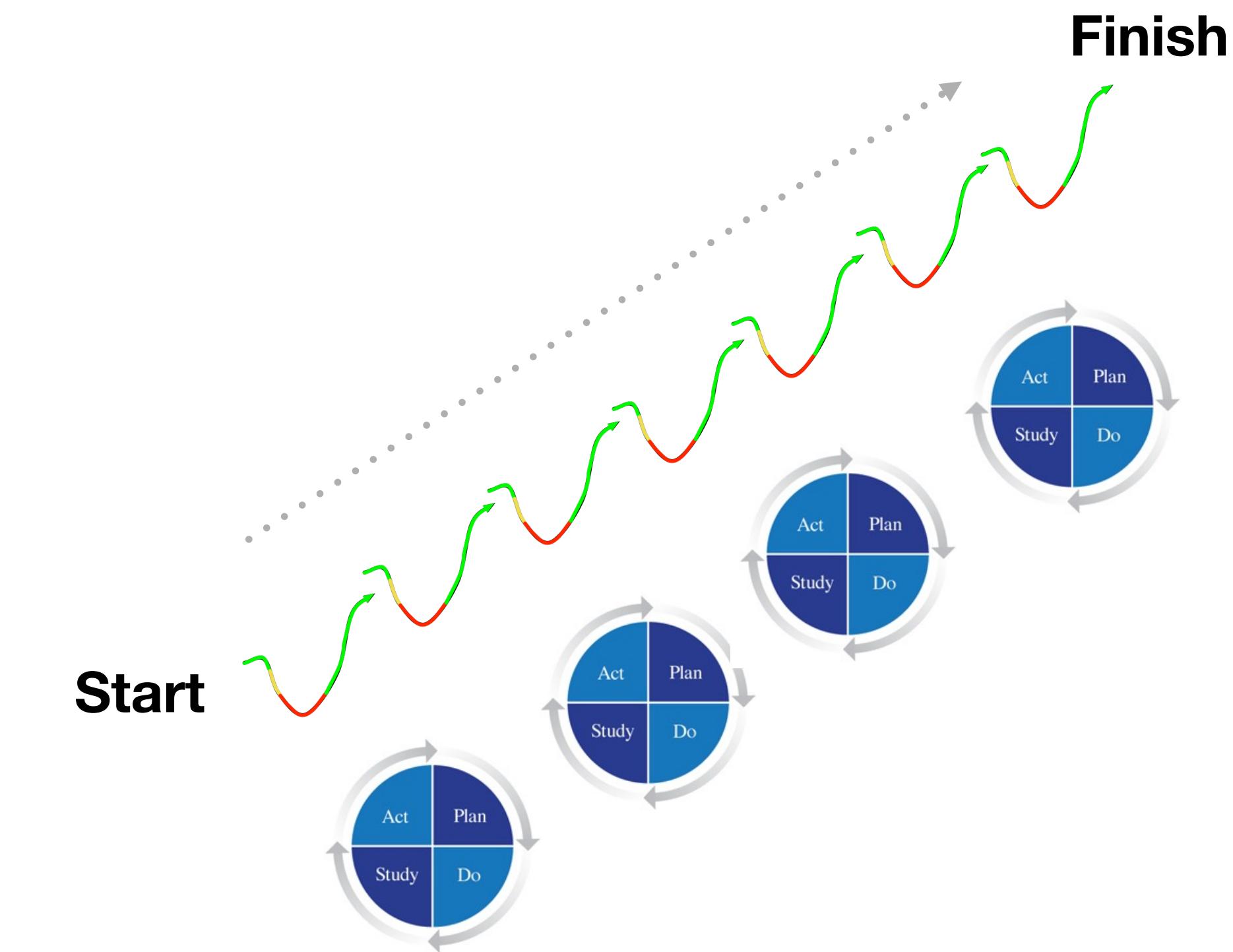
Engineers



# How to teach an organization to fix itself: “Little J’s” instead of “Big J”



**“Big Bang”**



**Continuous Improvement**

# **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**

# **Remember what you are up against....**

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

# **“DevOps 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)**
- **DevOps context:**
  - **Continuously improve the flow of work through the full value stream in order to improve customer outcomes**

# **“DevOps Kaizen”**

---

**Proven Lean Techniques**

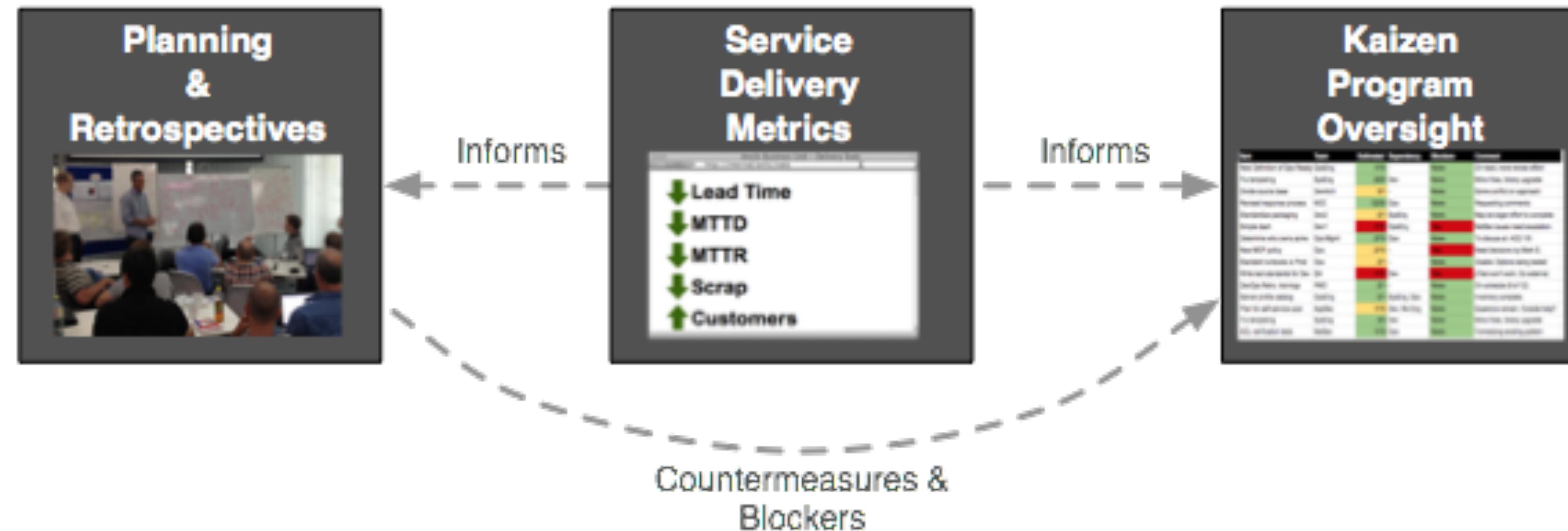
**+**

**DevOps Context**

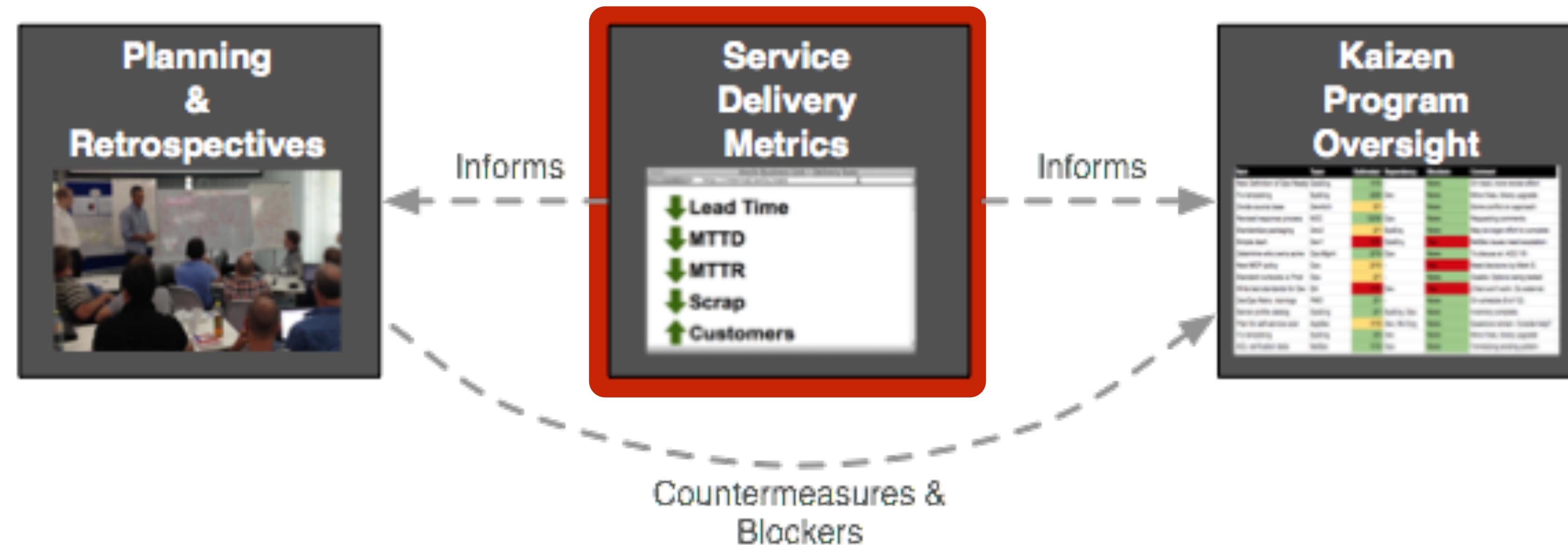
*“If I have seen further, it is by standing on the shoulders of giants.”*

*-Sir Isaac Newton*

# Elements of a DevOps Kaizen Program

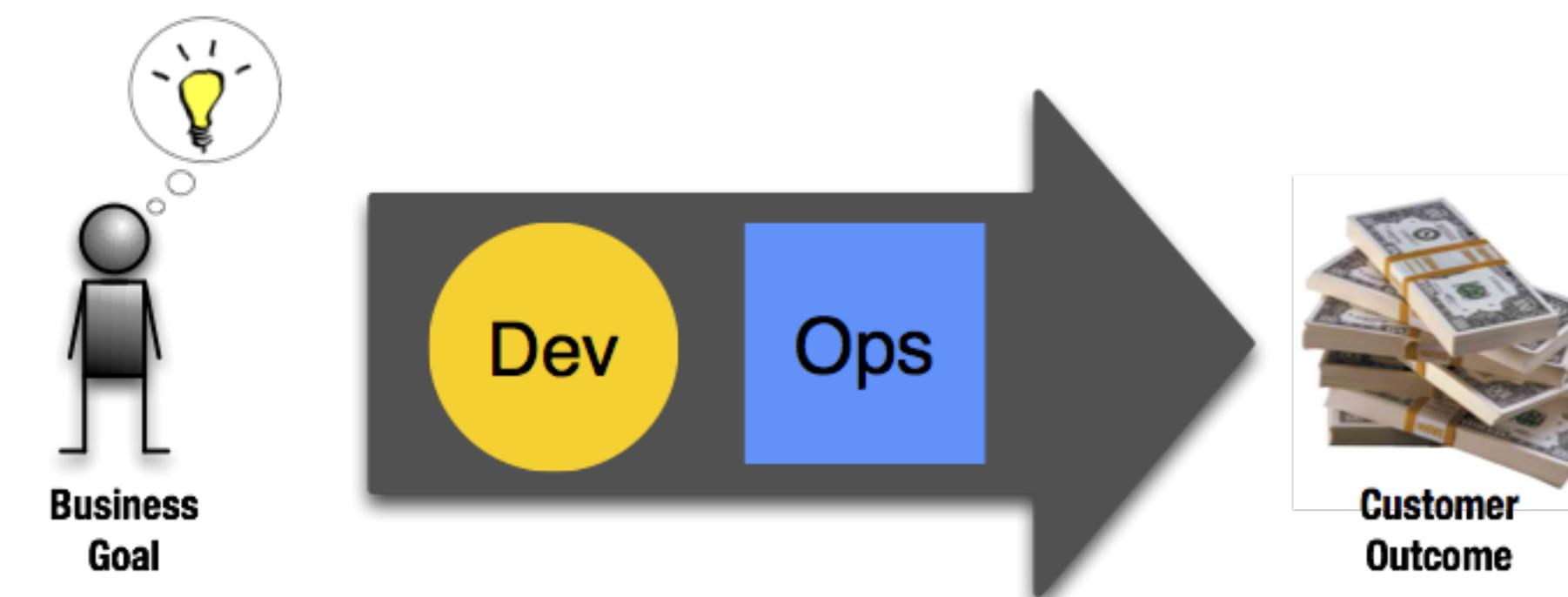


# Elements of a DevOps Kaizen Program

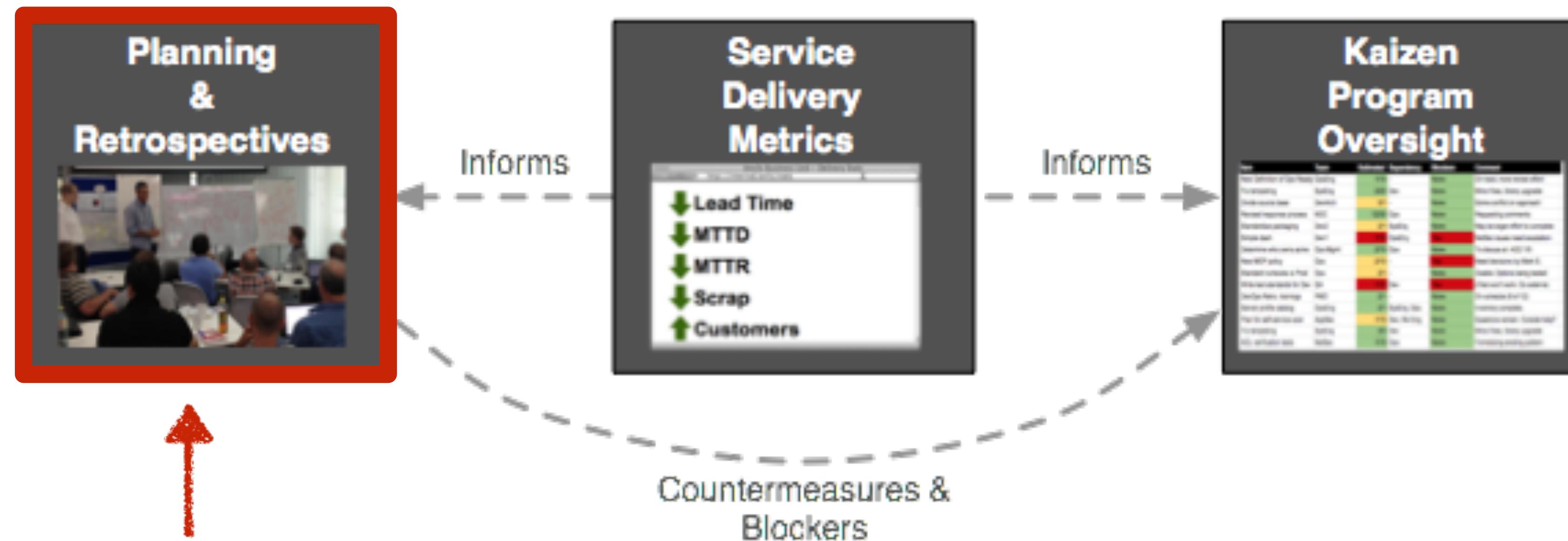


# Organization-wide focus on service delivery metrics

- **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)

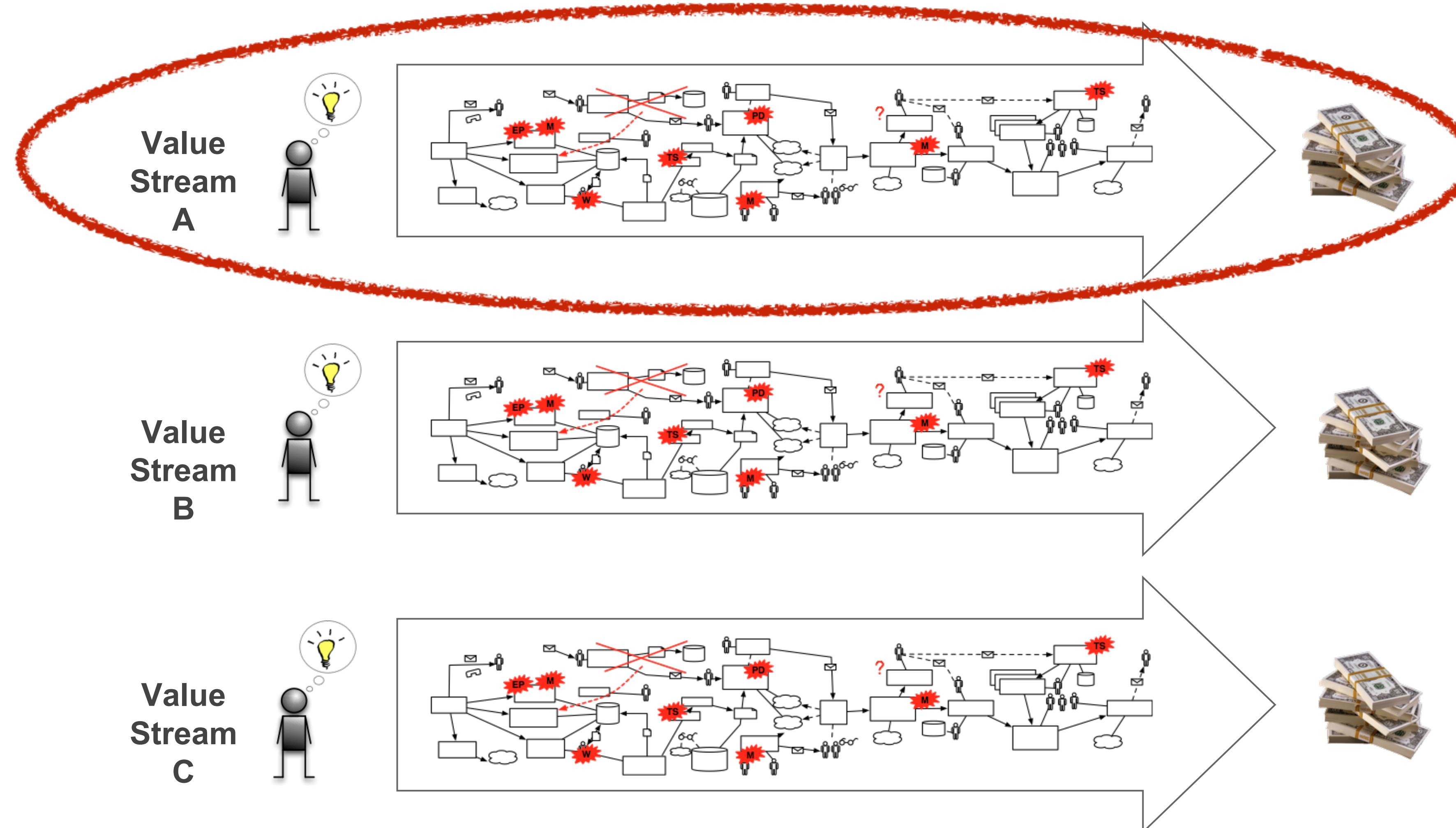


# Elements of a DevOps Kaizen Program



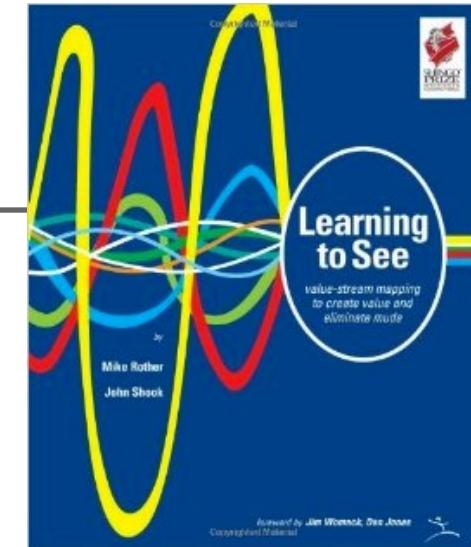
This is where the work  
becomes visible!

# Retrospectives are a per value stream tool



Key: “horizontal thinking”

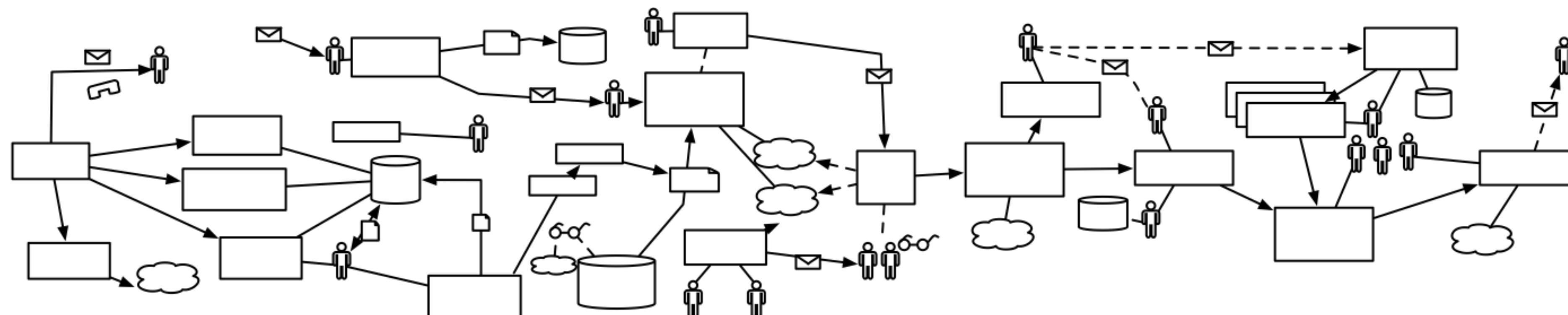
# DevOps Kaizen: Retrospective Technique



1

## Map end-to-end process

Inspiration: value stream mapping



Note: “go to the gemba”  
requires making it  
visible together

Include key process metrics:  
**Lead Time**  
**Processing Time**  
**Scrap Rate**  
**Head Count**

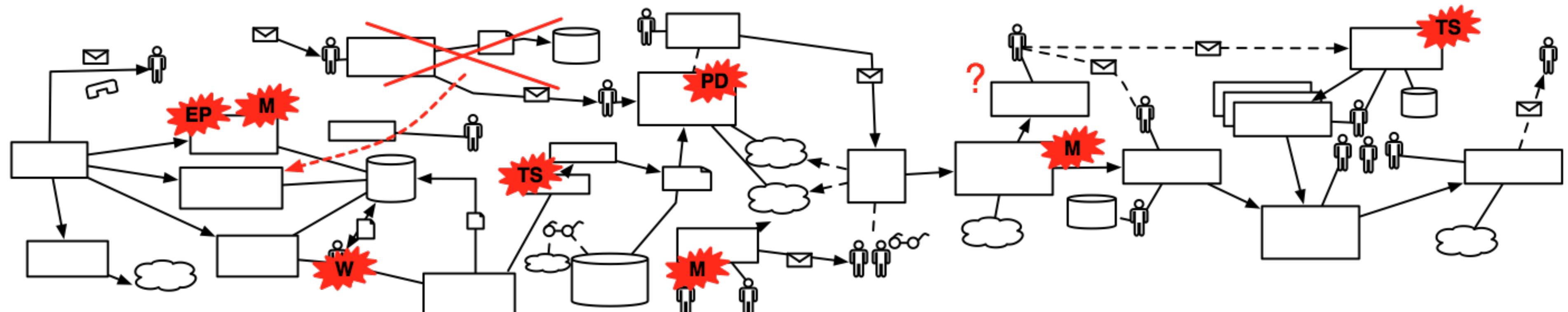
Key: graphical  
facilitation above all  
else!

# DevOps Kaizen: Retrospective Technique

Inspiration: 7 Wastes of Software Development

2

Identify wastes, inefficiencies, bottlenecks



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

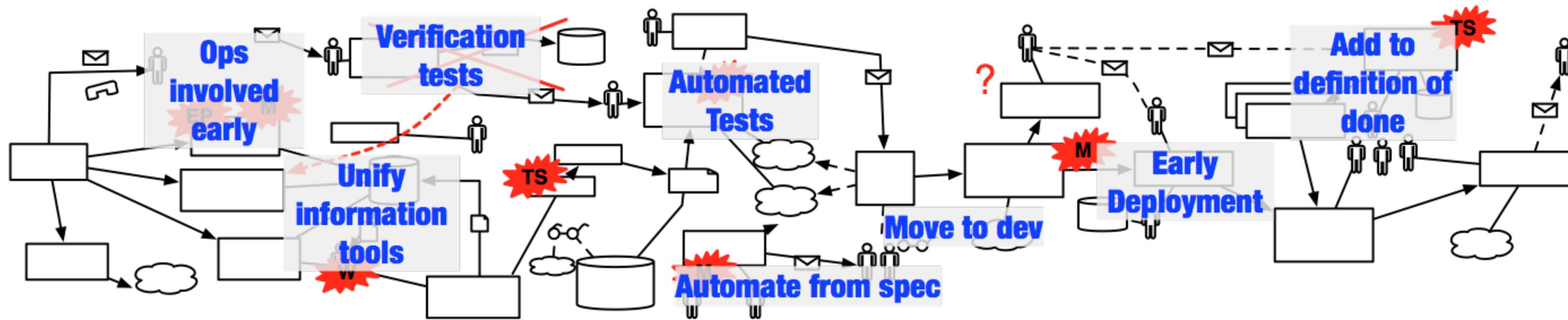
<b>PD - Partially Done</b>	<b>D - Defects</b>
<b>TS - Task Switching</b>	<b>EP - Extra Process</b>
<b>W - Waiting</b>	<b>EF - Extra Features</b>
<b>M - Motion / Manual</b>	<b>HB - Heroics</b>

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

# DevOps Kaizen: Retrospective Technique

3

## Identify countermeasures



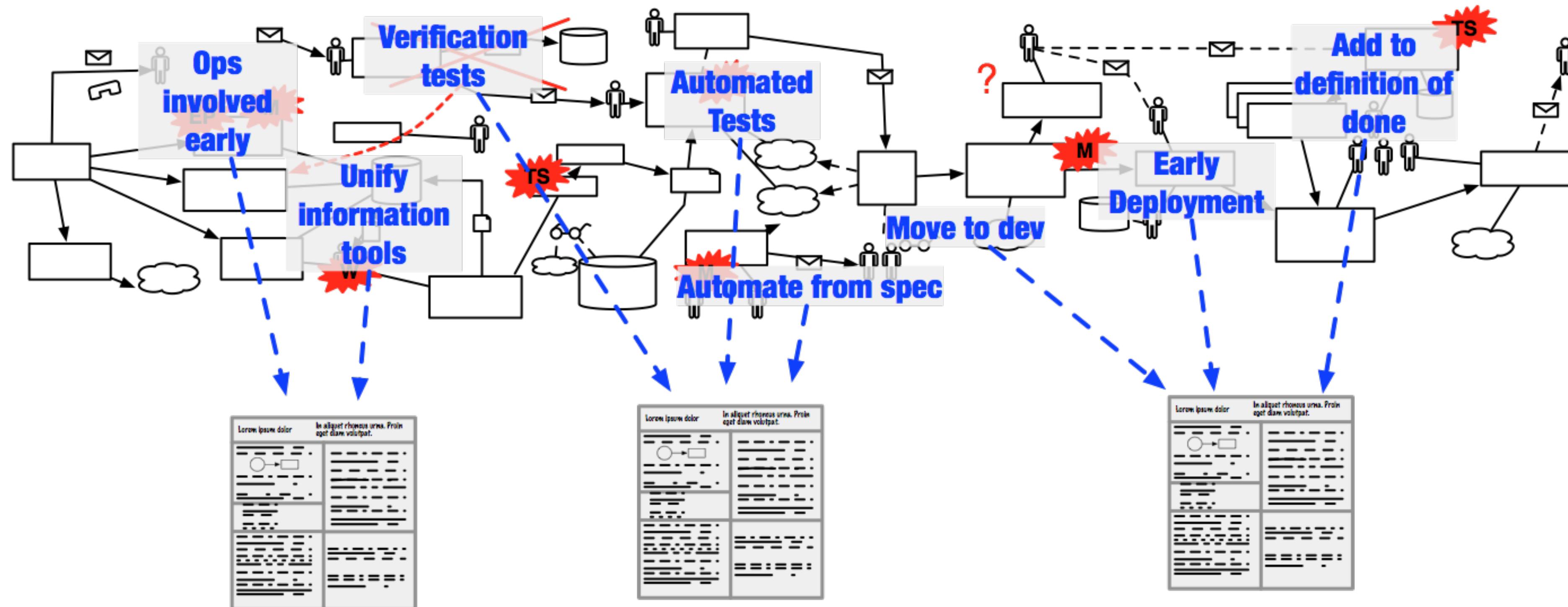
Key: “small j’s,  
not big j’s”

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)

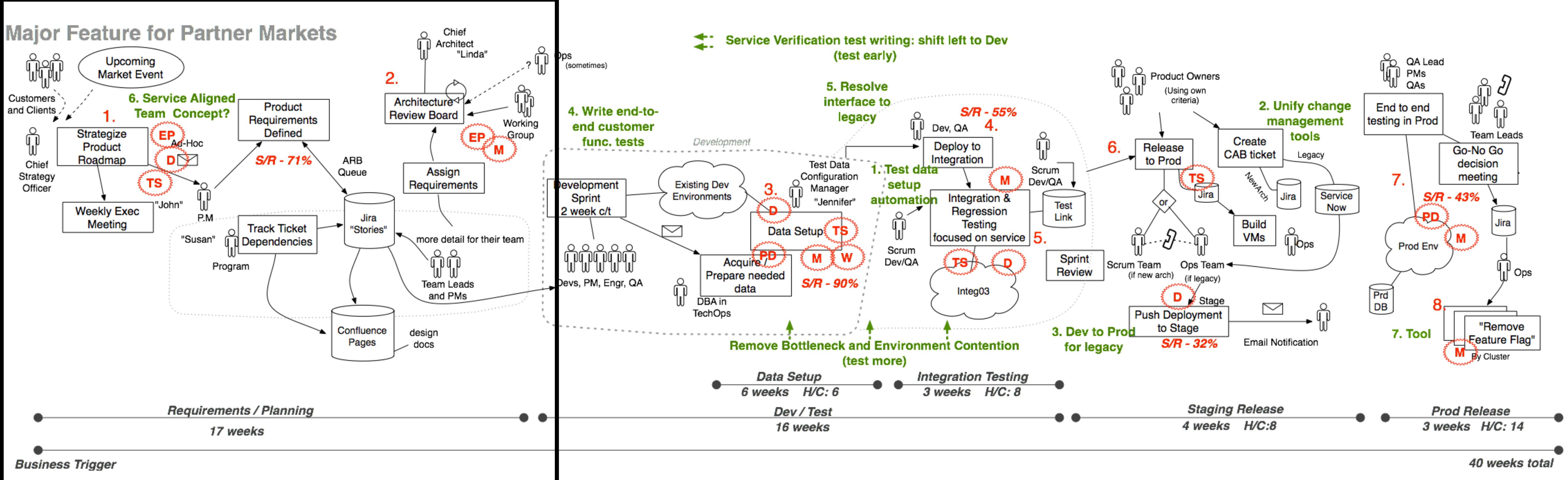


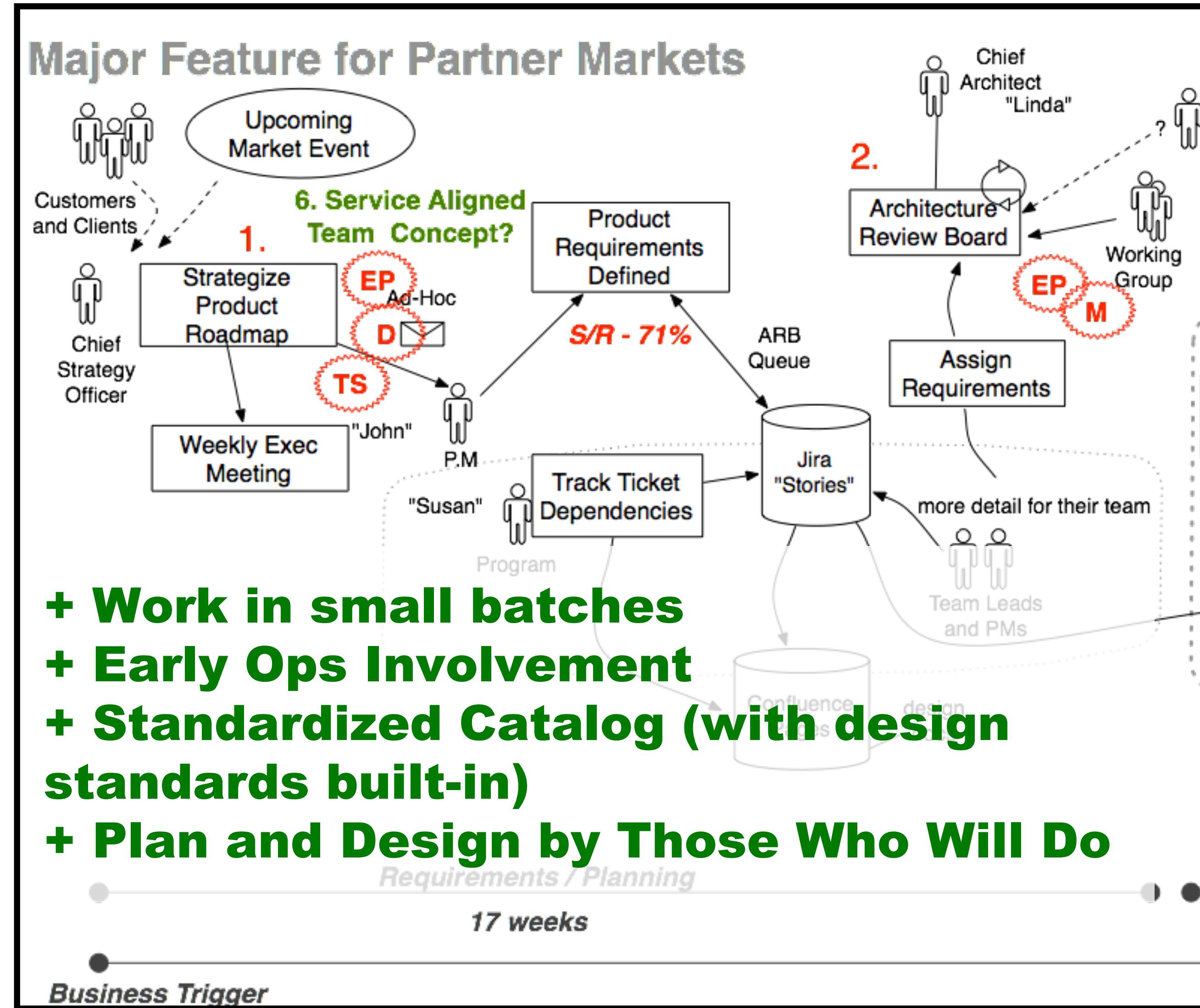
Key: actionable short-term “baby steps”... “what are we going to do next?”

# **DevOps Kaizen: Retrospective Technique**

---

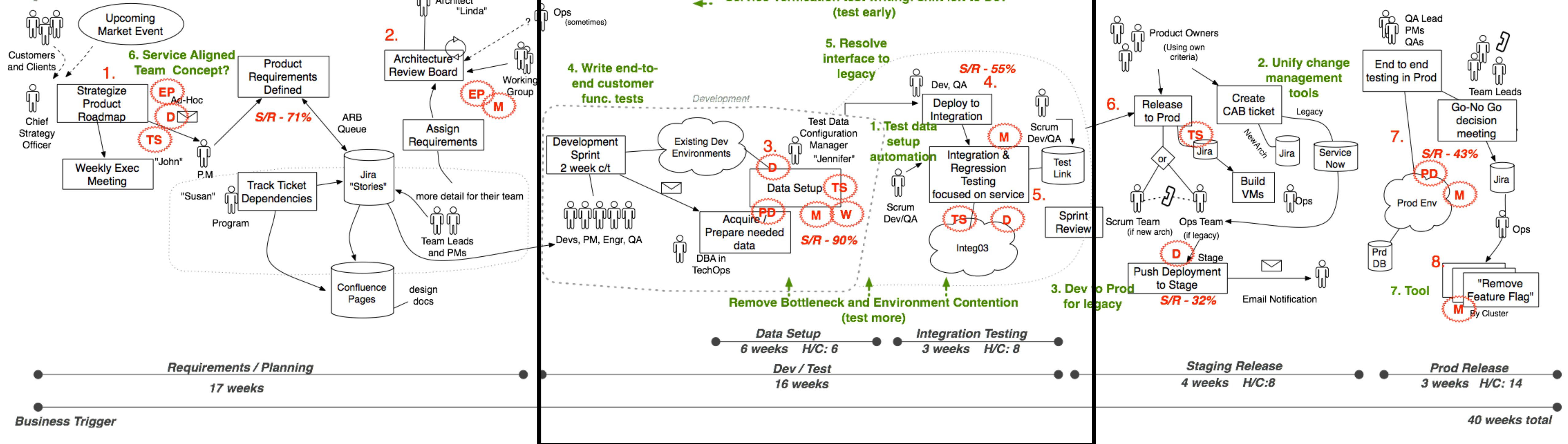


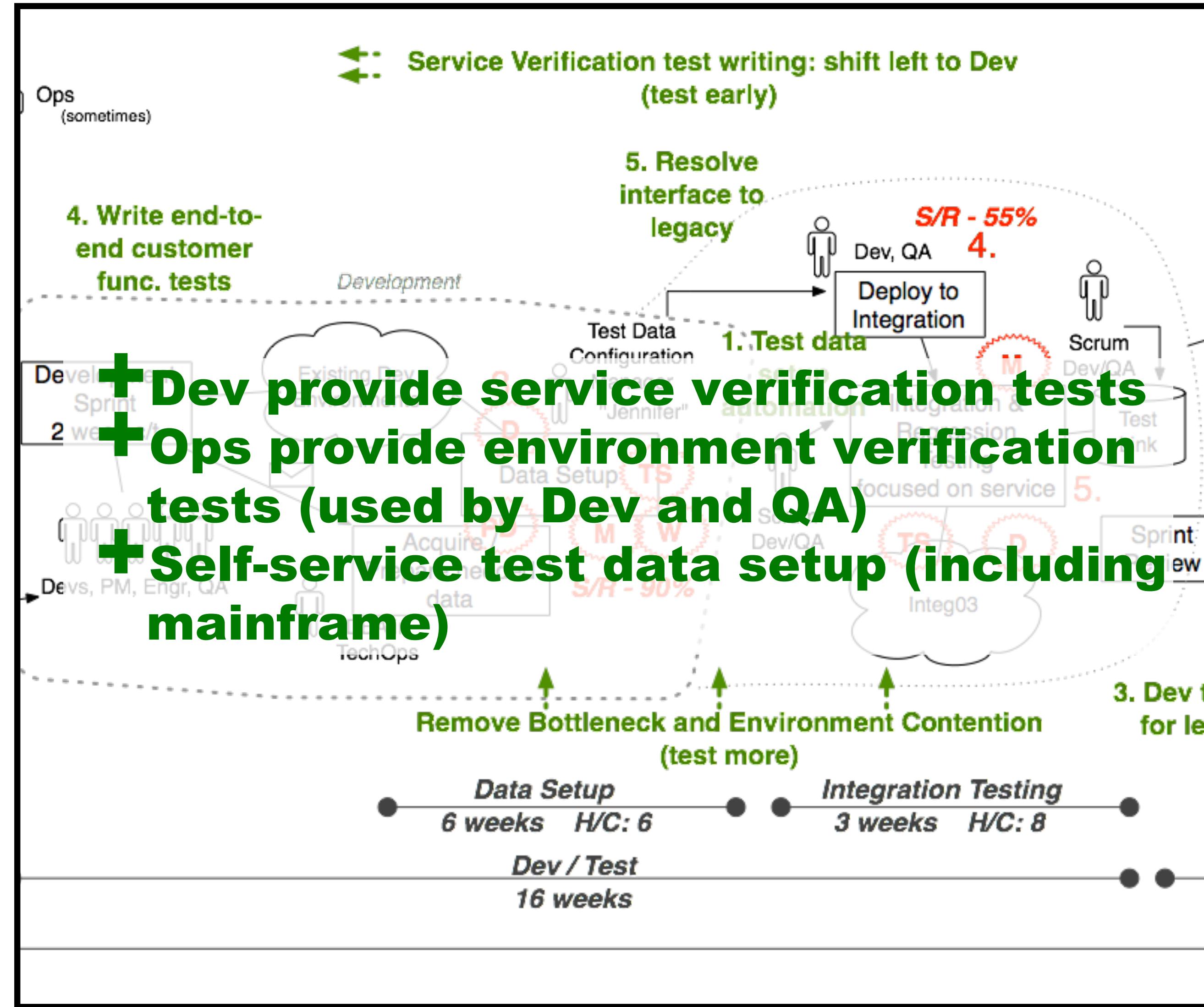




Key: "What can we do next?" NOT "what is nirvana?"

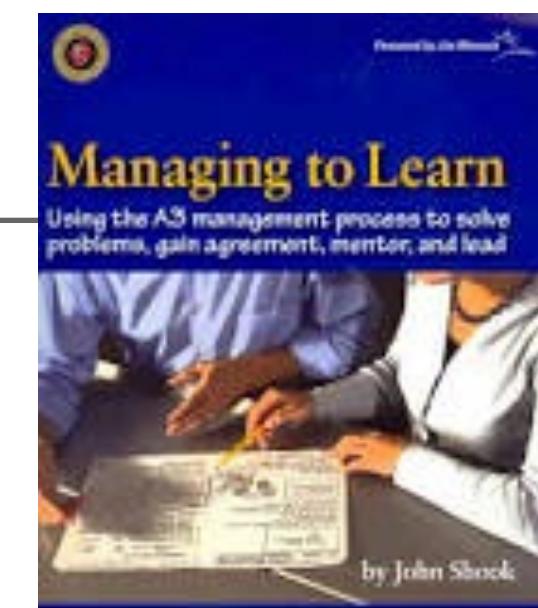
## Major Feature for Partner Markets





# Improvement Storyboards

Inspiration: A3 management process



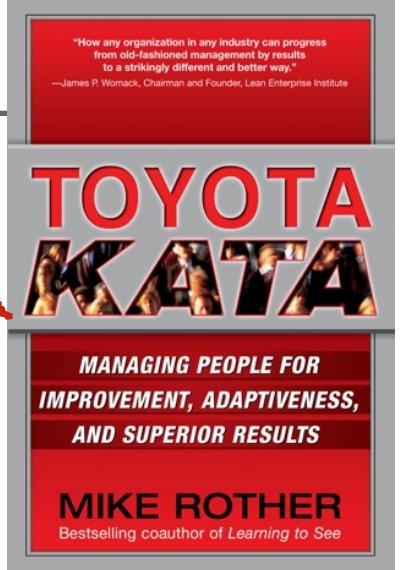
Template

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

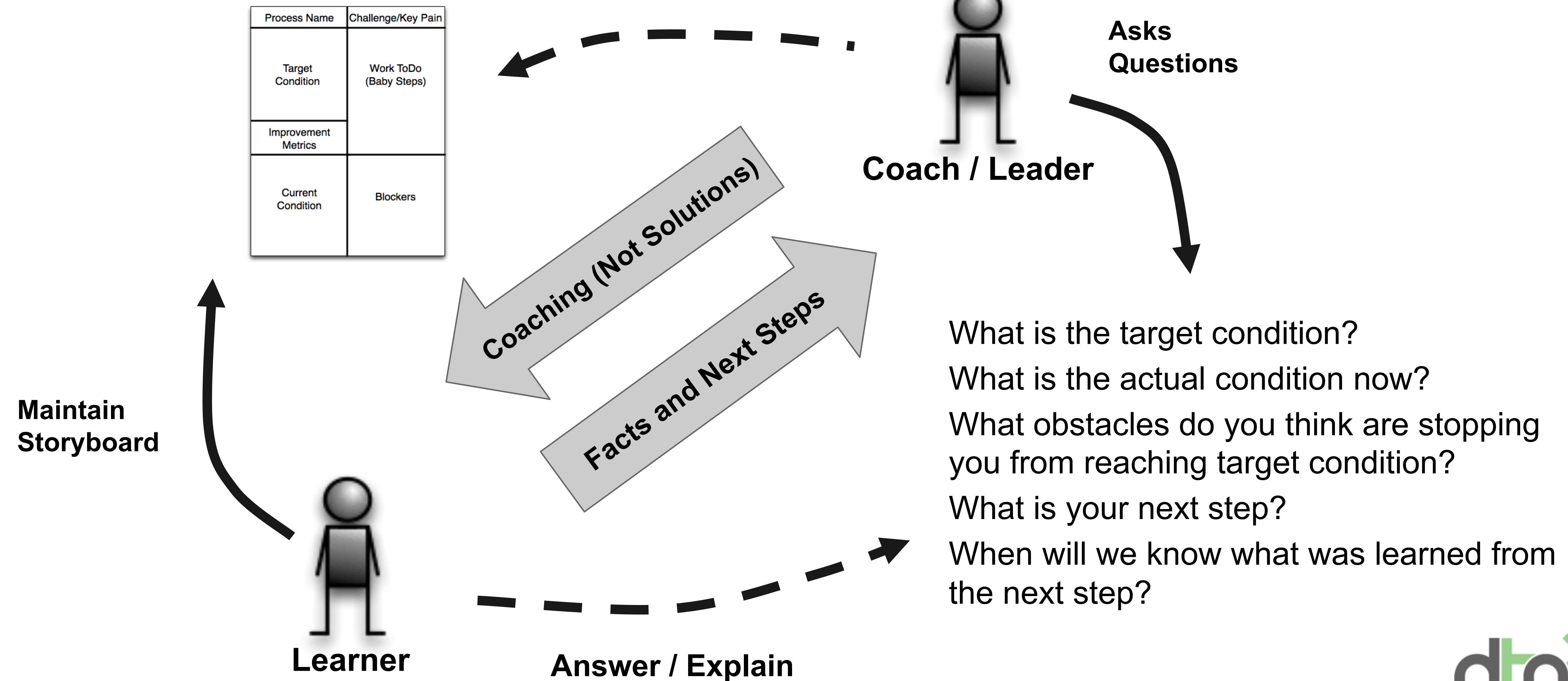
Example

<b>Process</b> GTM/LTM (Traffic manager configuration process)	<b>Challenge/Key Pain</b> Changes are being introduced / tested in production the first time causing delays, rework, outages
<b>Target Condition</b> <ul style="list-style-type: none"> <li>GM/TLM functionality across all SDLC environments (capex request needed)</li> <li>change window reduction for non-prod environments (turn those around instantaneously less than 13 days)</li> <li>Provide read-only to all F5 consoles</li> <li>Standardize GM pattern</li> </ul>	<b>Work ToDo (Baby Steps)</b> <ul style="list-style-type: none"> <li>Acquire the F5 hardware or software to support envs throughout SDLC</li> <li>Make these changes L3 or 5 change requests</li> <li>Write automation scripts</li> <li>provide read only access to all environments... can include API access to facilitate automation script writing</li> <li>Create design template with customer pattern</li> </ul>
<b>Improvement Metrics</b> <ul style="list-style-type: none"> <li>Lead Time (post-dev to prod)</li> <li>Scrap Rate</li> </ul>	
<b>Current Condition</b> <ul style="list-style-type: none"> <li>Apps are not developed in production-like environments (not testing F5 behavior)</li> <li>Ops teams cannot practice or learn</li> <li>App teams have no visibility into constraints</li> <li>No remediation capabilities for app support teams</li> <li>No repeatable pattern for GM health activity</li> <li>80% S/R with 2-3 rework cycles</li> <li>50% cause outages</li> </ul>	<b>Blockers</b> <ul style="list-style-type: none"> <li>Financial approval (Jennifer)</li> <li>Segregation between environments (Mark)</li> <li>Non-standard request types (Susan)</li> <li>Two network teams with different rules (Mark)</li> </ul>

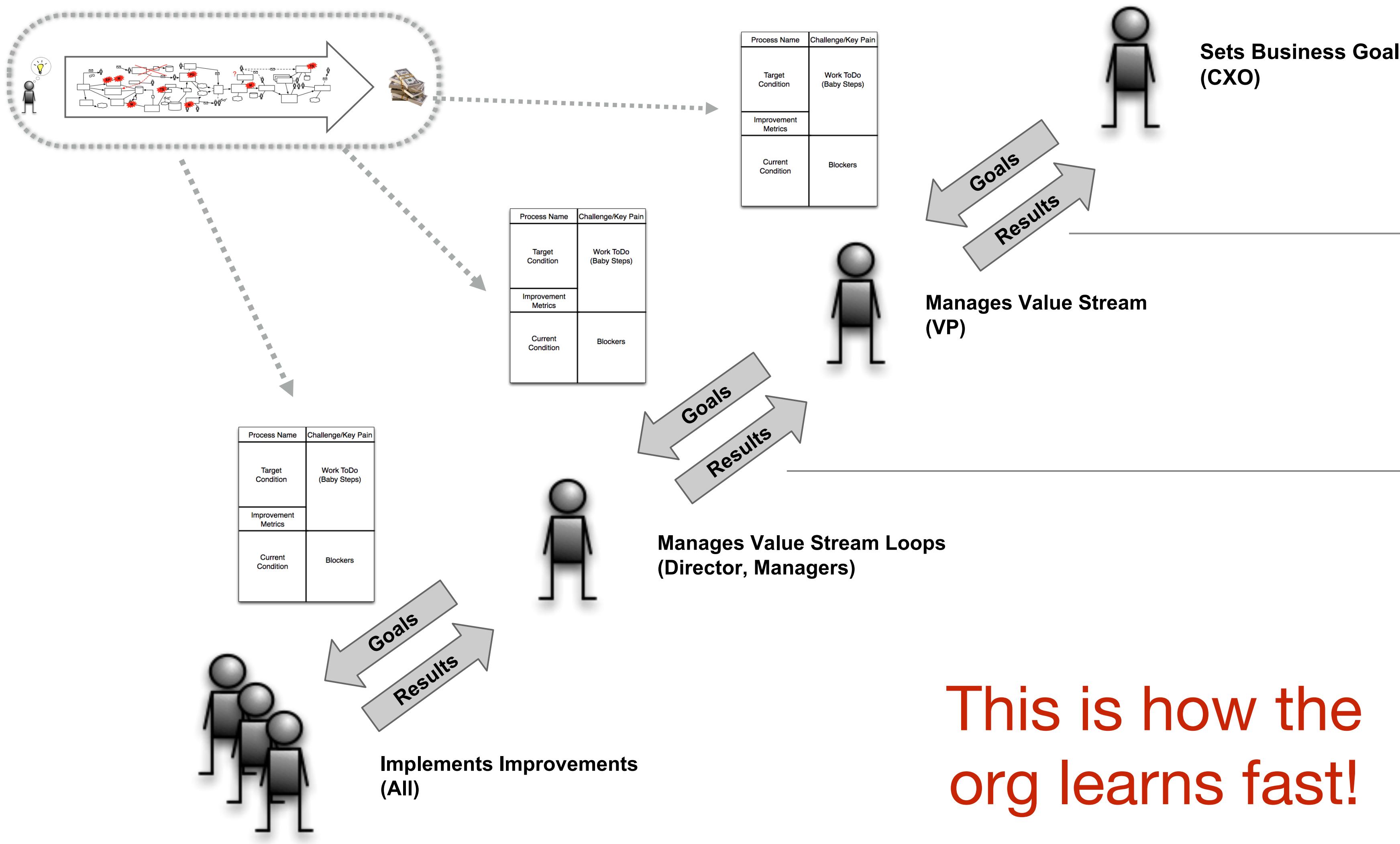
# Using Storyboards: Part Sales, Part Coaching



Inspiration: Toyota Kata

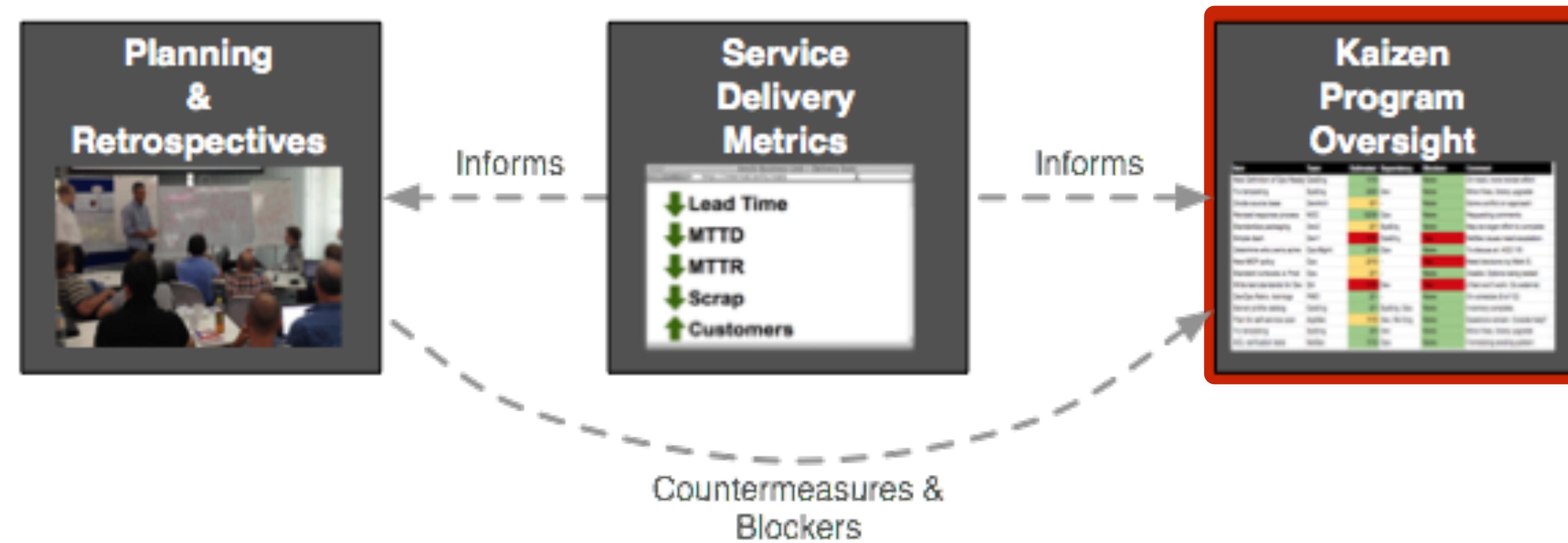


# Repeatable and Scalable Coaching Pattern



This is how the  
org learns fast!

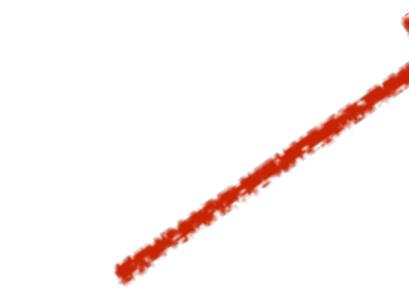
# Elements of a DevOps Kaizen Program



# Kaizen Program Oversight

---

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



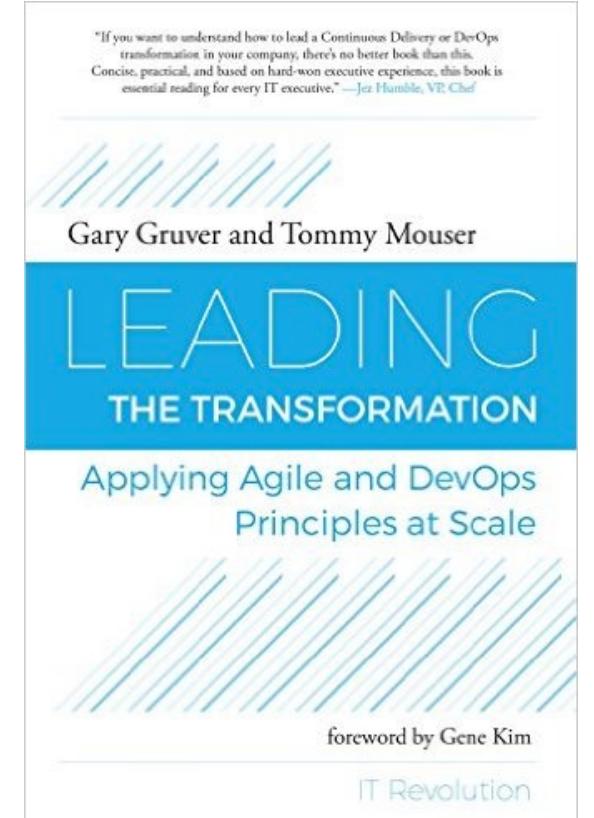
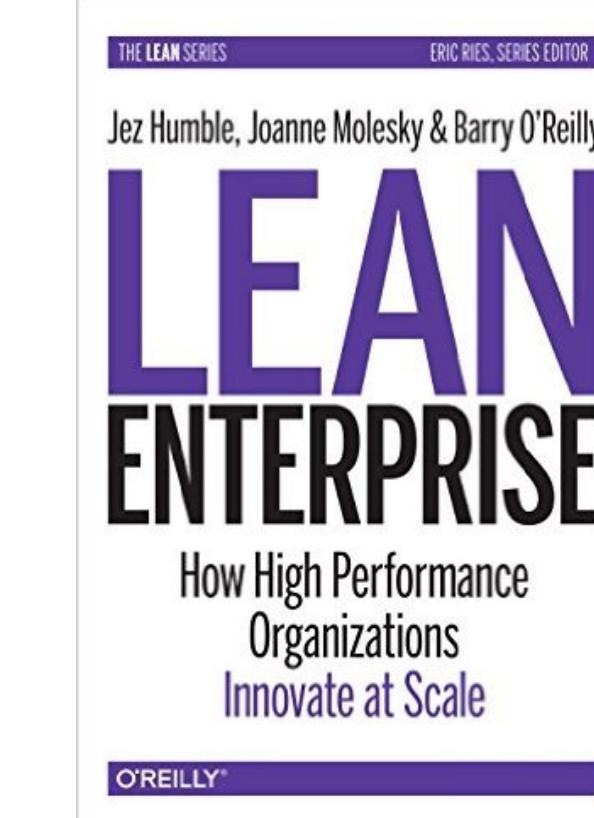
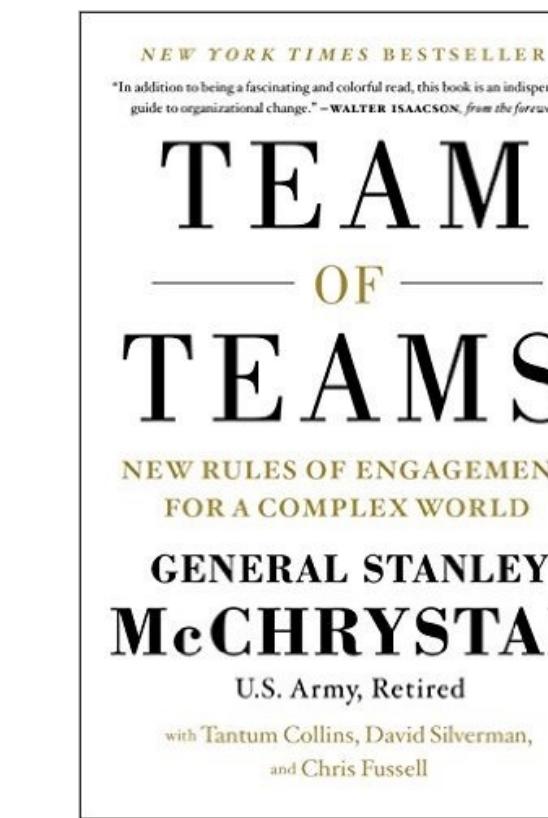
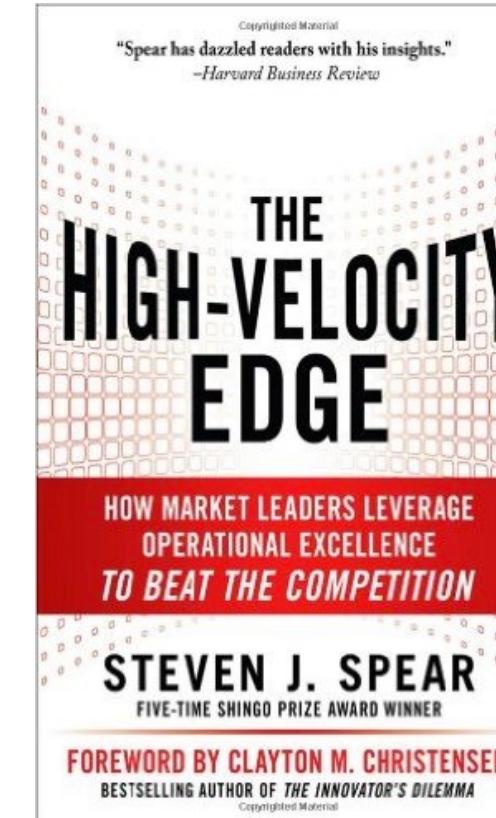
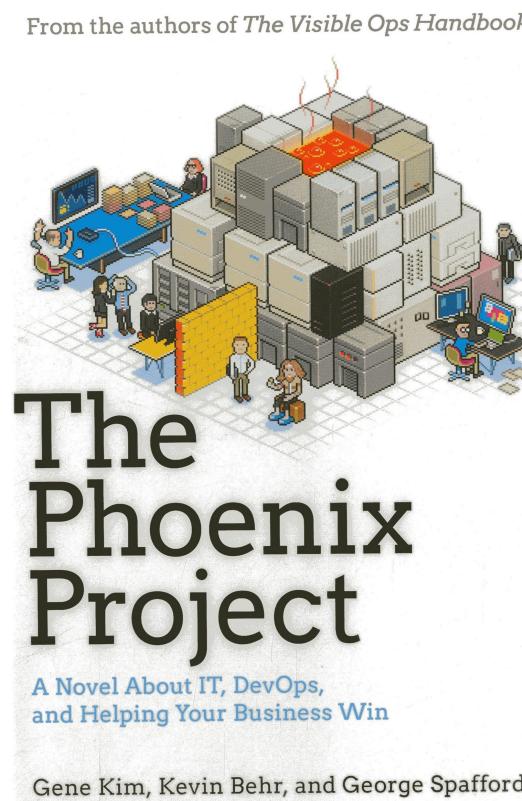
This (and only this) is what the Kaizen Program Oversight Group does!

# Kaizen Program Oversight

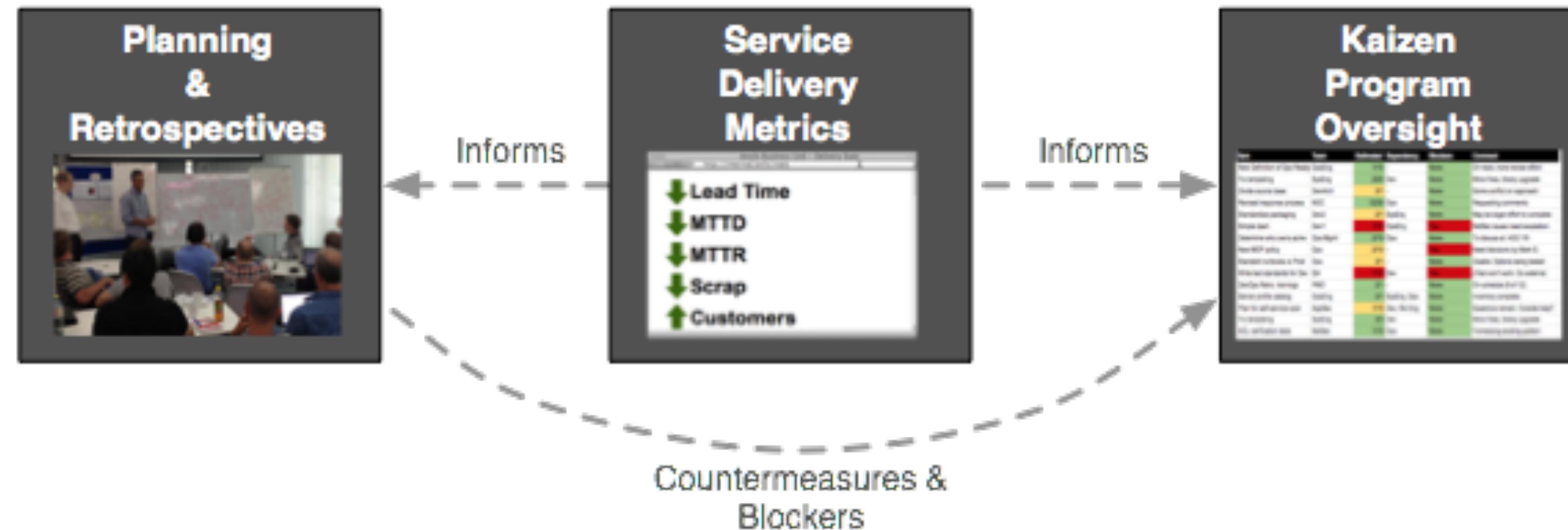
---

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

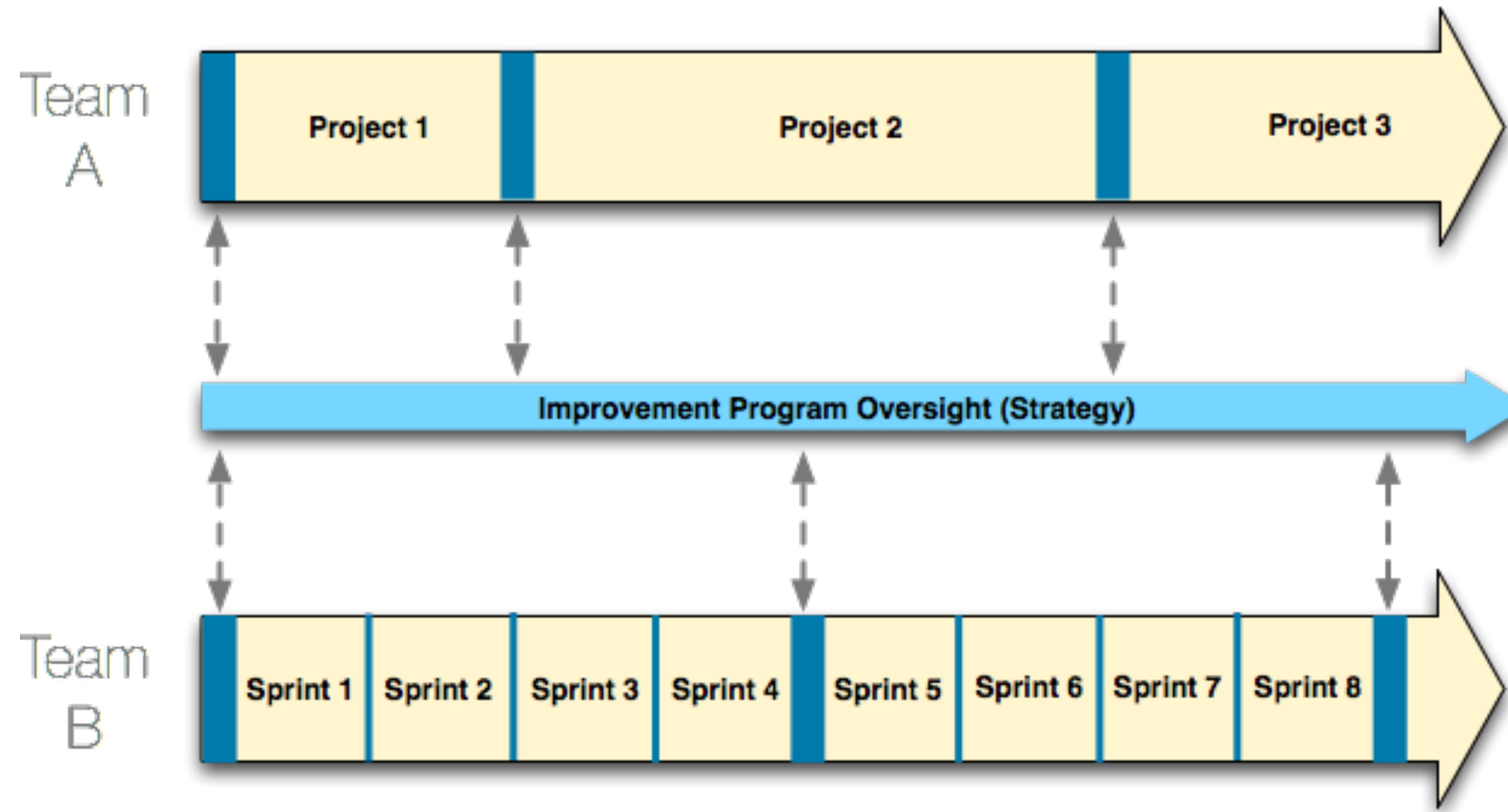
Inspire Executives with:



# Elements of a DevOps Kaizen Program



# DevOps Kaizen Program is an overlay for any delivery methodology

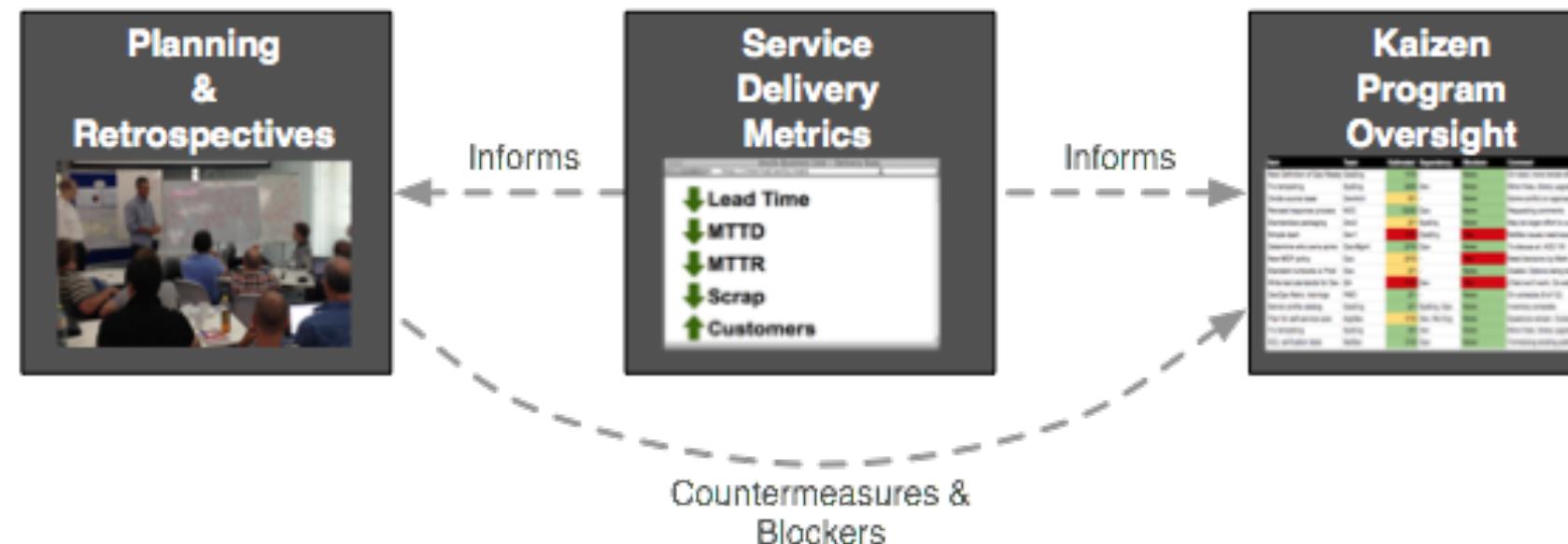


Full Retrospective & Planning

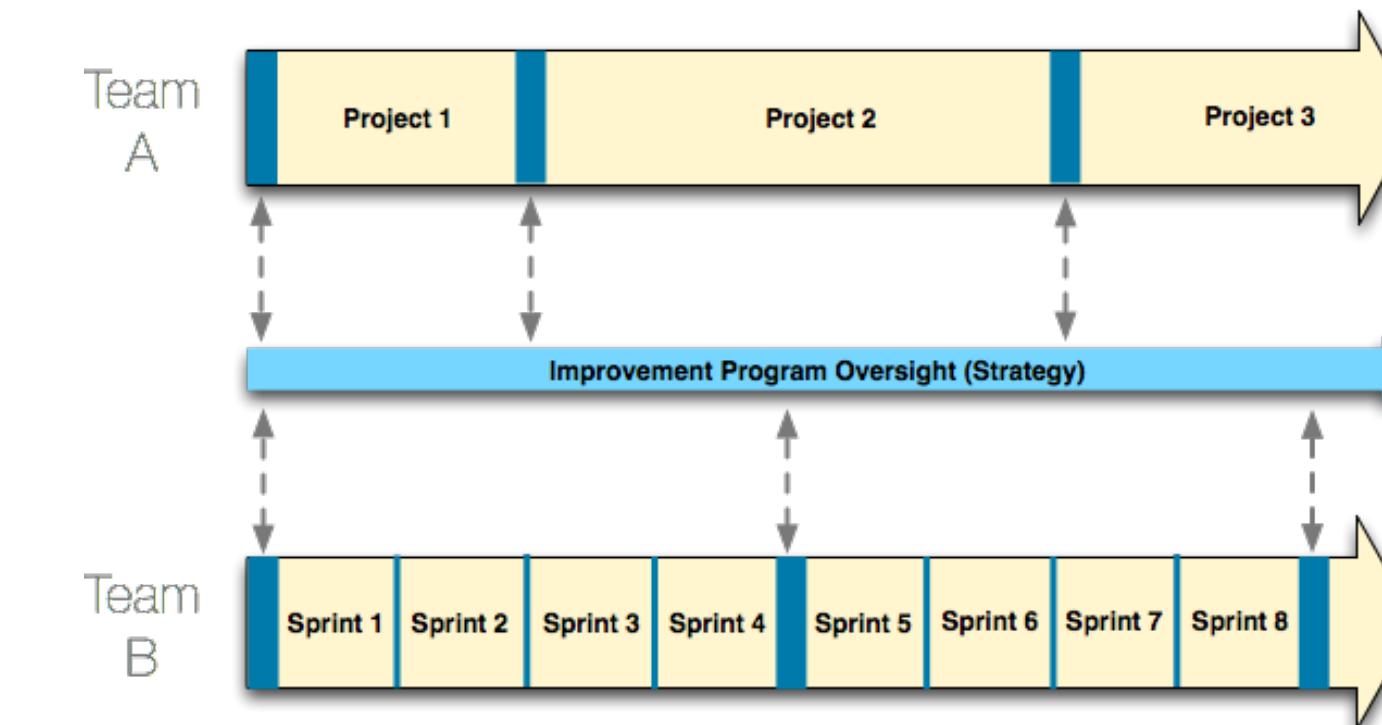
Refresh Retrospective

# DevOps Kaizen: Let's Recap!

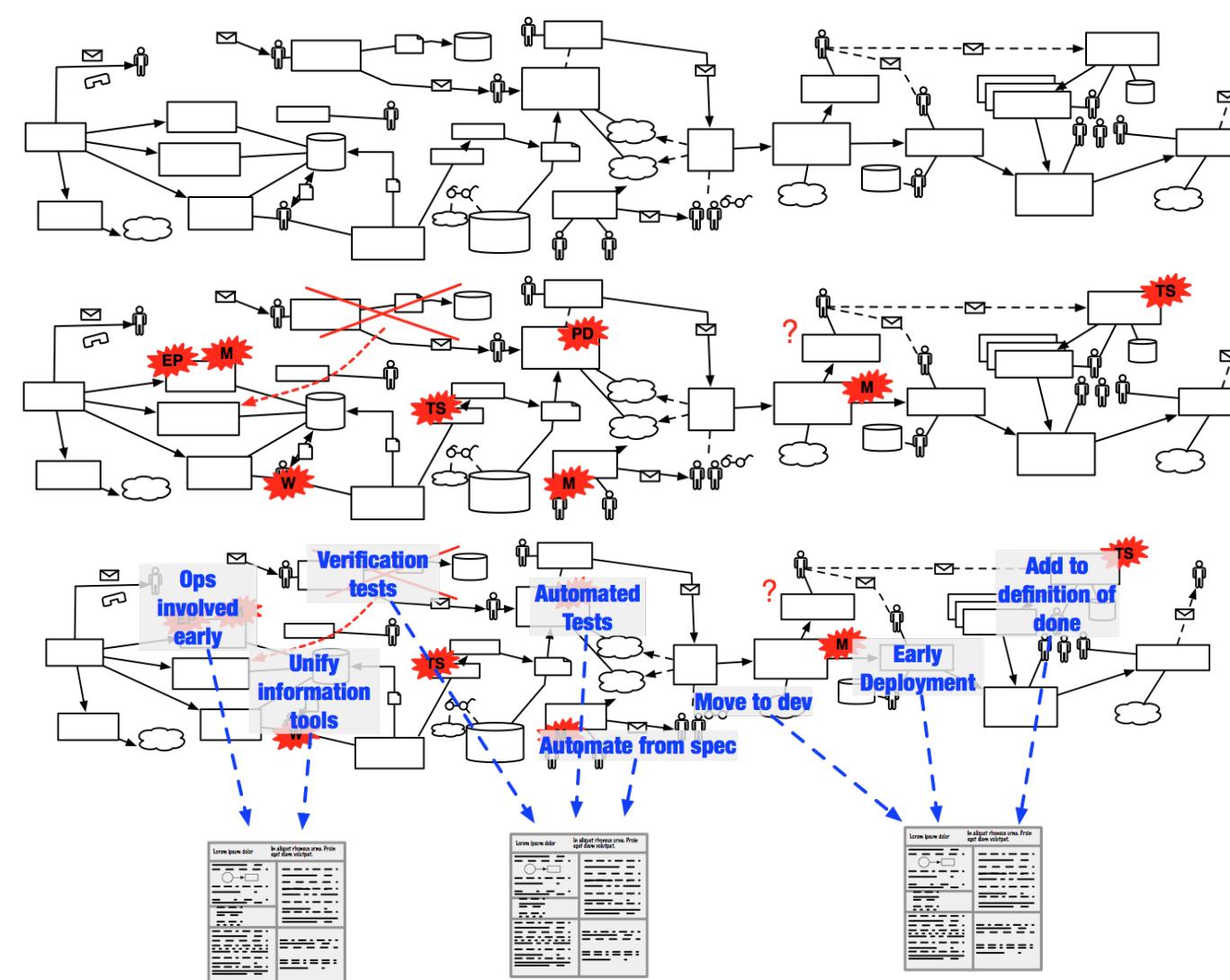
## Establish program elements



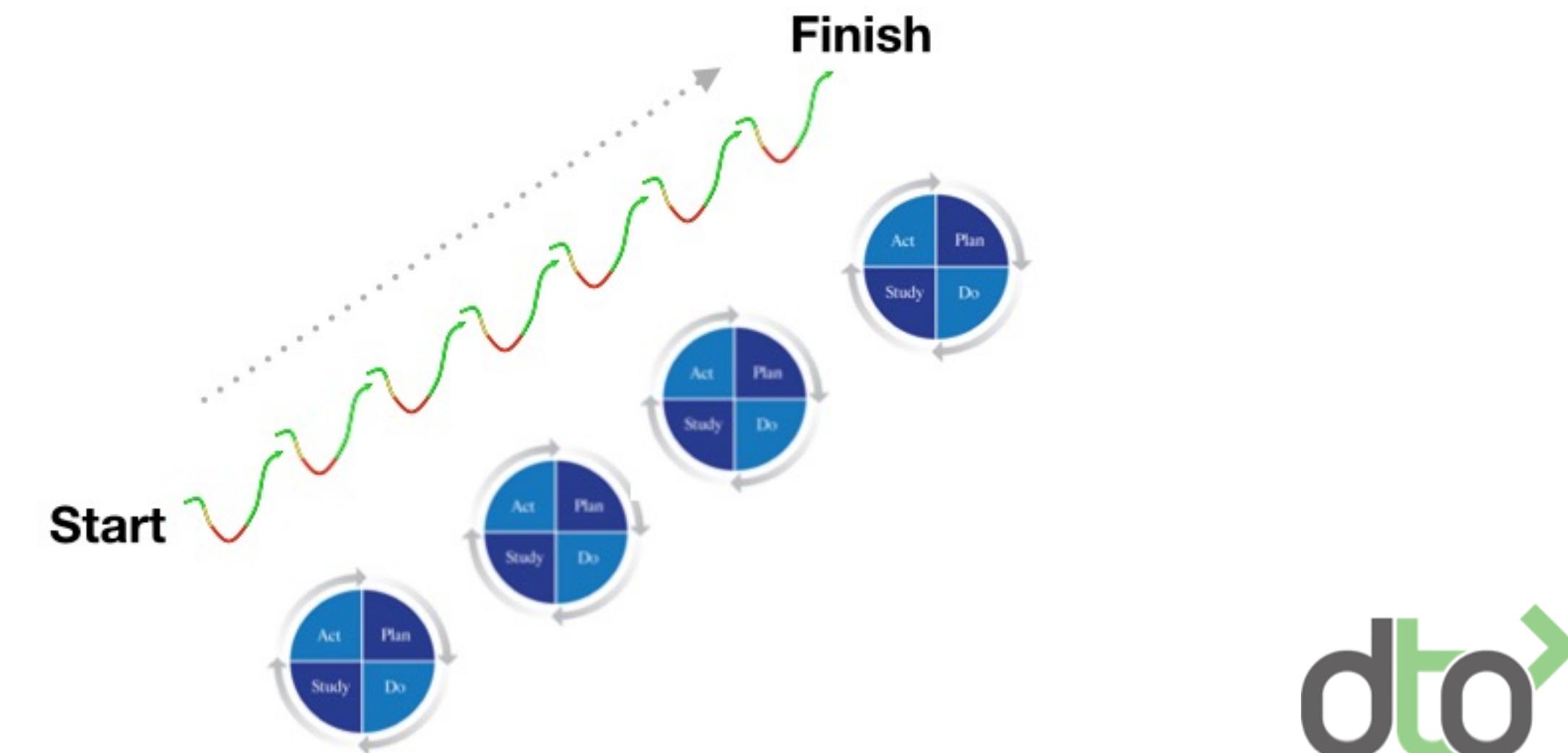
## Build into your operating model



## Make the work visible



## Focus on Continuous Improvement



# Damon Edwards



**@damonedwards**



**damon@dtosolutions.com**