

# *Top Learnings Writing Wiring The Winning Organization*

**Gene Kim  
Dr. Steve Spear**

# 2014: Dr. Steve Spear at MIT Sloan



# The “Magic” That Winning Orgs Have

- Winning organizations do extraordinary things, more than any single individual could ever do alone, fully unleashing people's creativity and capabilities
- Versus organizations that constrain, or even extinguish entirely, the creativity and problem solving capabilities of people within them

# The Book

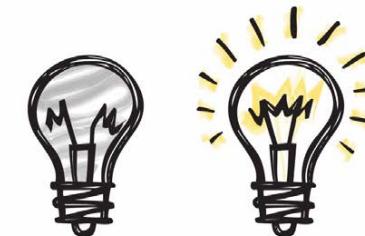
- *Wiring the Winning Organization:*

*Liberating Our Collective Greatness Through Slowification, Simplification, and Amplification*

- Coming November 2023

LIBERATING OUR COLLECTIVE GREATNESS THROUGH  
SLOWIFICATION, SIMPLIFICATION, AND AMPLIFICATION

## Wiring the Winning Organization



GENE KIM and  
STEVEN J. SPEAR

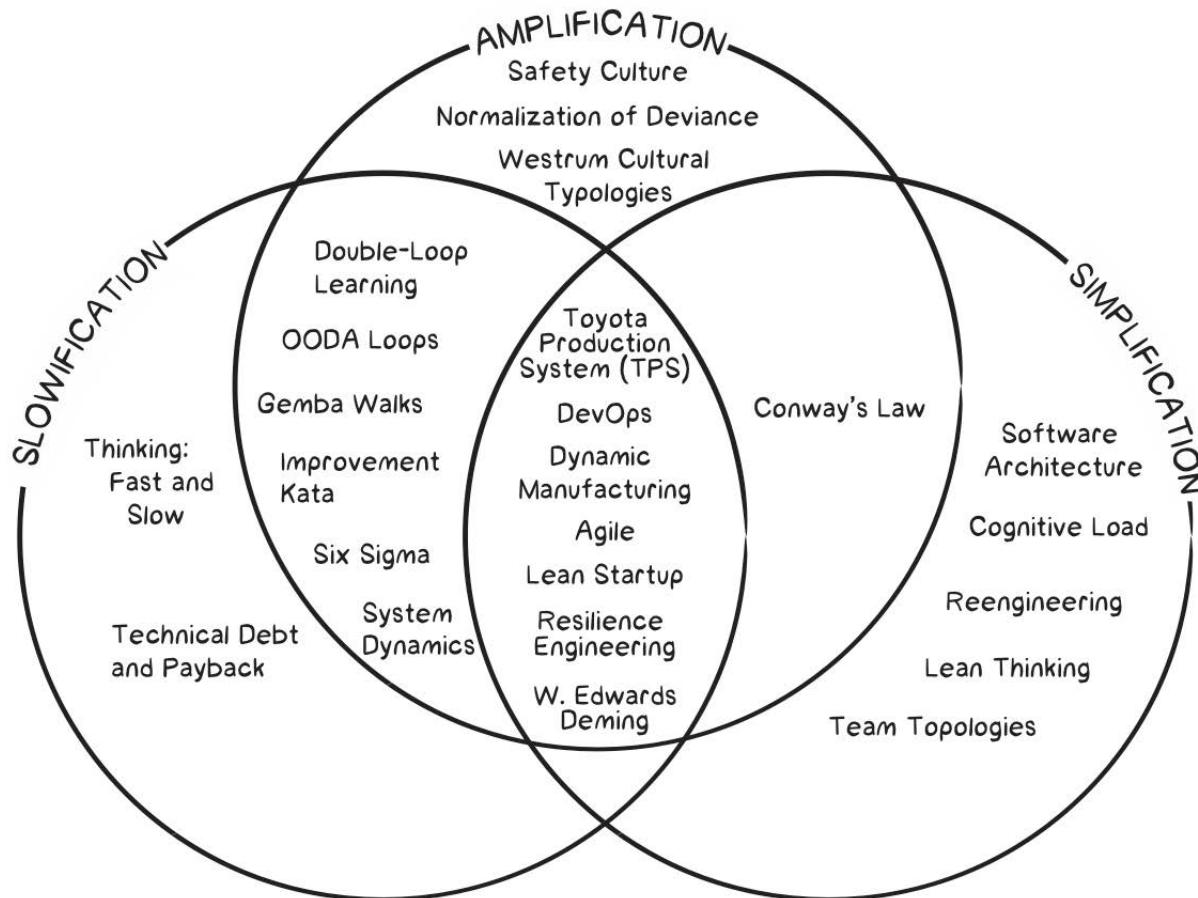
*Foreword by ADM John Richardson, US Navy (Retired)*  
*former Chief of Naval Operations*

# The Book



GENE KIM and  
STEVEN J. SPEAR

*Foreword by ADM John Richardson, US Navy (Retired)*  
former Chief of Naval Operations





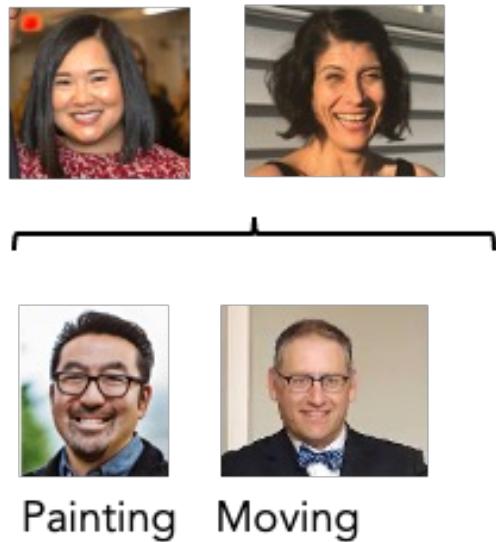


# Part 1



COUPLED: those that are connected together

# Part 2



- Three interdependent steps
  - Remove furniture
  - Paint room
  - Restore furniture

# **Slowification**

# The “Magic” That Winning Orgs Have

- Ideal
  - Enough time is scheduled for planning, practice, experimentation, and improvement.
  - Practice environments simulate performance conditions but with lower stakes, so teams can safely fail, learn, and improve.
  - Capturing lessons learned throughout the process to continually improve.
- Not ideal
  - Work must be done in high-stakes performance environments with no ability to redo, with high consequences of failures.
  - No time to pause in production, to slow down to speed up, to stop sawing to sharpen the saw.

# Slowification: DevOps Examples

- Amazon Game Days (Jesse Robbins)
- Google DiRT (Kripa Krishnan)
- Netflix Chaos Monkey (Adrian Cockcroft)
- Vanguard (Christina Yakomin, Robbie Daitzman)

## Amazon EC2 outage downs Reddit, Quora



@SCVNNGR  
SCVNNGR  
The sky is falling! Amazon's cloud seems  
to be down (raining?) so we're  
experiencing some issues too. Be back  
soon!

5 hours ago via web

Retweeted by [RealAmandaStone](#) and others



SCVNNGR and other sites took to Twitter after a rare and major outage of Amazon's cloud-based Web service.

Recommend 990 people recommend this.

By Julianne Pepitone, staff reporter April 22, 2011: 7:29 AM ET

@RealGeneKim

# Inject Failures Often

## The Netflix Tech Blog

### 5 Lessons We've Learned Using AWS

We've sometimes referred to the Netflix software architecture in AWS as our Rambo Architecture. Each system has to be able to succeed, no matter what, even all on its own. We're designing each distributed system to expect and tolerate failure from other systems on which it depends.

One of the first systems our engineers built in AWS is called the Chaos Monkey. The Chaos Monkey's job is to randomly kill instances and services within our architecture. If we aren't constantly testing our ability to succeed despite failure, then it isn't likely to work when it matters most – in the event of an unexpected outage.

**You Don't Choose Chaos Monkey...  
Chaos Monkey Chooses You**



# Underpinning Theory

- Dr. Daniel Kahneman and Dr. Amos Tversky described how there's two types of thinking
  - Fast thinking: biases, habits, routines
  - Slow thinking: deliberative, creative, contemplative
- This led to Kahneman's book, "Thinking, Fast and Slow"
- Kahneman won the Nobel Prize for Economic Sciences



There is an adage that slowing down to improve is worth the investment.

Many popular cliches exist for this: "slow down to sharpen the saw, so you can saw faster," "slow is smooth, smooth is fast," "improvement of daily work can be more important than daily work itself."

Consider that "accelerate" doesn't capture these meanings.

What is a verb that does capture this meaning?



The verb "refine" could capture the meaning you're describing. It implies taking the time to analyze, improve, and perfect a skill or process. Through refinement, you're slowing down to make the necessary adjustments for a more efficient or effective output in the future.

Another possible term could be "cultivate." It refers to the act of nurturing or developing something over time, often in the context of skills, qualities, or relationships. The emphasis on time and care is in alignment with the adages you mentioned.

Lastly, "honing" might work too. It implies the focused effort to improve or perfect an existing skill or practice, much like honing a blade makes it more effective.

While none of these words might perfectly encapsulate all the nuances of the saying:

Regenerate

Source: <https://chat.openai.com/share/be2455c5-7a9-44e2-9db7-49f95e373c63>



can you provide more candidate verbs?



Absolutely, here are some additional verbs that might convey the concept of improving through slowing down:



1. "Polish": To refine or perfect something over time.
2. "Nurture": To care for and encourage the growth or development of.
3. "Temper": To bring to a desired hardness or strength by heating and cooling, metaphorically it can mean to make something more balanced, moderate, or controlled.
4. "Season": Over time, to become or make something more mature, experienced, or proficient.
5. "Foster": To encourage the development or growth of ideas or skills.
6. "Enhance": To intensify, increase, or further improve the quality, value, or extent of.
7. "Elevate": To raise or lift something up to a higher position, but can also mean to improve or make something more refined.
8. "Mature": To become fully developed or ripe, used metaphorically it can mean to become more effective or refined over time.
9. "Sculpt": The process of creating a refined, three-dimensional object, which metaphorically can be used to indicate the process of careful refinement or improvement.
10. "Mold": To shape or form something in a gradual way, often used metaphorically for personal or skill development.

11. "Optimize": Make the best or most effective use of a situation or resource, often imp

Regenerate

@RealGeneKim



In our new book, we have three concepts: "simplify" (which signifies making problems easier to solve, through linearization, modularization, and linearization), "amplify" (which signifies making problems easier to detect, as in signal theory), and the third concept is called "slowify" (which signifies ensuring that problems are solved not in production, but during planning and performance)

What other names for "slowify" can you suggest, ideally not a made-up word



Based on the context you provided, here are a few suggestions for alternative terms to "slowify":



1. "Anticipate": This term signifies being aware and preparing for potential future problems or changes, which seems to align with your intent of addressing problems during planning and preparation rather than in production.
2. "Preempt": This suggests addressing problems before they even occur, which matches the idea of solving issues during planning and performance.
3. "Strategize": This signifies the process of creating a strategy or plan, which includes identifying potential problems and working out how to solve them beforehand.
4. "Forecast": Similar to anticipate, but with a more explicit focus on predicting future events or issues.
5. "Prepare": This could signify the action of getting ready in advance, which includes t

Regenerate



The German language is known for its unique ability to combine words to create compound terms that convey complex concepts. For the concept you described, the German language may not have a single existing word that perfectly fits, but you could consider creating a compound term.



Here are some suggestions:

- 
1. "Vorausplanungsverbesserung": This compound word combines "Vorausplanung" (advance planning) with "Verbesserung" (improvement), implying the idea of improving through advance planning.
  2. "Langsamkeitsverbesserung": Combines "Langsamkeit" (slowness) with "Verbesserung" (improvement), emphasizing improvement through slowing down.

For short phrases, you might consider:

1. "Langsam planen und verbessern" (slowly plan and improve).
2. "Verbesserung durch Vorausplanung" (improvement through anticipation/advance planning).

Remember that these suggestions are approximate translations, and the specific nuances might vary. You may want to consult with a native German speaker or professional translator to ensure the chosen term or phrase best fits your concept.

 Regenerate

@RealGeneKim

# **Simplification**

# Examples Of DevOps Practices

- Simplification
  - Incrementalization: Agile, incremental delivery
  - Modularization: service oriented architectures, APIs
  - Linearization: automation, continuous integration/delivery

# The “Magic” That Winning Orgs Have

- Ideal
  - Everyone solving important problems, all the time, in parallel
  - Everyone has what they need (information, approvals, requirements, decision rights), in the right format, at the right time, interacting with all the right people
- Not ideal
  - Everyone is “stuck,” unable to do what needs to be done
  - No one has what they need, when they need it, in the right format, at the right time
  - Even small efforts require super-heroic efforts

# Amazon 1998

Wednesday, October 13, 1999

SEARCH  
All Products  Search of the Day: saffron

WELCOME BOOKS MUSIC VIDEO TOYS & GAMES ELECTRONICS e-CARDS AUCTIONS zSHOPS  
HOW TO ORDER GIFT SERVICES OUR GUARANTEE SITE GUIDE COMMUNITY

WELCOME TO amazon.com

Hello! Shopping at Amazon.com is 100% secure--guaranteed.  
Already a customer? [Sign in.](#)

Vote in our [Millennium Poll](#)--you could win 300 CDs, books, and videos!

In Books  
Test Case  
  
The postwar inventors of the Scholastic Aptitude Test hoped to produce a brainier brand of meritocracy in the United States. But as Nicholas Lemann reveals in [The Big Test](#), the SAT hit a great many ideological potholes--and ended up creating yet another, pencil-pushing elite. Go to [Books](#)

Tricks, treats, and costumes for all in our [Halloween Bootique](#).

Amazon.com 100 Hot Books  
Updated Hourly  
1. [The Carbohydrate Addict's Lifespan Program : A](#)

# Amazon 2002

The screenshot shows the Amazon.com homepage from 2002. A red box highlights the top navigation bar, which includes the 'amazon.com' logo, 'Ann's Store', 'Books', 'See All 32 Product Categories', 'Your Account', 'Cart', 'Wish List', 'Help', and a yellow gift icon. Below this, there are links for 'Advanced Search', 'Browse Subjects', 'Bestsellers', 'The New York Times® Best Sellers', 'Magazines', 'Corporate Accounts', 'Amazon Shorts', 'AmazonConnect', 'Bargain Books', 'Used Books', and 'Textbooks'. A search bar at the top contains the query 'joseph stiglitz' with a dropdown menu showing 'Books'. To the right of the search bar are 'GO' and 'Find Gifts' buttons. Further right is a 'Web Search' bar with an 'A9' logo and a 'GO' button.

**Results for joseph stiglitz**

**Related Searches:** [stiglitz](#); [jeffrey sachs](#); [paul krugman](#)

**So You'd Like to... Offer your advice**

**Customers who searched for joseph stiglitz ultimately chose:**

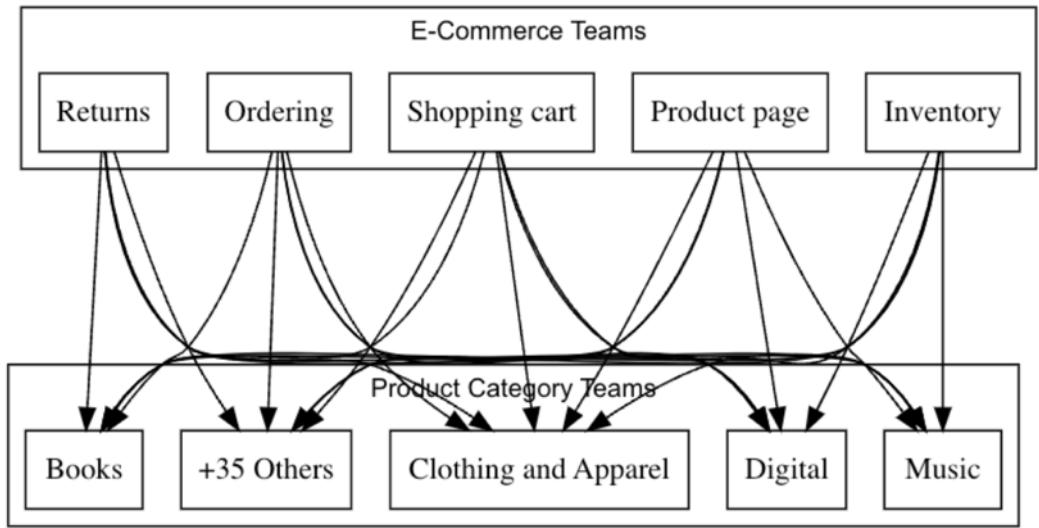
1.  [Globalization and Its Discontents](#) -- by Joseph E. Stiglitz; Paperback  
(Rate it)  
Buy new: \$10.37 -- Used & new from: \$8.65

**Listmania!** ▶ [Create a Listmania! list](#)

  
THE CORROSION OF CHARACTER  
COLLEGE YEARS -

# Amazon 2002

- E-commerce teams
  - Product page
  - Shopping cart
  - Ordering
  - Returns
  - Inventory
- Product teams
  - Books
  - Music
  - Digital
  - Apparel
  - +35 more...



# Amazon 2004

- With the introduction of digital products, Dr. Werner Vogels, CTO of Amazon since 2005, described how when those teams “wanted to add something to the order pipeline, a physical delivery address was required.

“There was no way around [not providing a physical delivery address]. They would walk to the 80 different ordering teams and say, ‘We need to change this.’ The ordering teams would respond that they hadn’t budgeted for it.” So now those teams were stuck, unable to ship products.

# Amazon Results

- 1999: thousands of deployments/year
- 2001: tens of deployments/year

# The \$1 Billion Amazon API Rearchitecture

1. All teams will henceforth expose their data and functionality through service interfaces.
2. Teams must communicate with each other through these interfaces.
3. There will be no other form of interprocess communication allowed
4. It doesn't matter what technology you use, HTTP, Corba, Pubsub, Bezos doesn't care.
5. Service interfaces without exception must be designed from the ground up to be externalizable
6. Anybody who doesn't do this will be fired.
7. Thank you, have a nice day.

("#7 is obviously a joke, because obviously Bezos doesn't care whether you have a good day or not")

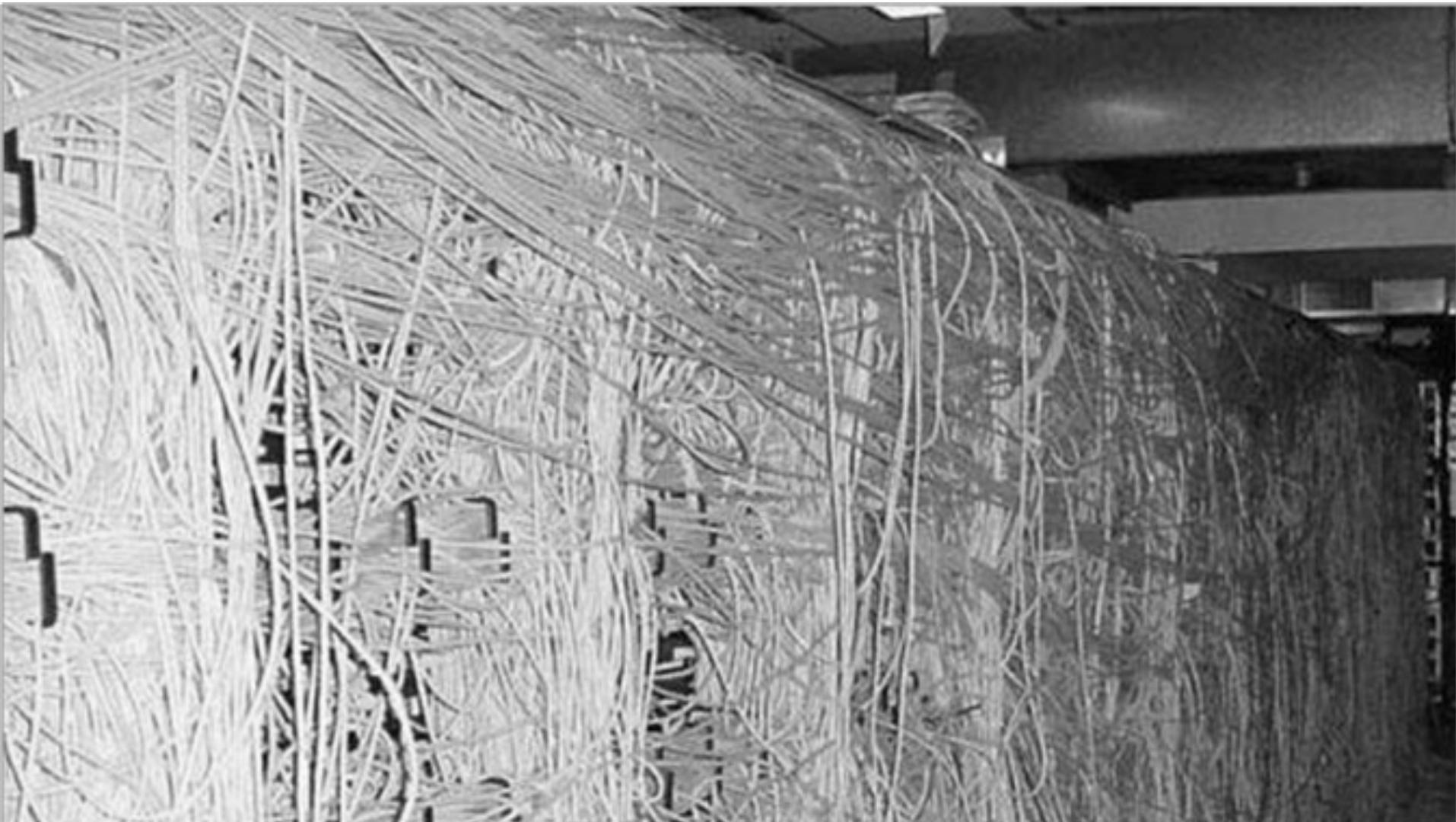
Who enforced this?

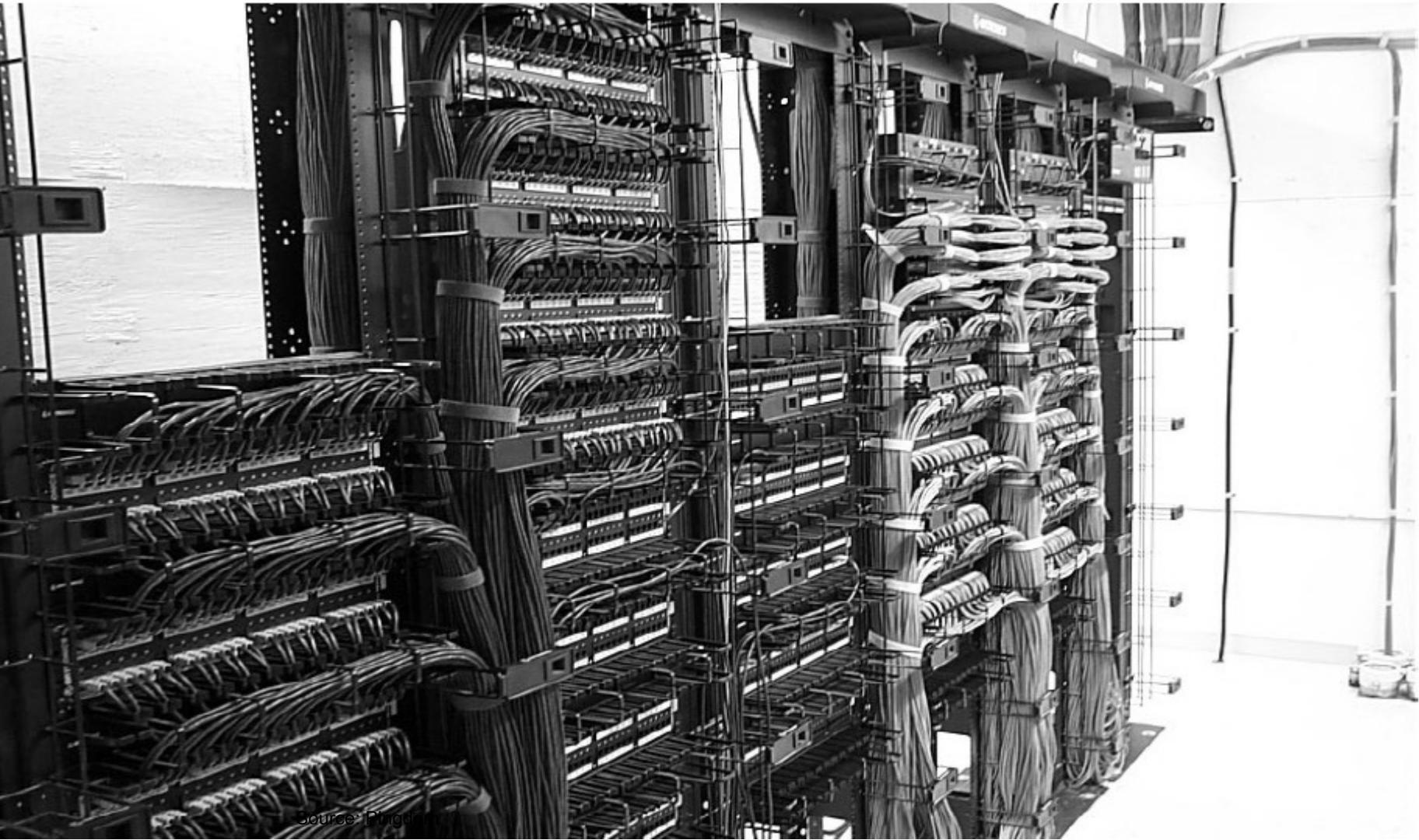
Amazon CIO: Rick Dalzell, a former U.S. Army Ranger

# Amazon Results

- 1999: thousands of deployments/year
- 2001: tens of deployments/year
- 2011: 15K deployments/day
- 2015: 136K deployments/day

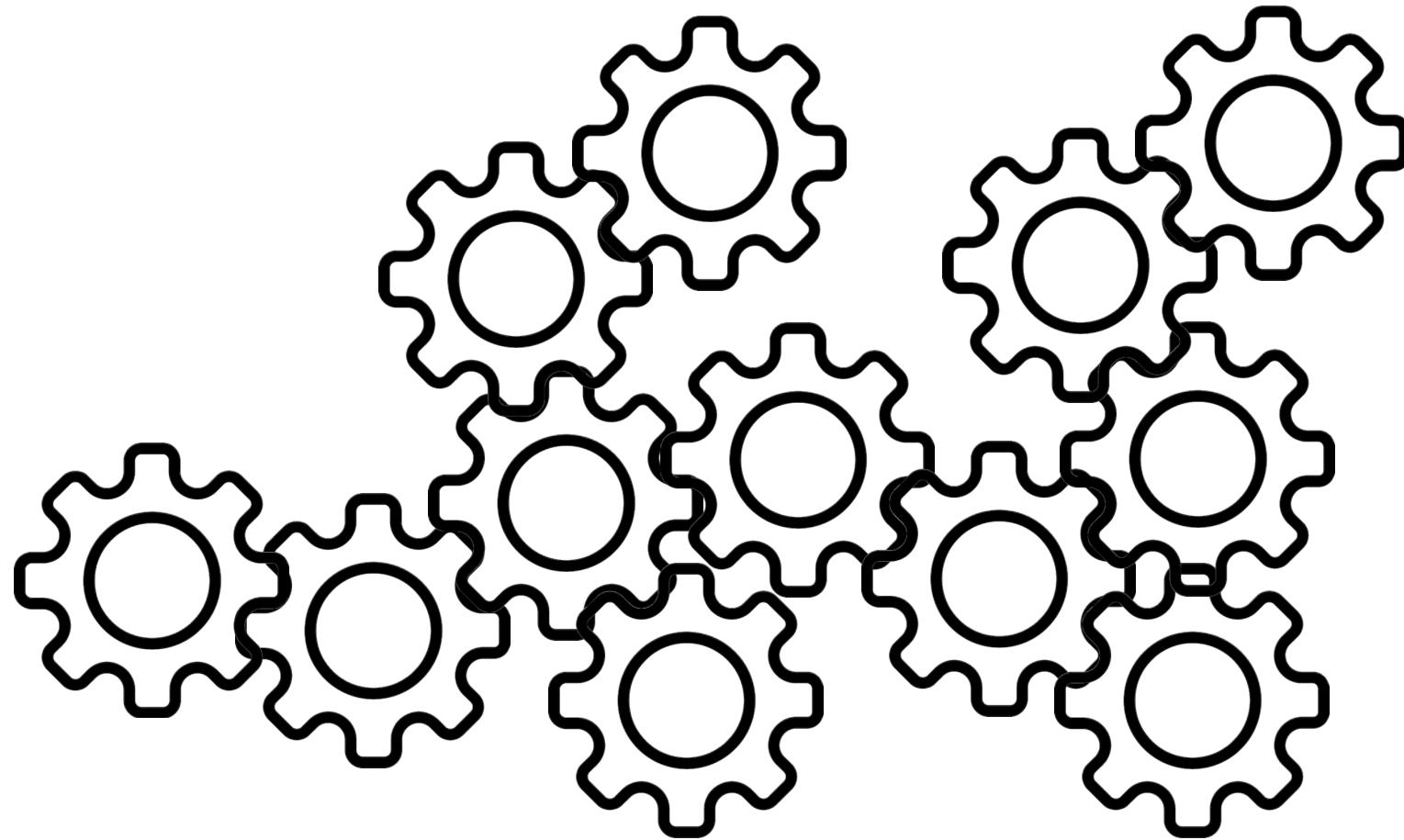


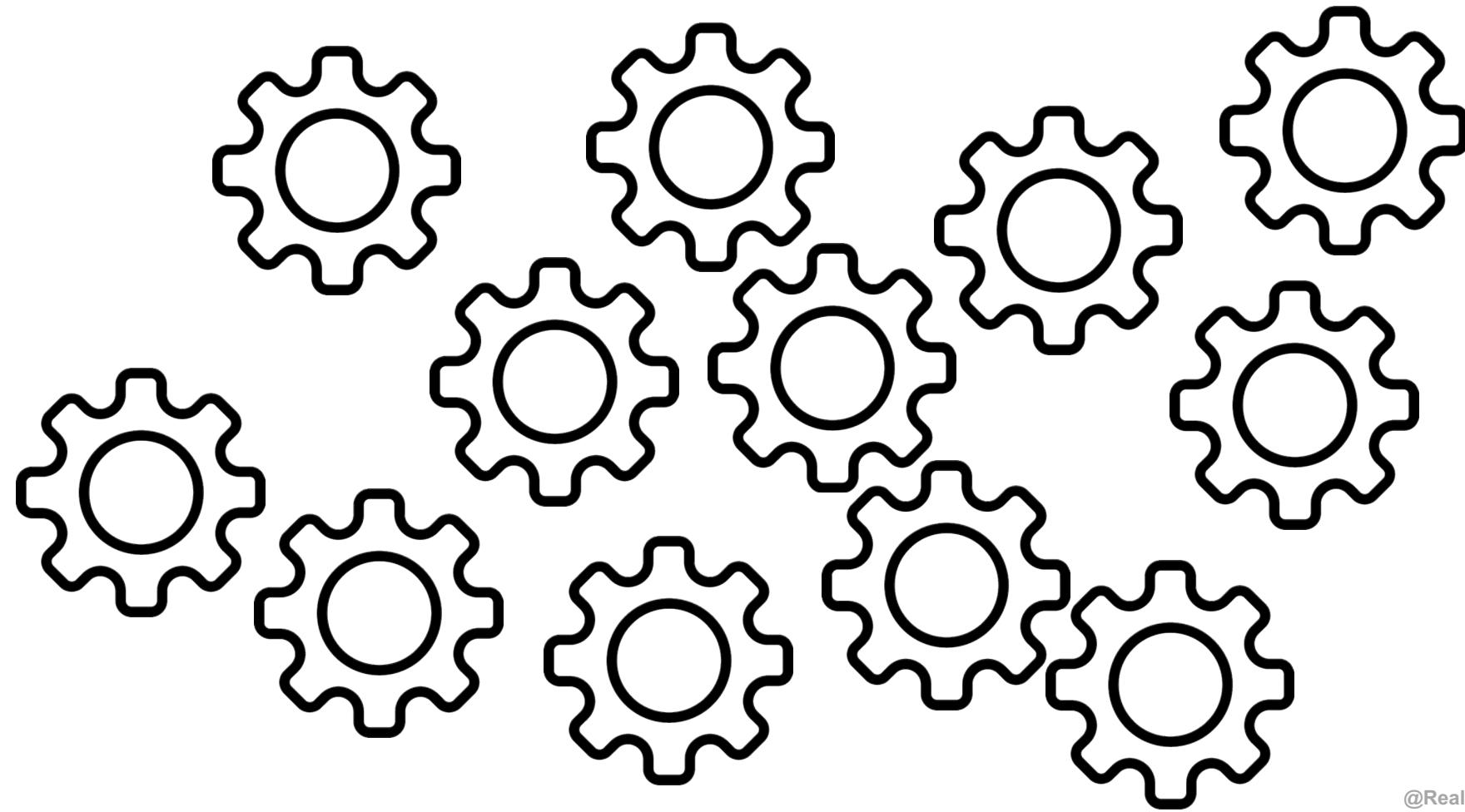




Source: Pingdom

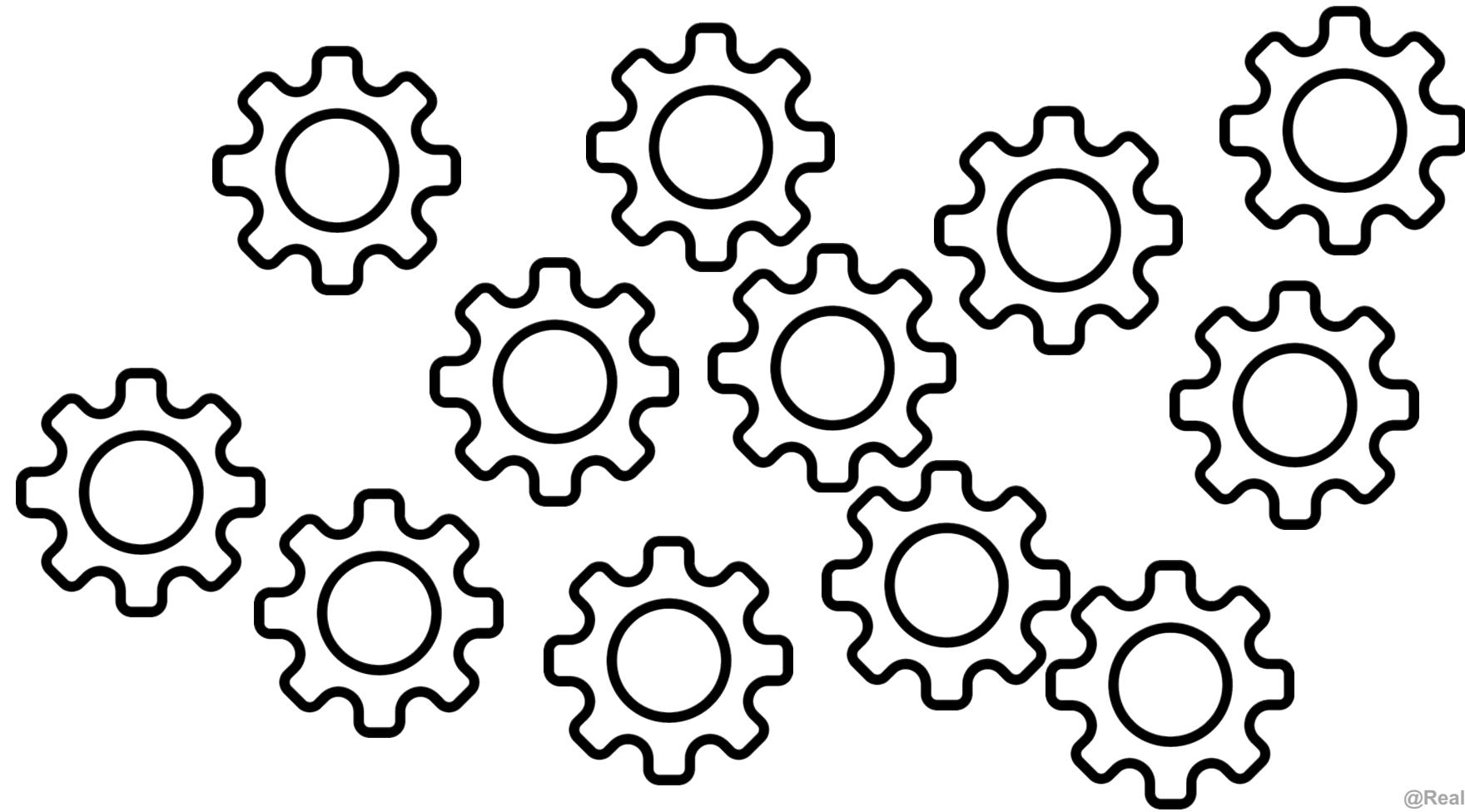
im

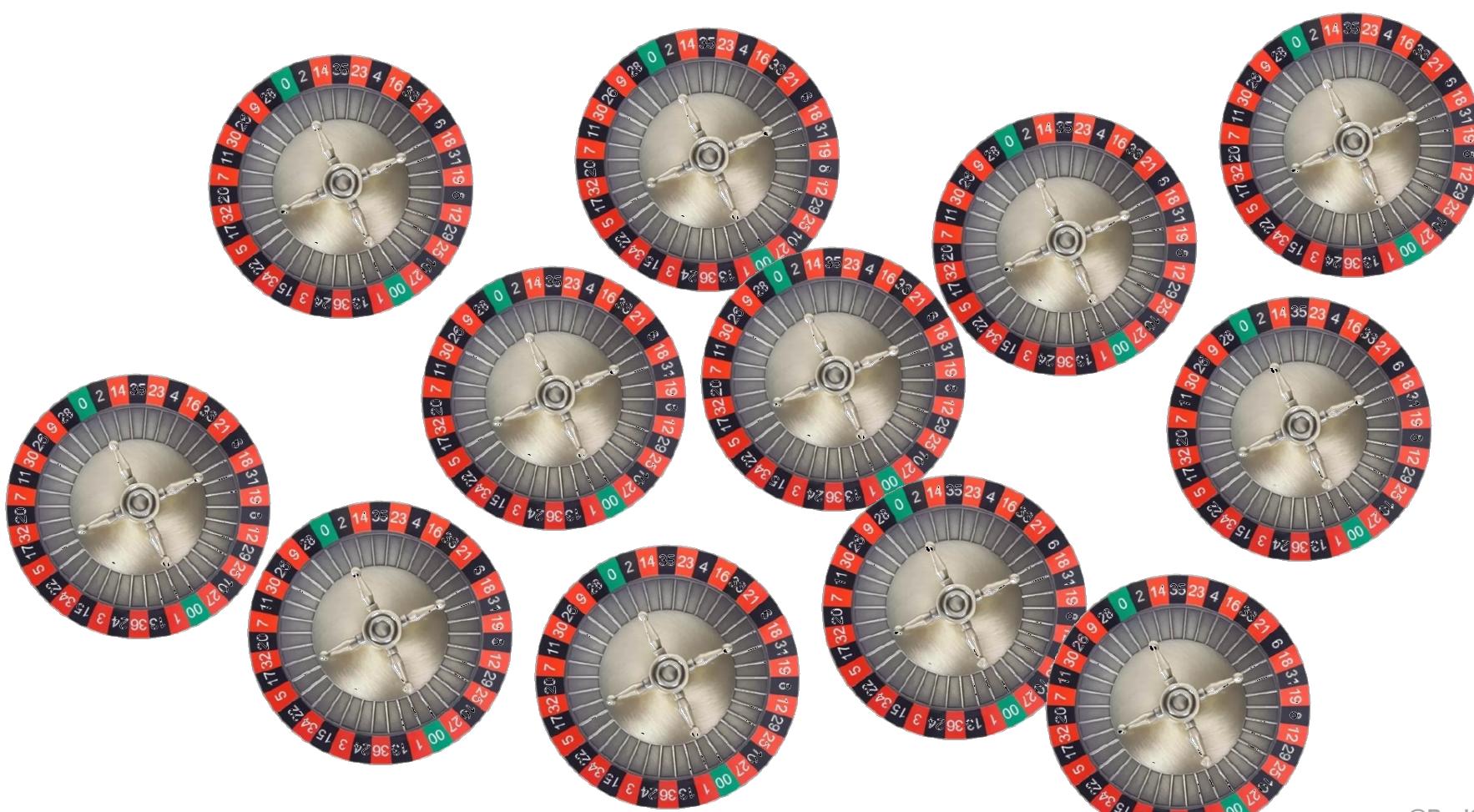


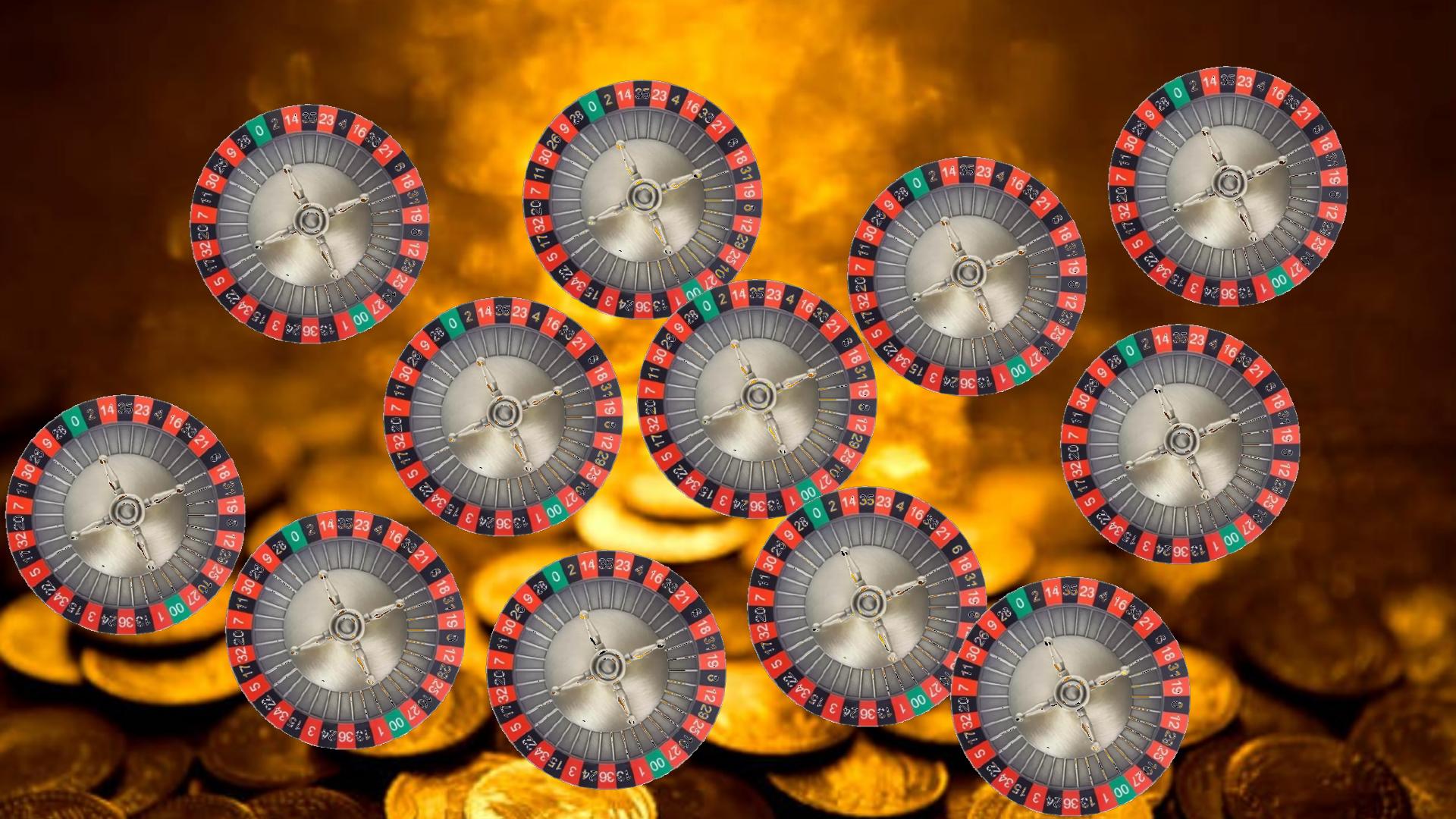


# Underpinning Theory

- Drs. Robert Merton, Fischer Black, and Myron Scholes work on option theory showed how one can decouple decisions tomorrow from conditions today — this gives latitude of action to decision-makers that they otherwise wouldn't have.
  - This won them the Nobel Prize in Economic Sciences in 1997
- Drs. Baldwin and Clark showed how one can decouple actions (spatially) in one location from those in another, providing independence of action that otherwise wouldn't have existed







# Modularity vs. Linearize

- What modularity does for parallel processes, linearization does for sequential processes (such as a manufacturing system, or CI/CD processes)

# **Amplification**

# Underpinning Theories

- James Maxwell: control theory (1867)
- Dr. Harry Nyquist and Dr. Claude Shannon: information theory (1946)

■ “This is a fabulous book that I highly recommend. Successful organizations flow from leaders who create the conditions in which many others thrive.”

—**Paul Gaffney, former CTO and head of technology, The Home Depot, Kohl’s, Dick’s Sporting Good**

■ “This book clearly teaches you how to rewire your organization to move with focused sustained urgency and win.”

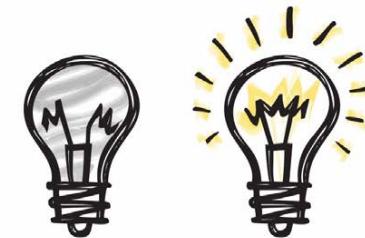
—**Courtney Kissler, SVP Customer and Retail Technology, Starbucks**

■ “The framework in this book brings together the most useful insights I’ve learned over my thirty years of study and practice. Save yourself the time and start here.”

—**Jeffrey Fredrick, coauthor of Agile Conversations, cohost of Troubleshooting Agile podcast**

LIBERATING OUR COLLECTIVE GREATNESS THROUGH  
SLOWIFICATION, SIMPLIFICATION, AND AMPLIFICATION

# Wiring the Winning Organization



GENE KIM and  
STEVEN J. SPEAR

*Foreword by* ADM John Richardson, US Navy (Retired)  
former Chief of Naval Operations

- “Kim and Spear lay out an amazing vision of the social circuitry for organizations to not only handle this, but thrive while doing so.”

**—Phil Venables, Chief Information Security Officer, Google Cloud; former Board Director, Goldman Sachs Bank**

- “With expert insights and practical strategies, Kim and Spear unravel the web of organizational structures and offer a blueprint to create more resilient and adaptive organization.”

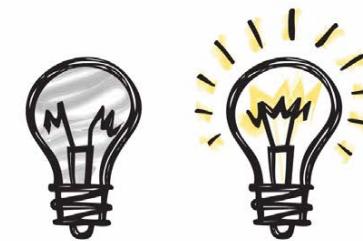
**—David Silverman, CEO of CrossLead, coauthor of Team of Teams**

- “The authors not only provide an encompassing schema for those looking to drive performance by improving operations, but they help take practices developed for particular industries and functions and generalize them across all contexts. The guidance they provide is outstanding.”

**— Dr. Joel Podolny, CEO, Honor Education, former VP & Dean of Apple University, former Dean of Yale School of Management**

LIBERATING OUR COLLECTIVE GREATNESS THROUGH  
SLOWIFICATION, SIMPLIFICATION, AND AMPLIFICATION

# Wiring the Winning Organization



GENE KIM and  
STEVEN J. SPEAR

*Foreword by* ADM John Richardson, US Navy (Retired)  
former Chief of Naval Operations

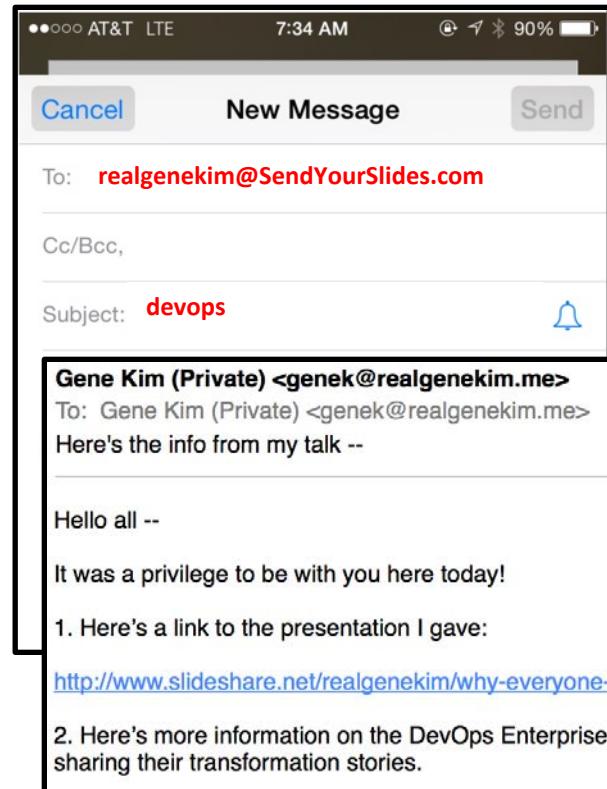
# Want More Learn More?

To receive this presentation and the following:

- PDF and audio excerpts from *The Unicorn Project*
- Eight excerpts from *Beyond The Phoenix Project* audio series w/John Willis
- The 140 page excerpt of *The DevOps Handbook*
- The 140 page excerpt of *The Phoenix Project*
- Videos and slides from DevOps Enterprise 2014-2019
- One hour excerpt of *The Phoenix Project* audiobook

Just pick up your phone, and send an email:

To: **realgenekim@SendYourSlides.com**  
Subject: **devops**



# Agenda

- Lecture
  - 10m: Gene introduction
  - 25m: Steve lecture
  - 15m: Gene concretization
- Exercises
  - 1h: Slowification
  - 55m: Simplification
  - 55m: Amplification

# My Definition of DevOps

The architecture, technical practices, and cultural norms  
that enable us to...

increase our ability to deliver applications and services...

quickly and safely, which enables rapid experimentation  
and innovation, and the fastest delivery of value to our  
customers...

while ensuring world-class security, reliability, and stability...

...so that we can win in the marketplace.

**Better Value,  
Sooner, Safer, Happier**

# **The Business Value Of DevOps Is Even Higher Than We Thought**

# Elite vs. Low Performers

|                                    | <i>Elite</i>                          | <i>Low</i>           | <i>Difference</i> |
|------------------------------------|---------------------------------------|----------------------|-------------------|
| <b><i>Deployment Frequency</i></b> | On-demand<br>(multiple times per day) | Monthly or quarterly | 208x              |

# Elite vs. Low Performers

|                             | <i>Elite</i>                          | <i>Low</i>           | <i>Difference</i> |
|-----------------------------|---------------------------------------|----------------------|-------------------|
| <i>Deployment Frequency</i> | On-demand<br>(multiple times per day) | Monthly or quarterly | 208x              |
| <i>Deployment Lead Time</i> | < 1 hour                              | 1 week to 1 month    | 106x              |

# Elite vs. Low Performers

|                             | <i>Elite</i>                          | <i>Low</i>           | <i>Difference</i> |
|-----------------------------|---------------------------------------|----------------------|-------------------|
| <i>Deployment Frequency</i> | On-demand<br>(multiple times per day) | Monthly or quarterly | 208x              |
| <i>Deployment Lead Time</i> | < 1 hour                              | 1 week to 1 month    | 106x              |
| <i>Deploy Failure Rate</i>  | 0-15%                                 | 46-60%               | 7x                |

# Elite vs. Low Performers

|                             | <b>Elite</b>                          | <b>Low</b>           | <b>Difference</b> |
|-----------------------------|---------------------------------------|----------------------|-------------------|
| <b>Deployment Frequency</b> | On-demand<br>(multiple times per day) | Monthly or quarterly | 208x              |
| <b>Deployment Lead Time</b> | < 1 hour                              | 1 week to 1 month    | 106x              |
| <b>Deploy Failure Rate</b>  | 0-15%                                 | 46-60%               | 7x                |
| <b>Mean Time to Restore</b> | < 1 hour                              | Less than one day    | 2,604x            |

# High Performers Are More Secure And Controlled

**2x**

less time spent  
remediating  
security issues

**29%**

more time spent  
on new work

# High Performers Win In The Marketplace

**2x**

more likely to exceed profitability, market share & productivity goals

**2x**

more likely to achieve organizational and mission goals, customer satisfaction, quantity & quality goals

# High Performers Win In The Marketplace

**2.2x**

higher employee  
Net Promoter Score

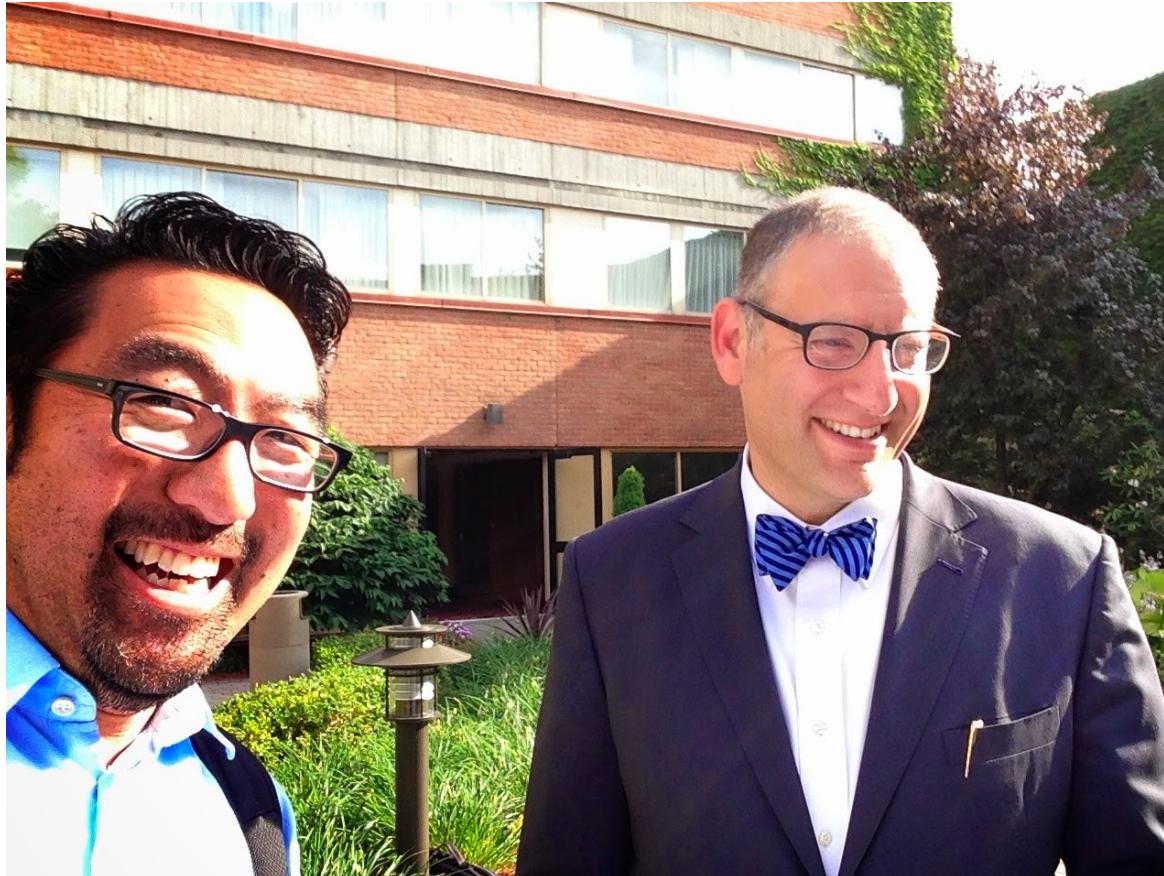
**50%**

higher market  
capitalization growth  
over 3 years\*

**When we can safely, quickly,  
reliably, securely achieve  
all the goals, dreams and  
aspirations of the organizations  
we serve...**

# **What I've Been Working On For The Last Three Years**

# 2014: Dr. Steve Spear at MIT Sloan



# The Book

- *Wiring the Winning Organization:*

*Liberating Our Collective Greatness Through Slowification, Simplification, and Amplification*

- Coming November 2023

LIBERATING OUR COLLECTIVE GREATNESS THROUGH  
SLOWIFICATION, SIMPLIFICATION, AND AMPLIFICATION

## Wiring the Winning Organization



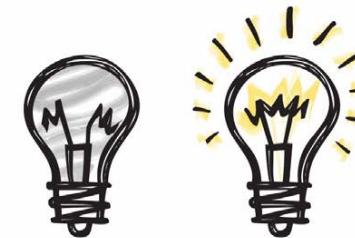
GENE KIM and  
STEVEN J. SPEAR

*Foreword by ADM John Richardson, US Navy (Retired)*  
*former Chief of Naval Operations*

- “This is a fabulous book that I highly recommend. Successful organizations flow from leaders who create the conditions in which many others thrive.”  
**—Paul Gaffney, former CTO and head of technology, The Home Depot, Kohl’s, Dick’s Sporting Good**
- “Kim and Spear lay out an amazing vision of the social circuitry for organizations to not only handle this, but thrive while doing so.”  
**—Phil Venables, Chief Information Security Officer, Google Cloud; former Board Director, Goldman Sachs Bank**
- “The authors not only provide an encompassing schema for those looking to drive performance by improving operations, but they help take practices developed for particular industries and functions and generalize them across all contexts. The guidance they provide is outstanding.”  
**— Dr. Joel Podolny, CEO, Honor Education, former VP & Dean of Apple University, former Dean of Yale School of Management**

LIBERATING OUR COLLECTIVE GREATNESS THROUGH  
SLOWIFICATION, SIMPLIFICATION, AND AMPLIFICATION

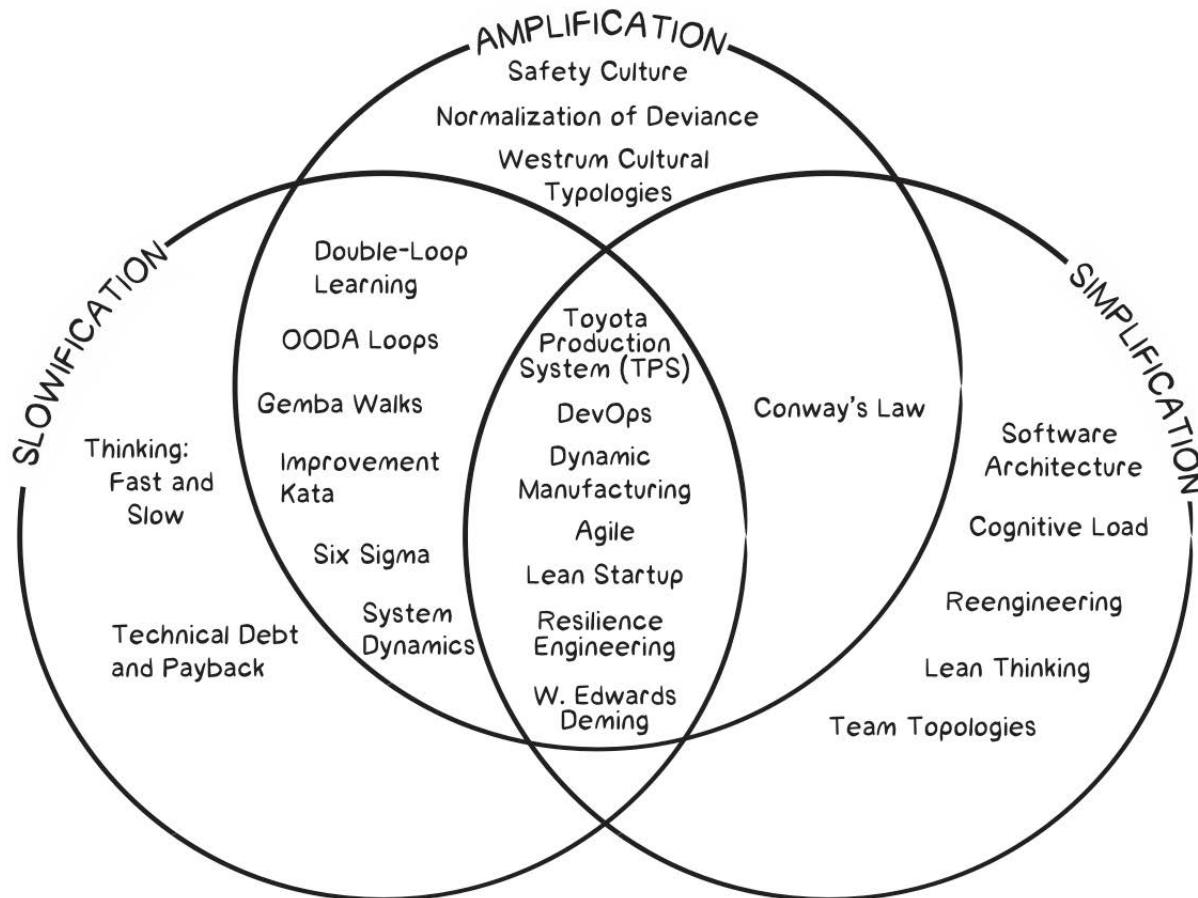
# Wiring the Winning Organization



GENE KIM and  
STEVEN J. SPEAR

*Foreword by* ADM John Richardson, US Navy (Retired)  
former Chief of Naval Operations

**“Explain the most amount of observable phenomena, using the fewest number of principles, confirm deeply held intuitions, reveal surprising insights”**



# The “Magic” That Winning Orgs Have

- Winning organizations do extraordinary things, more than any single individual could ever do alone, fully unleashing people's creativity and capabilities
- Versus organizations that constrain, or even extinguish entirely, the creativity and problem solving capabilities of people within them

# The “Magic” That Winning Orgs Have (1)

- Ideal
  - Everyone solving important problems, all the time, in parallel
  - Everyone has what they need (information, approvals, requirements, decision rights), in the right format, at the right time, interacting with all the right people
- Not ideal
  - Everyone is “stuck,” unable to do what needs to be done
  - No one has what they need, when they need it, in the right format, at the right time
  - Even small efforts require super-heroic efforts

# A Non-Ideal Example

- Present a checkbox to millions of customers to subscribe to a \$4.95/month for a third party service, which would get added to your monthly bill
  - 20+ teams
  - Requires near CEO-level support
  - \$28MM
  - 12 months
  - Projected 20% odds of success of hitting this deadline

# The “Magic” That Winning Orgs Have (2)

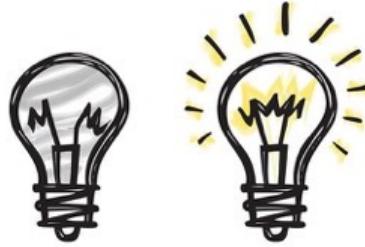
- Ideal
  - Energetic feedback loops where even weak failure signals are amplified and acted upon quickly.
  - People constantly learn from fast feedback generated on their work.
  - Feedback goes to the right people, at the right time, and in an actionable format.
- Not ideal
  - Feedback loops are weak, slow, nonexistent or go to the wrong people. Failures go undetected, and grow over time.
  - Signals are generated, but the system suppresses them or extinguishes them entirely.

# The “Magic” That Winning Orgs Have (3)

- Ideal
  - Enough time is schedule for planning, practice, experimentation, and improvement.
  - Practice environments can simulate performance conditions but with lower stakes, so teams can safely fail, learn, and improve.
  - Capturing lessons learned throughout the process to continually improve.
- Not ideal
  - Work must be done in high-stakes performance environments with no ability to redo, with high consequences of failures.
  - No time to pause in production, to slow down to speed up, to stop sawing to sharpen the saw.

LIBERATING OUR COLLECTIVE GREATNESS THROUGH  
SLOWIFICATION, SIMPLIFICATION, AND AMPLIFICATION

# Wiring the Winning Organization



GENE KIM and  
STEVEN J. SPEAR

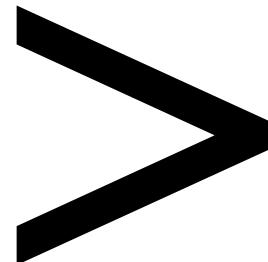
*Foreword by* ADM John Richardson, US Navy (Retired)  
former Chief of Naval Operations

# PARADOX: Winners and losers in level playing field contests



Worst to first

- Productivity
- Quality
- NPD time to market
- New brands
- New tech



First to worst

- Bail out
- Strikes
- Plant closures
- Cancelled brands
- Volt



# Google

- Opportunities
- Raw materials
- Capital equipment

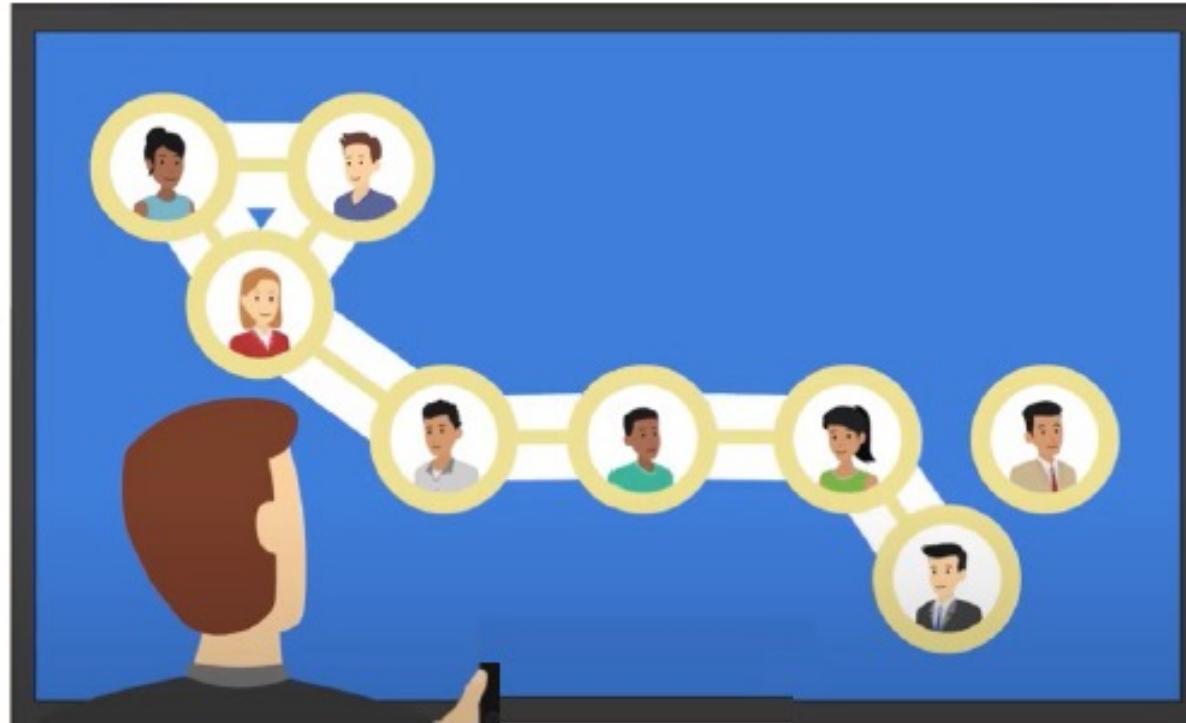
# yahoo!

- Science and technology
- IT systems
- Laws and regulations

# HARNESSING OUR INGENIETY

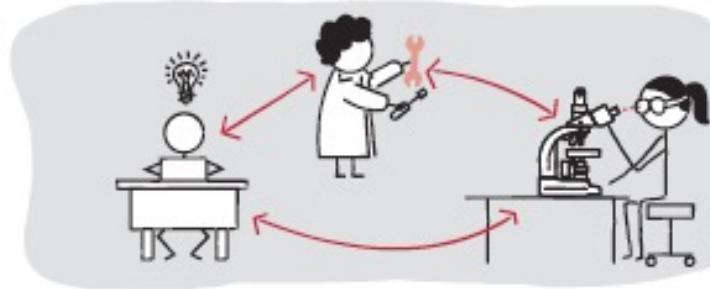
Answer: It's all about the brains

All else equal—commonality in resources and opportunities, then discommonality of outcomes must be due to discommonality in management systems. In particular, how easy they make for people to solve hard problems quicker and easier.



# HARNESSING INGENIETY

**LAYER 3**  
SOCIAL CIRCUITRY  
FOR FLOW OF IDEAS  
AND INFORMATION



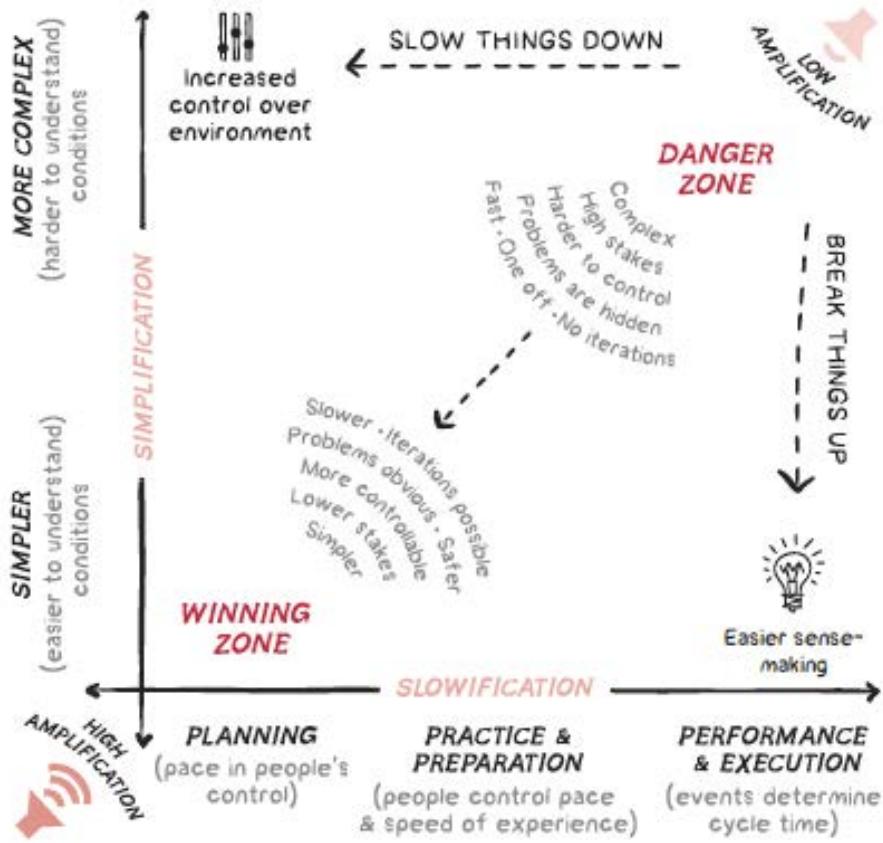
**LAYER 2**  
TOOLS AND  
INSTRUMENTATION



**LAYER 1**  
TECHNICAL OBJECT

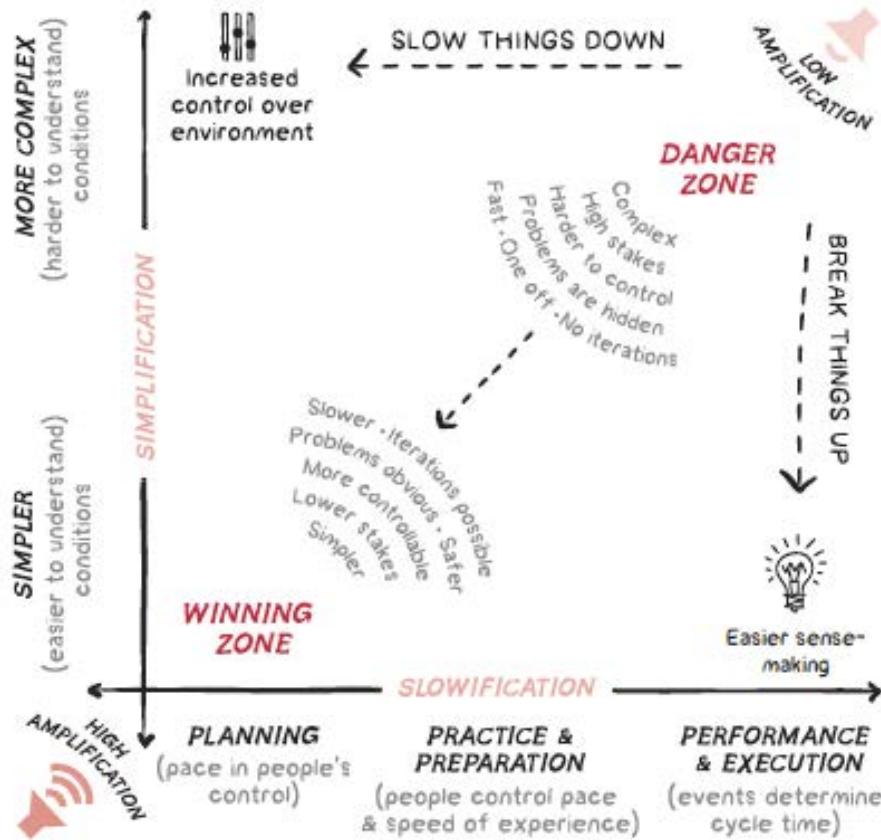


# “Danger zone” vs. “Winning Zone”



- Slowification
  - Make it easier to solve problem
- Simplification
  - Make problems easier to solve
- Amplification
  - Make it obvious you have problems to be solved.

# "Danger zone" vs. "Winning Zone"



## ■ Slowification

- Make it easier to solve problem

# SLOWIFICATION: STUDENT PILOT



**17 YEAR OLD PILOT  
STUDENT LANDS PLANE  
WITHOUT WHEELS**



awareness\_of\_sucess  
s

• Follow

...



awareness\_of\_success A 17-year-old student pilot successfully landed a plane that had lost a wheel Sunday, keeping calm under immense pressure.

Maggie Taraska had just taken off on her first solo flight in a single-engine Piper PA-28 Warrior when the plane's right main wheel fell off.

Taraska circled the Beverly Regional Airport in Massachusetts for the next half-hour as her flight instructor, John Singleton, and air traffic controllers coached her through what she needed to do.

Tag someone who should see this!



30,603 likes

JANUARY 4

Log in to like or comment.

# CLIP 1: Normal Operations



BVY Tower: Warrior 24 9'r 6 X-ray, runway nine, full length,  
cleared for takeoff, Warrior 24 9'r 6 X-ray.

## CLIP 2: Emergency Ops



# KEY POINT: SLOWIFICATION



Deliberative,  
generative,  
creative problem  
solving

Pre-wired habits,  
routines, instincts



# KEY POINT: SLOWIFICATION

Deliberative,  
generative,  
creative problem  
solving

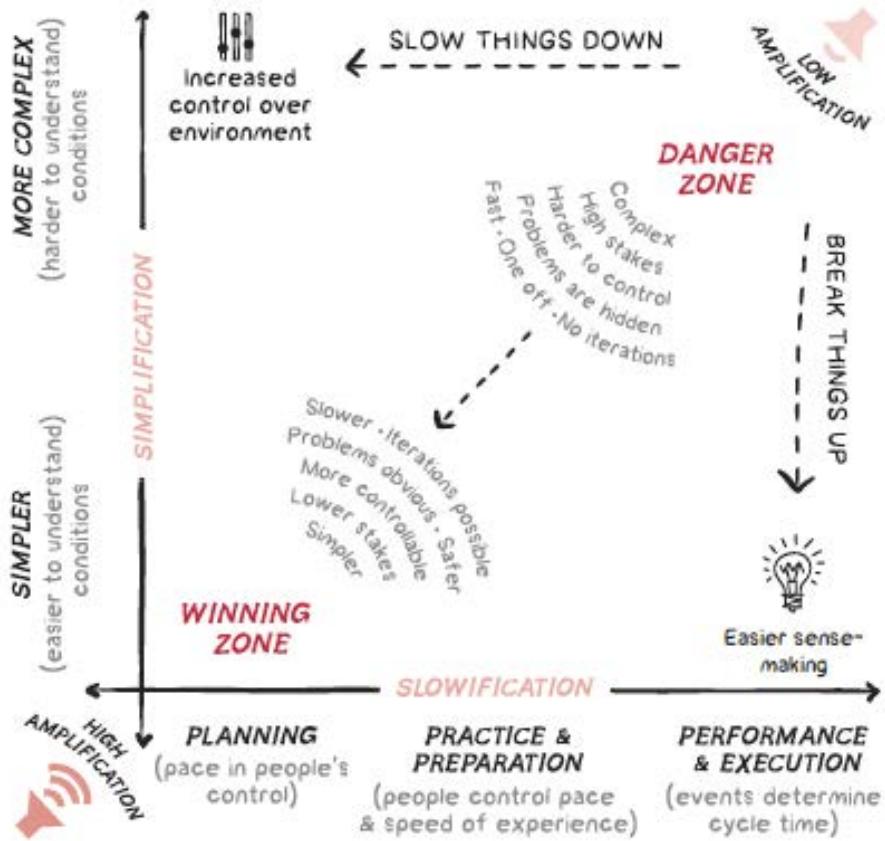
## OTHER OPPORTUNITIES

- In planning  
Models, prototypes, MVPs, sketches, "pin-ups", 'forceful backup', charetting, hack-a-thons
- In practice and preparation  
war-gaming, stress-testing, high tempo drills, dress rehearsals with audience
- In performance  
pauses, time-outs, andon cords, crew resource management, fast feedback

Pre-wired habits,  
routines, instincts



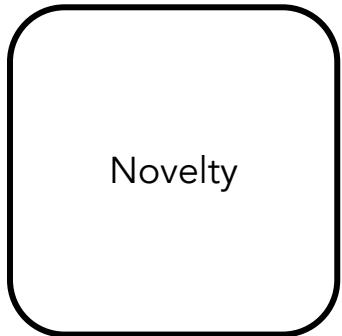
# “Danger zone” vs. “Winning Zone”



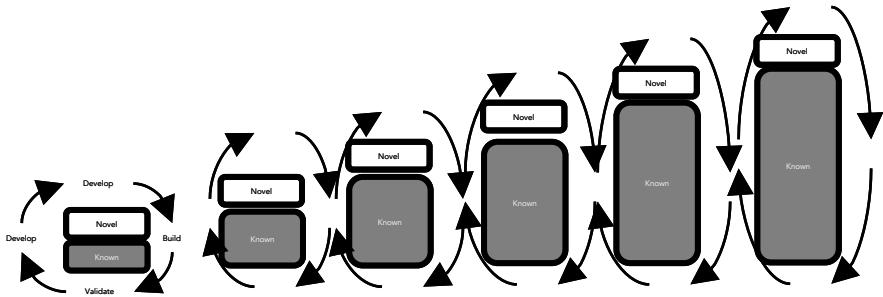
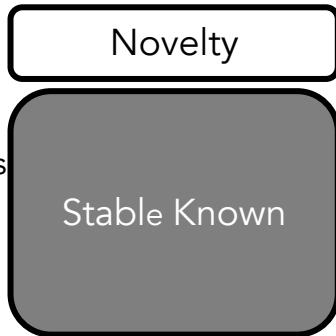
- Slowification
  - Make it easier to solve problem
- Simplification
  - Make problems easier to solve

# Simplification: Partitioning big problems into smaller more manageable pieces

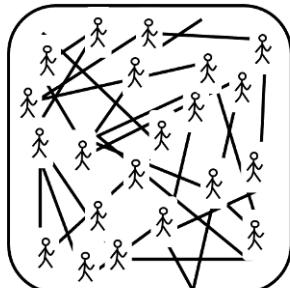
Incrementalize



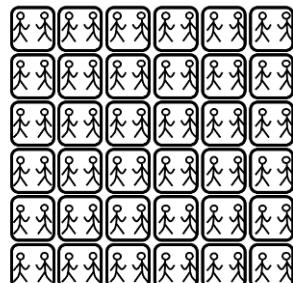
versus



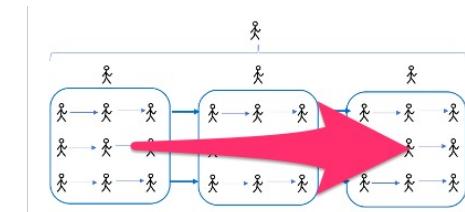
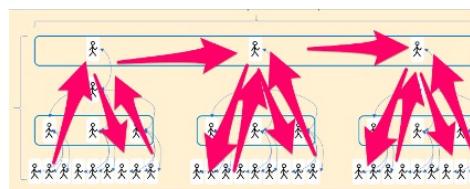
Modularize



versus



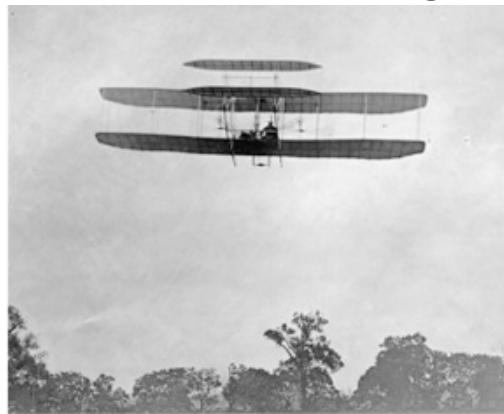
Linearize



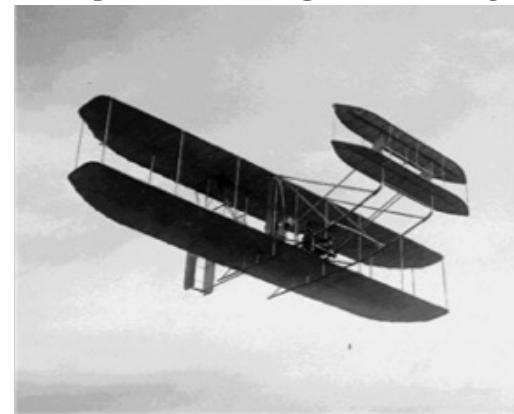
# SIMPLIFICATION BY INCREMENTALIZATION: WRIGHT BROTHERS



First flight 17 December 1903



Testing Flyer III, Ohio, 1905



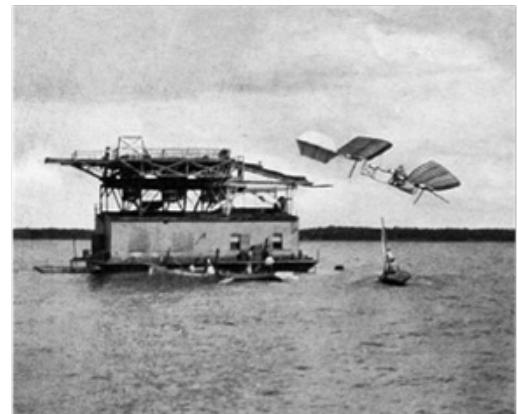
Wilber Wright in France, 1908



Air tunnel



Air foils



Langley Aerodome, 1903

# "Danger zone" vs. "Winning Zone"

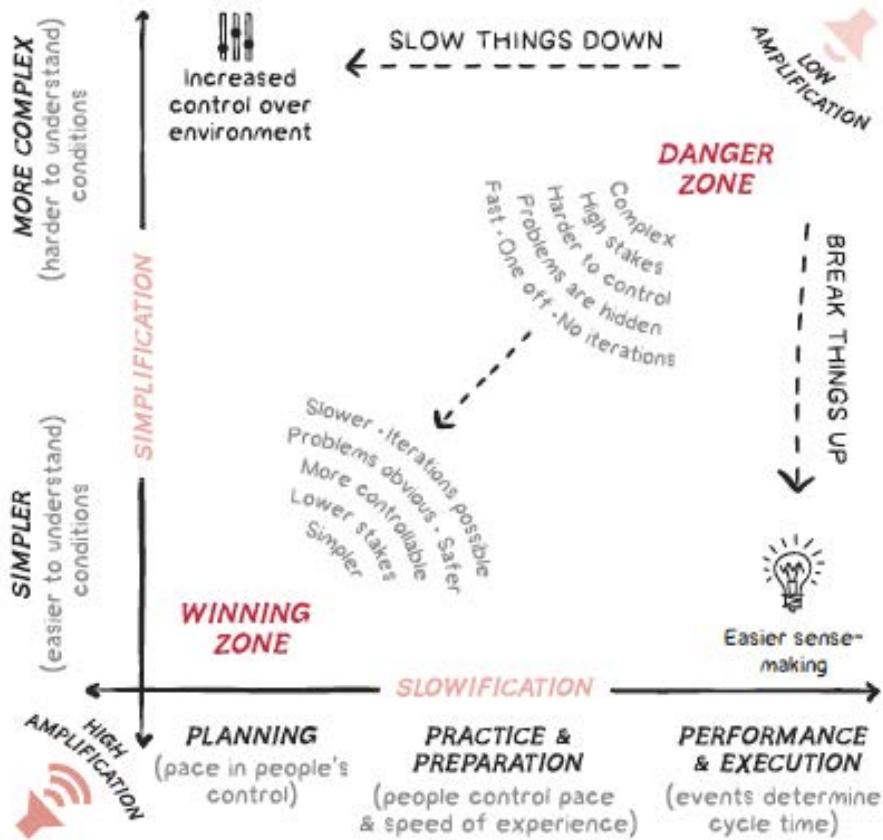


Big, complex problems with many intertwined factors

- Slowification
  - Make it easier to solve problem
- Simplification
  - Make problems easier to solve

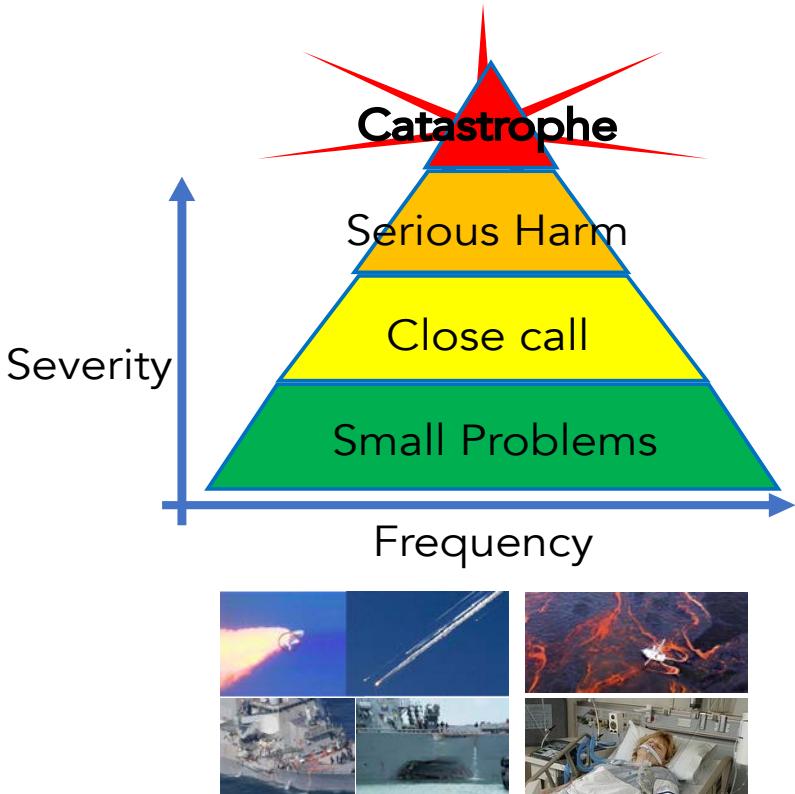
Big problems partitioned into smaller easier pieces to address

# "Danger zone" vs. "Winning Zone"



- Slowification
  - Make it easier to solve problem
- Simplification
  - Make problems easier to solve
- Amplification
  - Make it obvious you have problems to be solved.

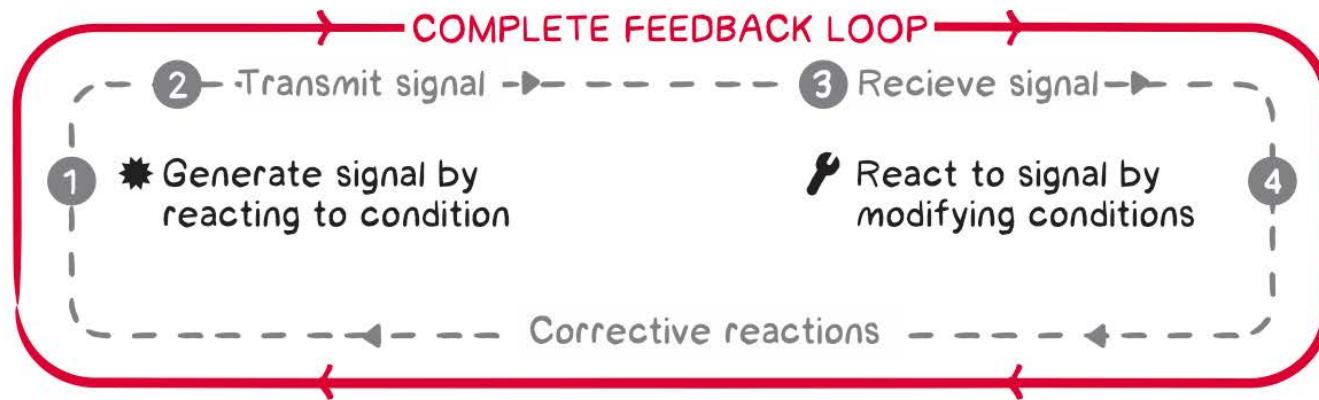
# Prioritizing maintaining operating tempo can nurture a culture of workarounds and waivers



- Pressure to maintain operational tempo causes small problems and missed requirements to be tolerated. Workarounds and even waivers become crutches.
- Leaders become insensitive to system vulnerabilities revealed by problems but masked by workarounds and waivers.
- Vulnerabilities accumulate to the point that the system fails gradually or catastrophically.
- Senior leaders foster workaround/waiver culture by not having effective system for reporting problems early and often...

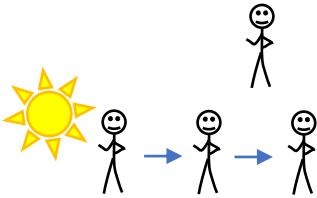
# AMPLIFICATION

**FIGURE 10.4** Amplification of Problems through Feedback Loop



# AMPLIFICATION: TOYOTA MOTOR MANUFACTURING

0: Define perfect



1: See imperfect where and when it occurs



2: Swarm to contain, investigate, and solve



3: Escalate for support



4: Leader's job: teaching, coaching, and modeling the necessary behaviors.

Toyota Motor Manufacturing, Texas (TMMTX)

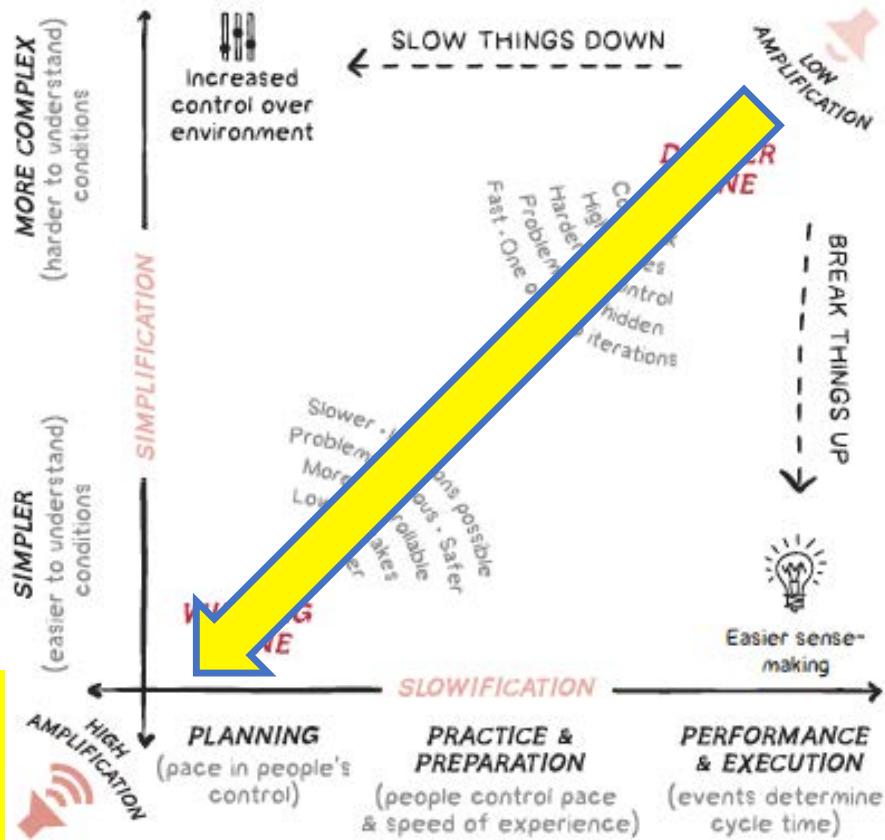


A new truck rolls off the line every 60 seconds at Toyota Motor Manufacturing, Texas (TMMTX). This efficiency and speed are a result of hard work and dedication from "Team Texas" — comprised of 3,800+ employees who assemble Tundras and Sequoias, as well as the more than 20 on-site suppliers that employ an additional 5,600 people. TMMTX broke ground in San Antonio in 2003 and began operations in 2006. The automaker purchased 2,678 acres of ranchland, with a legacy dating back to 1794. To establish greenspace along Leon Creek and the Medina River, which borders the property on two sides, Toyota donated 678 acres of land to the city.



**KEVIN P. VOELKEL**  
PRESIDENT

# "Danger zone" vs. "Winning Zone"



- Slowification
  - Make it easier to solve problem
- Simplification
  - Make problems easier to solve
- Amplification
  - Make it obvious you have problems to be solved.

# **Intro 3: Gene (15m)**

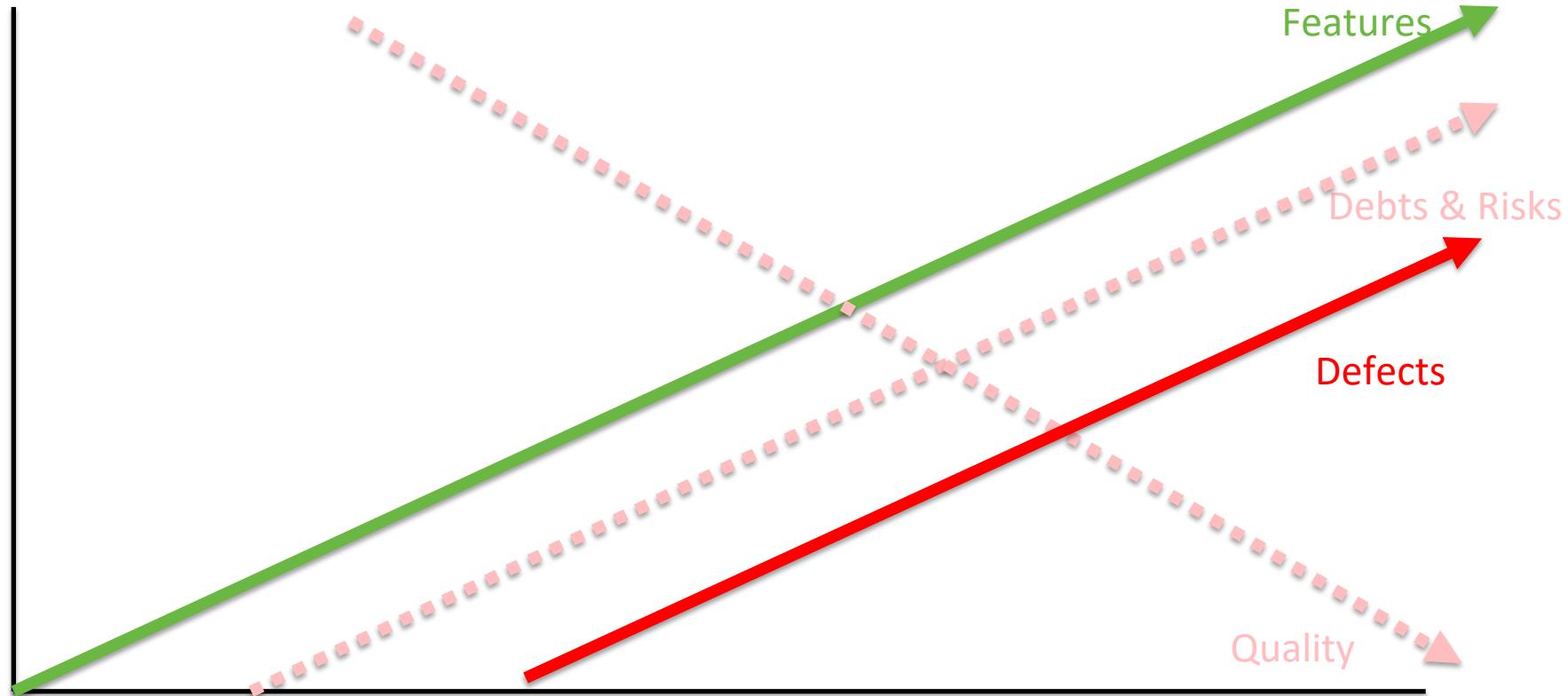
# Examples Of DevOps Practices

- Slowification
  - Pausing to pay down technical debt
  - Drills and injecting faults into the production environment
- Simplification
  - Incrementalization: Agile, incremental delivery
  - Modularization: service oriented architectures, APIs
  - Linearization: automation, continuous integration/delivery
- Amplification
  - Fast and frequent feedback in all aspects of our work
  - Blameless post-mortems
  - Resilience engineering and safety culture

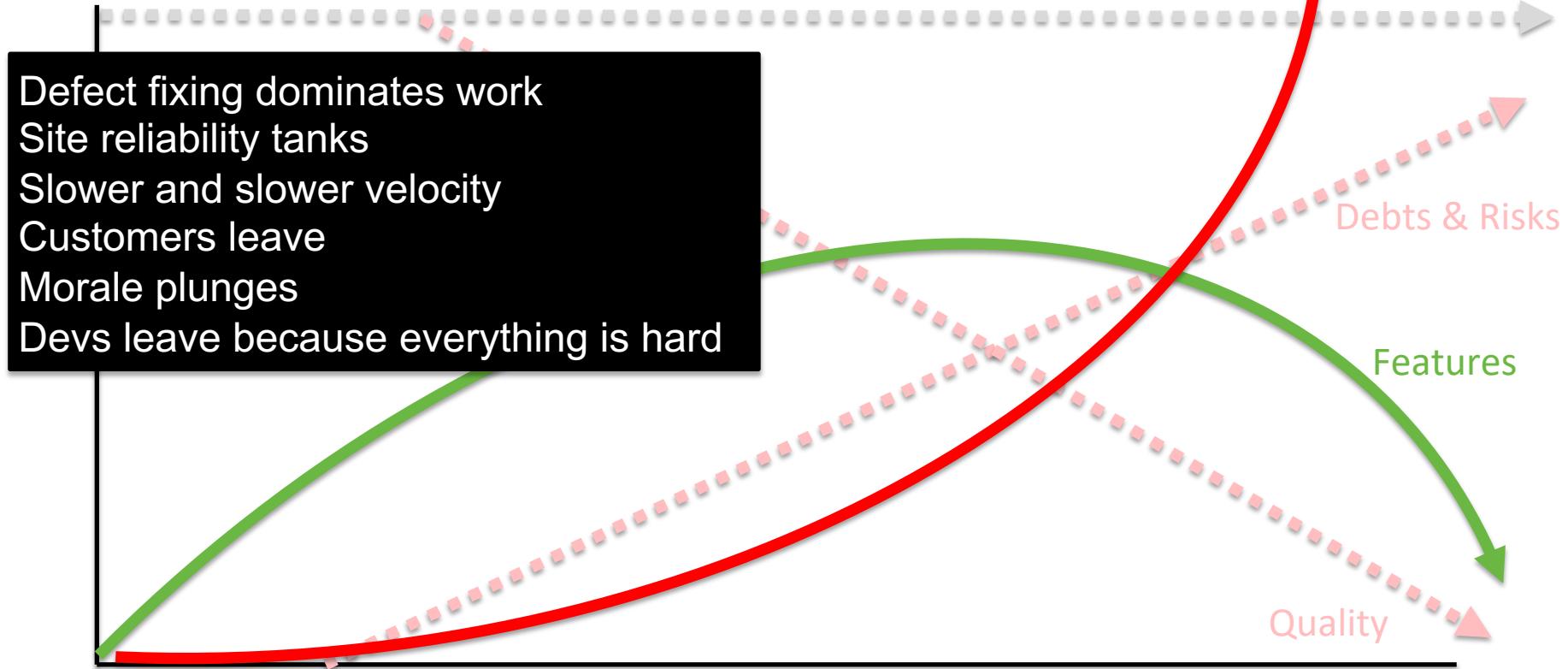
# **Slowification**

# **Greatness Isn't Free... The Need To Pay Down Technical Debt**

# Fast Push To Market



# Fast Push To Market — Continued



Who hasn't felt this?

You hire a bunch of developers, but you still can't ship the features you promised...

...and maybe you even have the feeling that things are slowing down...



John Cutler  
@johncutlefish

Follow

Case in point (from actual org)

- \* In 2015 reference feature took 15-30d.
- \* In 2018 same (class of) feature took 150-300d

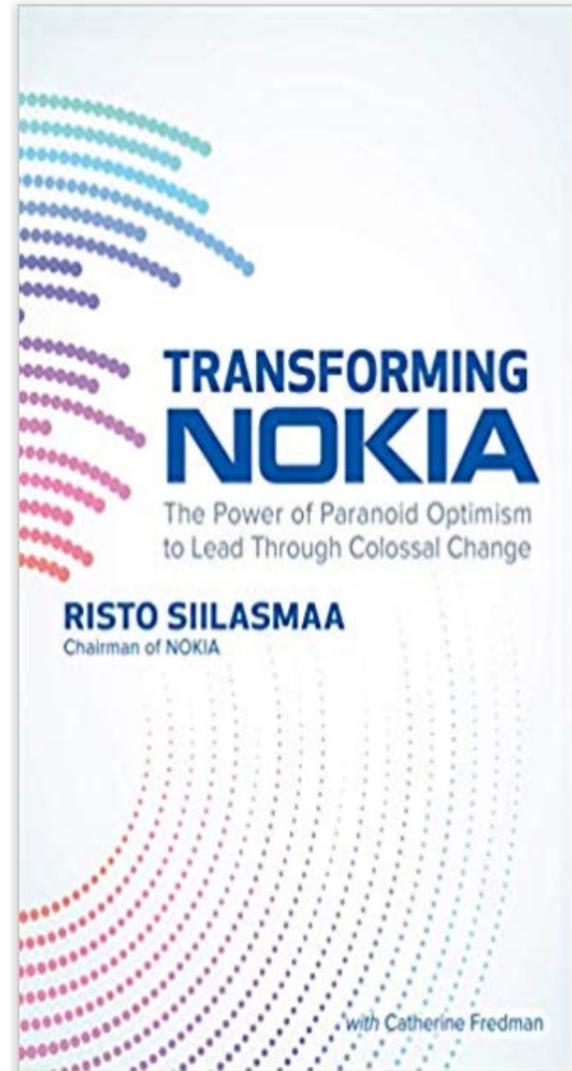
primarily bc of 1) tech debt, and 2) fast track silver bullets to drive success theater and/or acquisitions (for same effect)

Cc: [@realgenekim](#) [@mik\\_kersten](#)

# Risto Siilasmaa, NOKIA

“In 2010, Risto Siilasmaa was a board director at Nokia. When he learned that generating a Symbian build took *a whole forty-eight hours*, he said that it felt like someone hit him in the head with a sledgehammer,” Erik says. “He knew that if it took two days for anyone to determine whether a change worked or would have to be redone, there was a fundamental and fatal flaw in their architecture that doomed their near-term profitability and long-term viability. They could have had twenty times more developers, and it wouldn’t have made them go any faster.

Erik pauses. “It’s incredible. Sensei Siilasmaa knew that all the hopes and promises made by the engineering organization was a mirage. Even though there were numerous internal efforts to migrate off of Symbian, it was always shot down by the top executives until it was too late.



# Near Death Experiences

- Ebay (1999)
- Microsoft (2002): Bill Gates memo
- Google (2005): Automated testing culture
- Amazon (2004): Jeff Bezos memo
- Twitter (2008)
- LinkedIn (2009)
- Etsy (2009)

# 2002 Microsoft Security Standdown

- Famously, Microsoft after SQL Slammer required every product group to freeze feature

## BILL GATES: TRUSTWORTHY COMPUTING

In the past, we've made our software and services more compelling for users by adding new features and functionality, and by making our platform richly extensible. We've done a terrific job at that, but all those great features won't matter unless customers trust our software.

So now, when we face a choice between adding features and resolving security issues, we need to choose security. Our products should emphasize security right out of the box, and we must constantly refine and improve that security as threats evolve. A good example of this is the changes we made in Outlook to avoid e-mail-borne viruses. If we discover a risk that a feature could compromise someone's privacy, that problem gets solved first. If there is any way we can better protect important data and minimize downtime, we should focus on this. These principles should apply at every stage of the development cycle of every kind of software we create, from operating systems and desktop applications to global Web services.

# 50% Of R&D On Platforms!

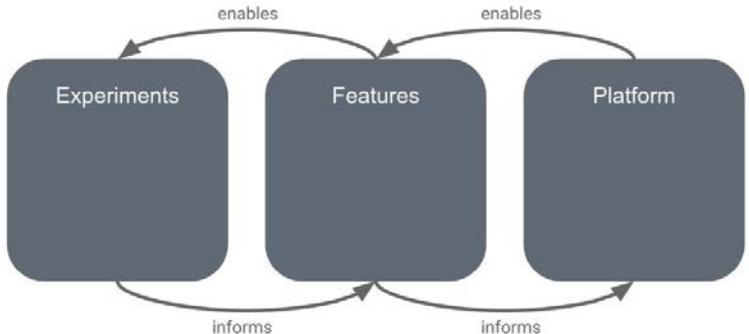
- Jean-Michel Lemieux (formerly VP Engr, Atlassian; former CTO, Shopify)
- “50% of R&D cycles should be spent on platforms”
  - 40% on features
  - 10% on experiments

Jean-Michel Lemieux  @jmwind · Dec 14, 2021 ...

10/ A way of visualizing your r&d investments is with 3 buckets. Having the right mix is critical. They work together.

A healthy investment distribution is 50% platform, 40% features, and 10% experiments.

You can skew for short periods, but over the long term its fatal.

$$\text{R\&D Velocity} = (\text{shipped experiments}) * (\text{shipped features})^2 * (\text{shipped infrastructure})^{10}$$


```
graph TD; Ex[Experiments] -- enables --> Fe[Features]; Fe -- enables --> Pl[Platform]; Pl -- informs --> Ex; Pl -- informs --> Fe;
```

2     14     122

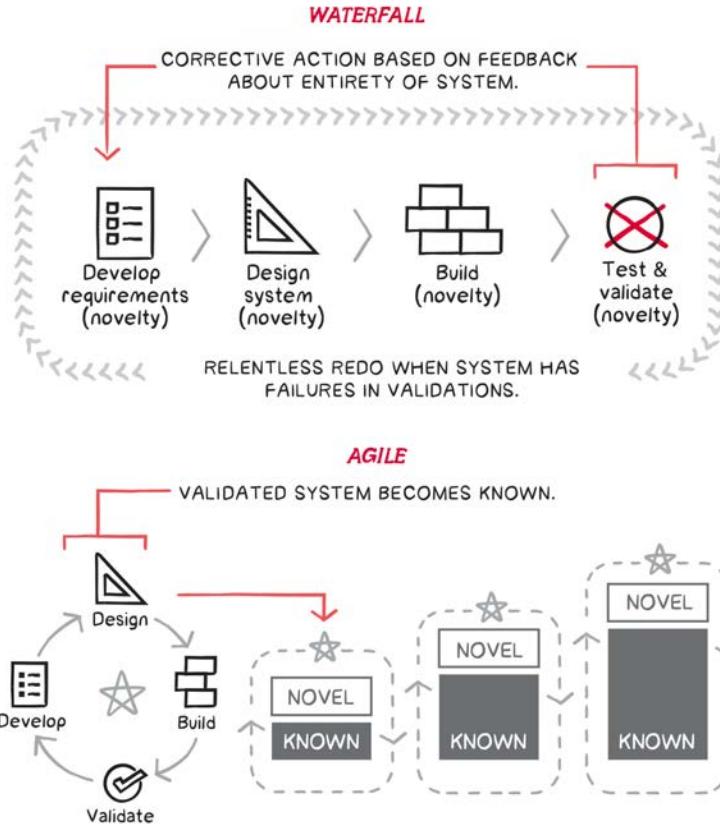
# **Simplification**

# Examples Of DevOps Practices

- Simplification
  - Incrementalization: Agile, incremental delivery
  - Modularization: service oriented architectures, APIs
  - Linearization: automation, continuous integration/delivery

# Incrementalization

FIGURE 7.3 Contrasting Waterfall Approaches with Incremental (Agile) Ones



# Modularization: It's Architecture!

- ...make large scale changes to the design of its system without the permission of someone outside the team, or depending on other teams
- ...complete its work without fine-grained communication and coordination with people outside the team
- ...deploy and release its product or service on demand, independently of other services the product or service depends upon
- ...do most of its testing on demand, without requiring an integrated test environment
- ...perform deployments during normal business hours with negligible downtime

# Amazon 2004

- “Amazon.com started 10 years ago as a monolithic application, running on a Web server, talking to a database on the back end. This application, dubbed Obidos, evolved to hold all the business logic, all the display logic, and all the functionality that Amazon eventually became famous for: similarities, recommendations, Listmania, reviews, etc.”
- “The many things that you would like to see happening in a good software environment couldn’t be done anymore; there were many complex pieces of software combined into a single system. It couldn’t evolve anymore. The parts that needed to scale independently were tied into sharing resources with other unknown code paths. There was no isolation and, as a result, no clear ownership”



Current Issue Past Issues Topics

## Interviews

June 30, 2006

Volume 4, issue 4



### A Conversation with Werner Vogels

Learning from the Amazon technology platform

Many think of Amazon as “that hugely successful online bookstore.” You would expect Amazon CTO Werner Vogels to embrace this distinction, but in fact it causes him some concern. “I think it’s important to realize that first and foremost Amazon is a technology company,” says Vogels. And he’s right. Over the past years, Vogels has helped Amazon grow from an online retailer (albeit one of the largest, with more than 55 million active customer accounts) into a platform on which more than 1 million active retail partners worldwide do business. Behind Amazon’s successful evolution from retailer to technology platform is its SOA (service-oriented architecture), which broke new technological ground and proved that SOAs can deliver on their promises.

Vogels came to Amazon from Cornell University, where he was working on high-availability systems and the management of scalable enterprise systems. He maintains that research spirit at Amazon, which regularly must solve problems never before encountered. “Maybe other companies call it research. We just call it

# The \$1 Billion Amazon API Rearchitecture

1. All teams will henceforth expose their data and functionality through service interfaces.
2. Teams must communicate with each other through these interfaces.
3. There will be no other form of interprocess communication allowed
4. It doesn't matter what technology you use, HTTP, Corba, Pubsub, Bezos doesn't care.
5. Service interfaces without exception must be designed from the ground up to be externalizable
6. Anybody who doesn't do this will be fired.
7. Thank you, have a nice day.

("#7 is obviously a joke, because obviously Bezos doesn't care whether you have a good day or not")

Who enforced this?

Amazon CIO: Rick Dalzell, a former U.S. Army Ranger

# Amazon Results

- 1999: thousands of deployments/year
- 2001: tens of deployments/year

# Amazon Results

- 1999: thousands of deployments/year
- 2001: tens of deployments/year
- 2011: 15K deployments/day
- 2015: 136K deployments/day

# Modularization

- Regain independence of action
- Avoid design-time coupling — so changes can be made on just one side of an interface (information hiding)
  - This is what makes technology migrations easier — enables large-scale changes behind the interface
  - Allows them to be feature-flagged
- Avoid run-time coupling — bad things stay local to the module

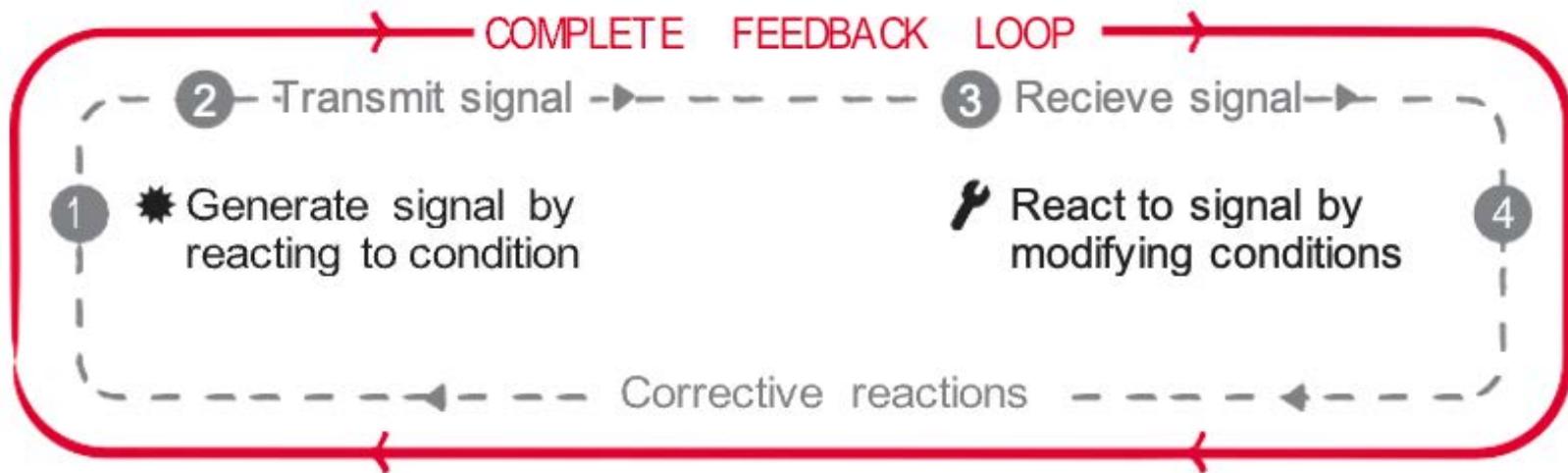
# Examples Of DevOps Practices

- Simplification
  - Incrementalization: Agile, incremental delivery
  - Modularization: service oriented architectures, APIs
  - Linearization: automation, continuous integration/delivery, other technical practices

# **Amplification**

# Examples Of DevOps Practices

- Amplification
  - Fast and frequent feedback in all aspects of our work
  - Blameless post-mortems
  - Resilience engineering and safety culture



**“We shape our buildings, and  
thereafter, they shape us.”**

**Winston Churchill**

# **Introduction Ends Here**

# **Exercise 1: Slowification**

## **Giving ourselves time to solve hard problems**

4: During execution, don't work around problems. Identify, diagnosis

- Identify
- Diagnosis
- Develop/propose/employ corrective action.



Avoid work around culture by which small recurring problems cause terrible tragedy.

*Range of capabilities*

*Agility Resilience*

*Amplified in practice*

*Honed in performance*

Dynamic Planning

(Re)Plan

Evaluate Practice

Stress Test

Improve/Train

Evaluate Practice

Perform

Practice several scenarios

**High velocity learning high velocity outcomes**

*Accelerated in planning*

Dynamic Planning

(Re)Plan

Evaluate

Practice

Dynamic Planning

(Re)Plan

Evaluate

Practice

Stress Test

Evaluate

Practice

Perform

Evaluate

Practice

Stress Test

Evaluate

Practice

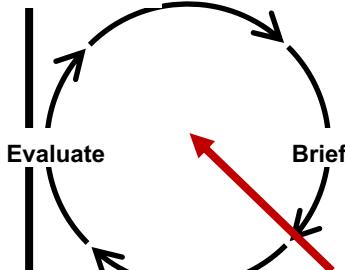
Perform

Practice several scenarios

2: Convert practice from rehearsal of few routines to mastering many.

- Avoid Air France tragedy of figuring out crisis 'on the fly'
-   many eventualities.

*Slow learning Re)Plan*



0: Failure modes of normal approach:

- Assuming planner can arrive at 'right' answer,
- Briefing is to get others to absorb right answer,
- Execute to the plan, and
- Evaluate the performers (not necessarily with planners)



1: Insert learning loops into planning. Repurpose it from 'developing the right answer' to finding faults in best known answers

- Have planner make declarations about expectations, hypotheses, etc.
- Expose 'best known answer' to stress testing, bona fide red teaming.
- Modify plan in response to exposed flaws in thinking.

Eg: Avoid the self-reinforcing/cold-aggrandizing behavior INJ admiralty in advance of Midway commitment to learning dynamics exposed before they

*Commitment to learning dynamics*

S. Spear Feb 14 2019

# Examples Of DevOps Practices

- Slowification
  - Pausing to pay down technical debt
  - Drills and injecting faults into the production environment

## Amazon EC2 outage downs Reddit, Quora



@SCVNNGR  
SCVNNGR  
The sky is falling! Amazon's cloud seems  
to be down (raining?) so we're  
experiencing some issues too. Be back  
soon!

5 hours ago via web

Retweeted by [RealAmandaStone](#) and others



SCVNNGR and other sites took to Twitter after a rare and major outage of Amazon's cloud-based Web service.

Recommend 990 people recommend this.

By Julianne Pepitone, staff reporter April 22, 2011: 7:29 AM ET

@RealGeneKim

# Inject Failures Often

## The Netflix Tech Blog

### 5 Lessons We've Learned Using AWS

We've sometimes referred to the Netflix software architecture in AWS as our Rambo Architecture. Each system has to be able to succeed, no matter what, even all on its own. We're designing each distributed system to expect and tolerate failure from other systems on which it depends.

One of the first systems our engineers built in AWS is called the Chaos Monkey. The Chaos Monkey's job is to randomly kill instances and services within our architecture. If we aren't constantly testing our ability to succeed despite failure, then it isn't likely to work when it matters most – in the event of an unexpected outage.

**You Don't Choose Chaos Monkey...  
Chaos Monkey Chooses You**



# The 2014 AWS Reboot

“When we got the news about the emergency EC2 reboots, our jaws dropped. When we got the list of how many Cassandra nodes would be affected, I felt ill.

“Then I remembered all the Chaos Monkey exercises we’ve gone through. My reaction was, ‘Bring it on!’”

– Christos Kalantzis  
Netflix Cloud DB Engineering

# The 2014 AWS Reboot

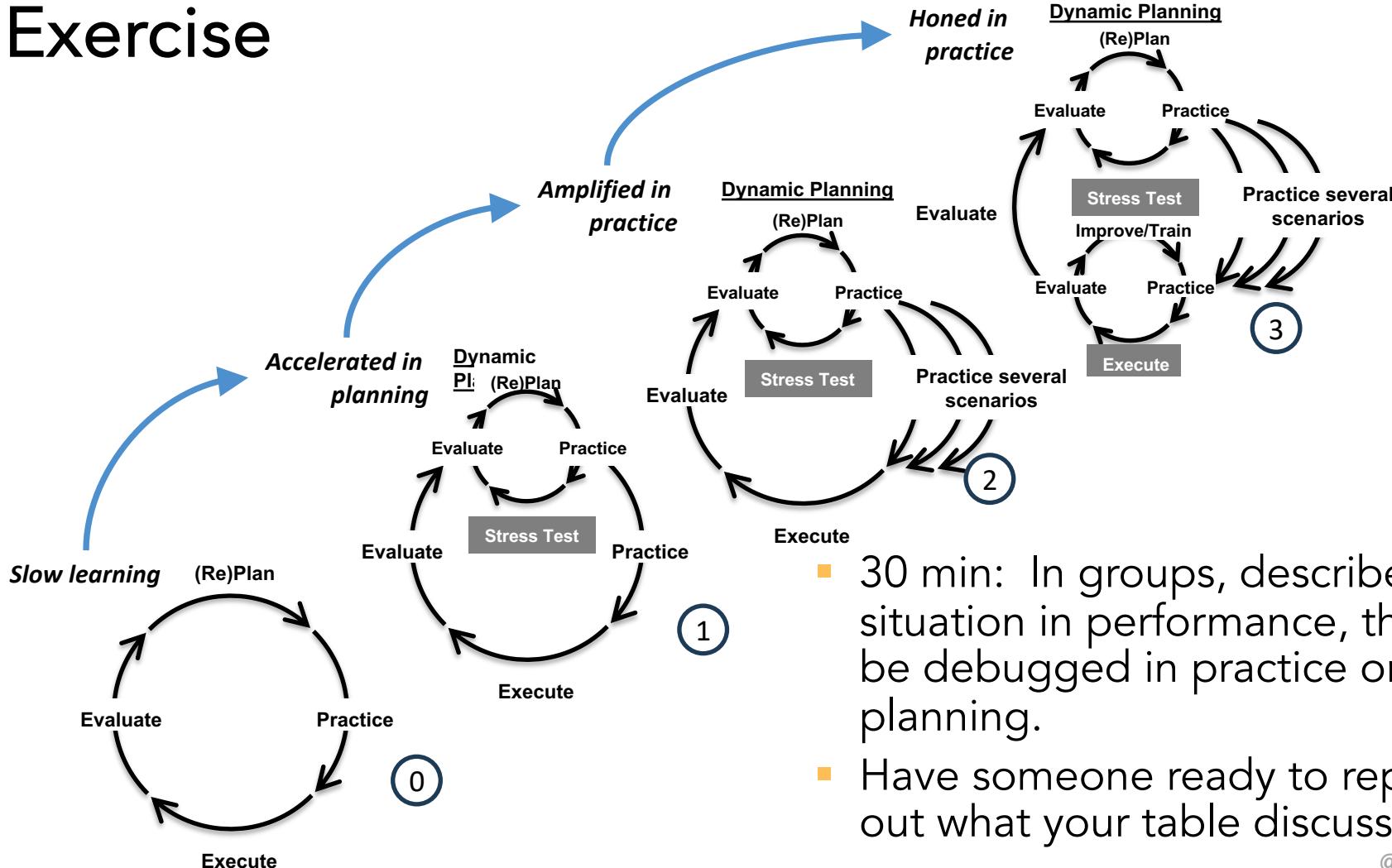
“Out of our 2700+ production Cassandra nodes, 218 were rebooted. 22 Cassandra nodes did not reboot successfully.

“Netflix customers experienced no downtime that weekend.”

– Bruce Wong  
Netflix Chaos Engineering

# Netflix and Service Catalog

# Exercise



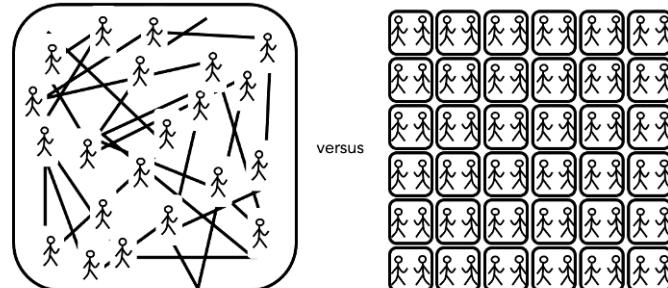
- 30 min: In groups, describe a situation in performance, that can be debugged in practice or planning.
- Have someone ready to report out what your table discussed

**Exercise 2: Simplification**  
**Making hard problems easier to  
solve by partitioning big complex  
things into smaller pieces.**

# Examples Of DevOps Practices

- Simplification
  - Incrementalization: Agile, incremental delivery
  - Modularization: service oriented architectures, APIs
  - Linearization: automation, continuous integration/delivery

Examples of Modularization:



# Architecture Enables Teams To...

- ...make large scale changes to the design of its system without the permission of someone outside the team, or depending on other teams
- ...complete its work without fine-grained communication and coordination with people outside the team
- ...deploy and release its product or service on demand, independently of other services the product or service depends upon
- ...do most of its testing on demand, without requiring an integrated test environment
- ...perform deployments during normal business hours with negligible downtime

# Amazon 1998

- Amazon tech stack
  - Netscape Server with Obidos plug-ins written in C++
  - Berkeley DB (for book listings)
  - Oracle DB (for e-commerce)
- E-commerce teams
  - Product page
  - Shopping cart
  - Ordering
  - Returns
  - Inventory
- Product team
  - Product listing for books

# Amazon 1998

Wednesday, October 13, 1999

SEARCH  
All Products  Search of the Day: saffron

WELCOME BOOKS MUSIC VIDEO TOYS & GAMES ELECTRONICS e-CARDS AUCTIONS zSHOPS  
HOW TO ORDER GIFT SERVICES OUR GUARANTEE SITE GUIDE COMMUNITY

WELCOME TO amazon.com

Hello! Shopping at Amazon.com is 100% secure--guaranteed.  
Already a customer? [Sign in](#).

Vote in our [Millennium Poll](#)--you could win 300 CDs, books, and videos!

In Books  
**Test Case**  
  
The postwar inventors of the Scholastic Aptitude Test hoped to produce a brainier brand of meritocracy in the United States. But as Nicholas Lemann reveals in [The Big Test](#), the SAT hit a great many ideological potholes--and ended up creating yet another, pencil-pushing elite. Go to [Books](#)

Tricks, treats, and costumes for all in our [Halloween Bootique](#).

**Amazon.com 100 Hot Books**  
**Updated Hourly**  
1. [The Carbohydrate Addict's Lifespan Program : A](#)

# Amazon 2002

amazon.com Ann's Store Books See All 32 Product Categories Your Account | Cart | Wish List | Help |

Advanced Search | Browse Subjects | Bestsellers | The New York Times® Best Sellers | Magazines | Corporate Accounts | Amazon Shorts | AmazonConnect | Bargain Books | Used Books | Textbooks

Search Books  Find Gifts

**Results for joseph stiglitz**

Related Searches: [stiglitz](#); [jeffrey sachs](#); [paul krugman](#)

**So You'd Like to... Offer your advice**

 **Customers who searched for joseph stiglitz ultimately chose:**

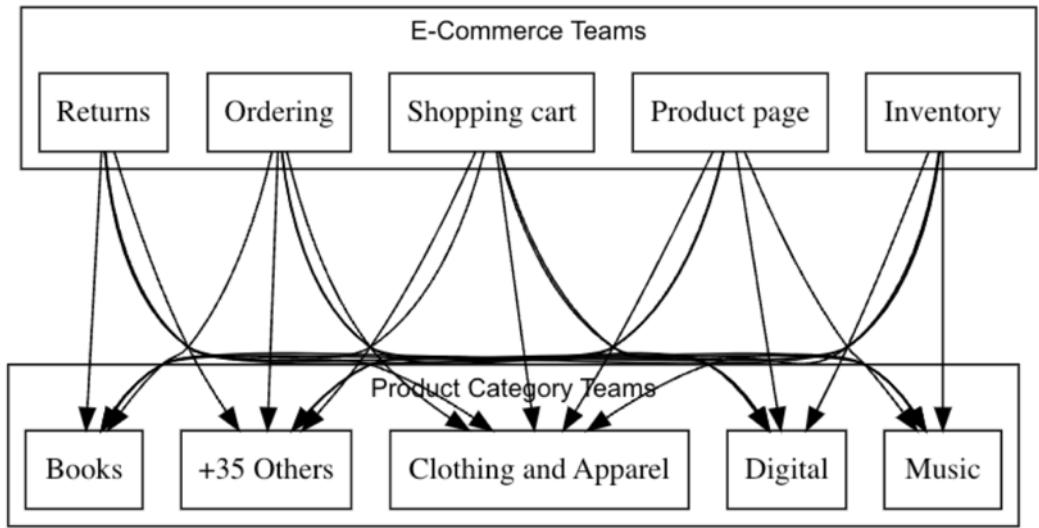
1.  [Globalization and Its Discontents](#) -- by Joseph E. Stiglitz; Paperback  
(Rate it)  
Buy new: \$10.37 -- [Used & new from: \\$8.65](#)

**Listmania!** [Create a Listmania! list](#)

 **College Years -**

# Amazon 2002

- E-commerce teams
  - Product page
  - Shopping cart
  - Ordering
  - Returns
  - Inventory
- Product teams
  - Books
  - Music
  - Digital
  - Apparel
  - +35 more...



# Amazon 2004

- With the introduction of digital products, Dr. Werner Vogels, CTO of Amazon since 2005, described how when those teams “wanted to add something to the order pipeline, a physical delivery address was required.

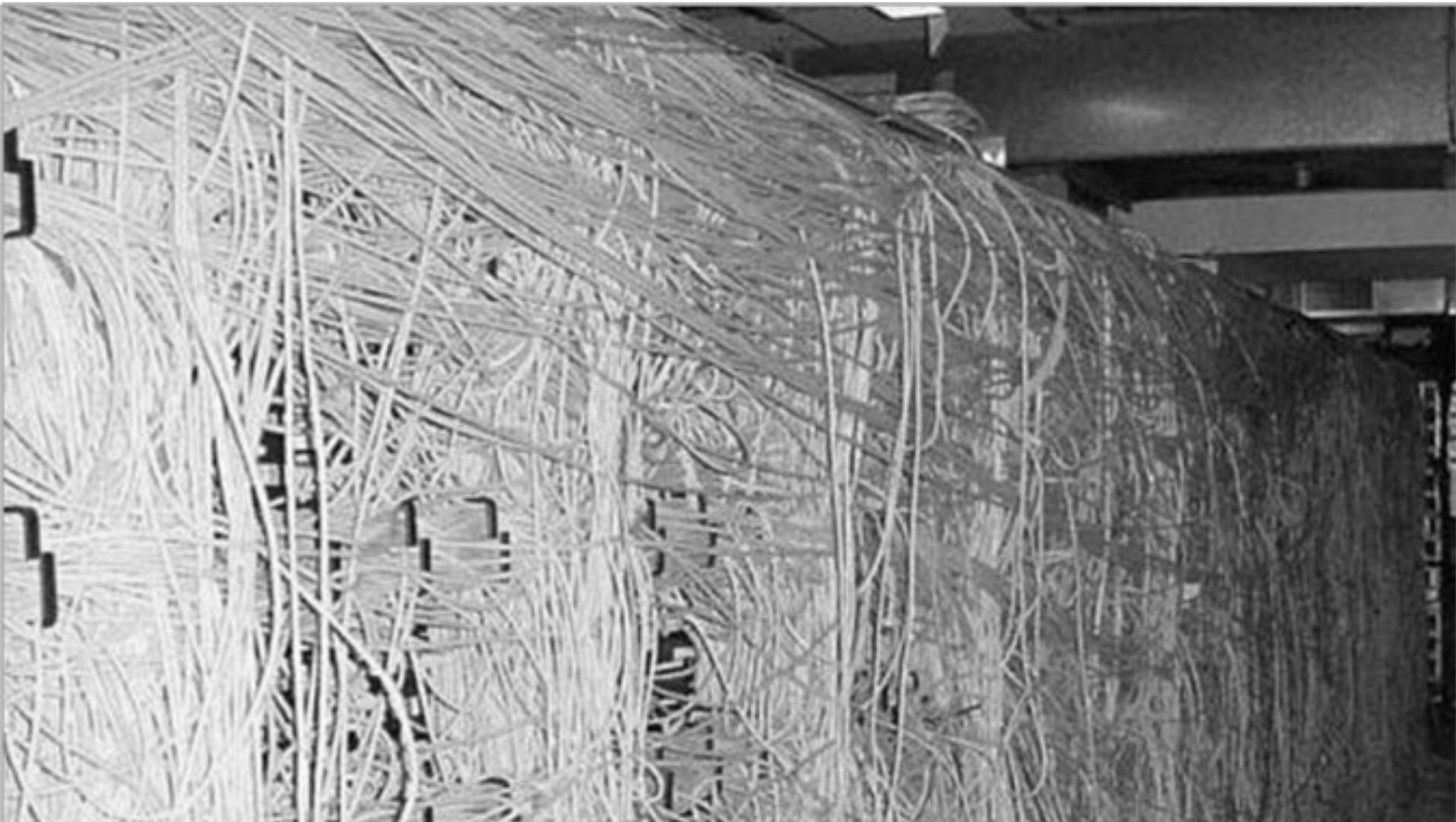
“There was no way around [not providing a physical delivery address]. They would walk to the 80 different ordering teams and say, ‘We need to change this.’ The ordering teams would respond that they hadn’t budgeted for it.” So now those teams were stuck, unable to ship products.

# Amazon 2002

- With the introduction of digital products, Dr. Werner Vogels, CTO of Amazon since 2005, described how when those teams “wanted to add something to the order pipeline, a physical delivery address was required.

“There was no way around [not providing a physical delivery address]. They would walk to the 80 different ordering teams and say, ‘We need to change this.’ The ordering teams would respond that they hadn’t budgeted for it.” So now those teams were stuck, unable to ship products.





# Amazon 2002: Clothing and Apparel

- In the early days of Amazon, Jeff Bezos, founder and then CEO, described how wanted all new hires to be “doers—engineers, developers, perhaps merchandise buyers, but not managers.”
- In 2002, Bezos comes back and issues the most sweeping edict: organize around two-pizza teams, so that they can “independently work on Amazon’s biggest problem”
- “Communicate less, not more”

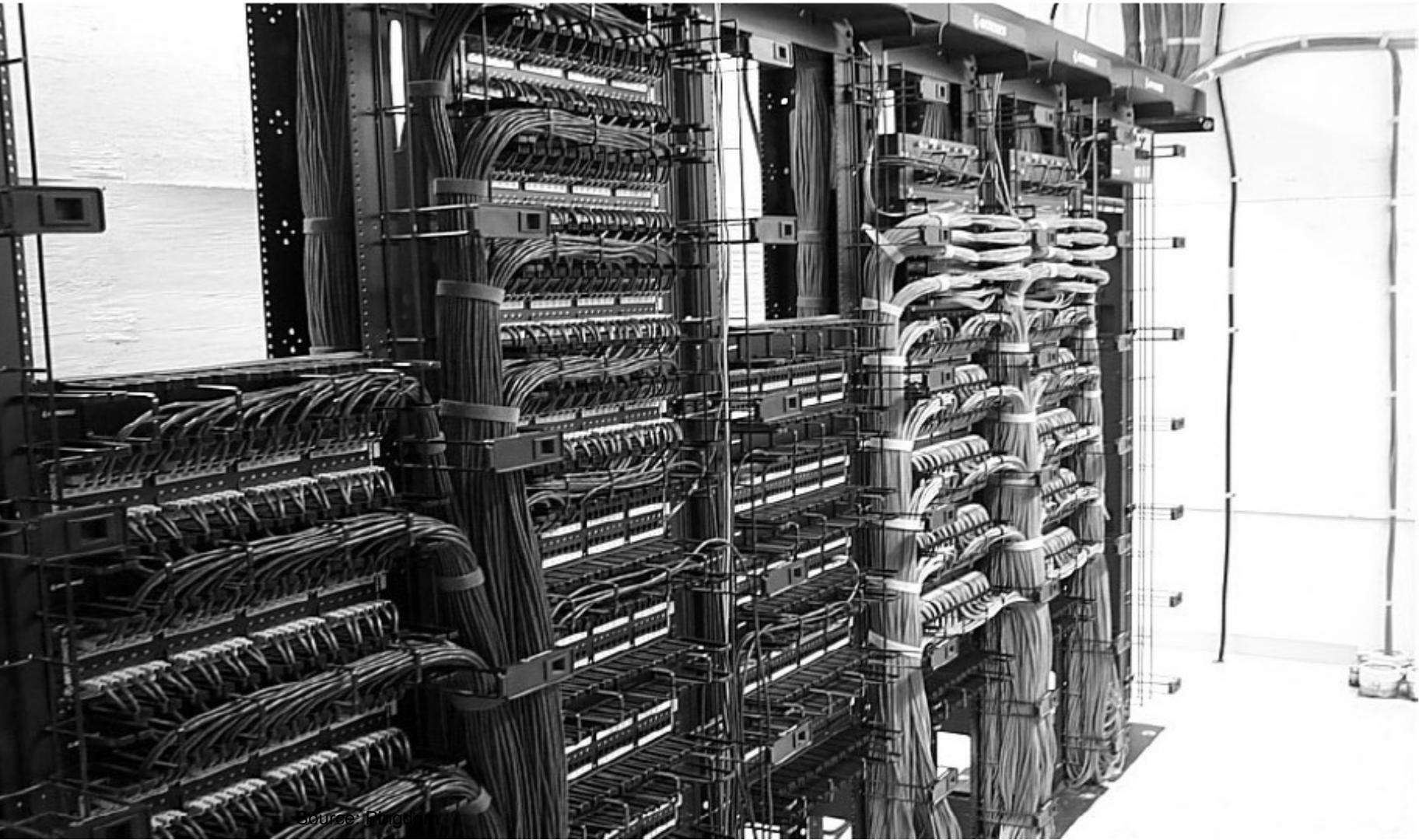
# The \$1 Billion Amazon API Rearchitecture

1. All teams will henceforth expose their data and functionality through service interfaces.
2. Teams must communicate with each other through these interfaces.
3. There will be no other form of interprocess communication allowed
4. It doesn't matter what technology you use, HTTP, Corba, Pubsub, Bezos doesn't care.
5. Service interfaces without exception must be designed from the ground up to be externalizable
6. Anybody who doesn't do this will be fired.
7. Thank you, have a nice day.

("#7 is obviously a joke, because obviously Bezos doesn't care whether you have a good day or not")

Who enforced this?

Amazon CIO: Rick Dalzell, a former U.S. Army Ranger



Source: Pingdom

im



# Amazon Results

- 1999: thousands of deployments/year
- 2001: tens of deployments/year

# Amazon Results

- 1999: thousands of deployments/year
- 2001: tens of deployments/year
- 2011: 15K deployments/day
- 2015: 136K deployments/day

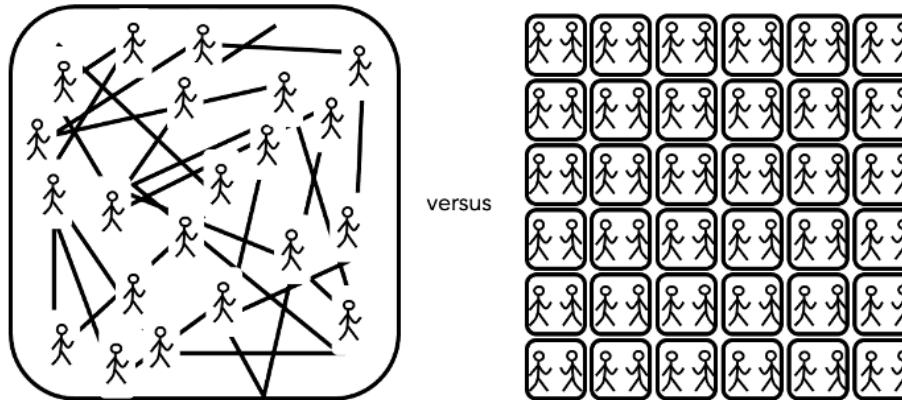
# Architecture Enables Teams To...

- ...make large scale changes to the design of its system without the permission of someone outside the team, or depending on other teams
- ...complete its work without fine-grained communication and coordination with people outside the team
- ...deploy and release its product or service on demand, independently of other services the product or service depends upon
- ...do most of its testing on demand, without requiring an integrated test environment
- ...perform deployments during normal business hours with negligible downtime

# Modularization

- Regain independence of action
- Avoid design-time coupling — so changes can be made on just one side of an interface (information hiding)
  - This is what makes technology migrations easier — enables large-scale changes behind the interface
  - Allows them to be feature-flagged
- Avoid run-time coupling — bad things stay local to the module

# Exercise: Simplification by modularization



- 30 min: In groups, describe situations where solving a problem happened in too large of a batch — that could have benefited from breaking the problem into many more, smaller simpler pieces.
- Choose someone to be ready to report out to the rest of the room on what your table discussed

# **Exercise 3: Amplification**

## **Steve (15m)**

# Steve Outline

- Workaround culture

# **Exercise 3: Amplification Gene (5m)**

# Examples Of DevOps Practices

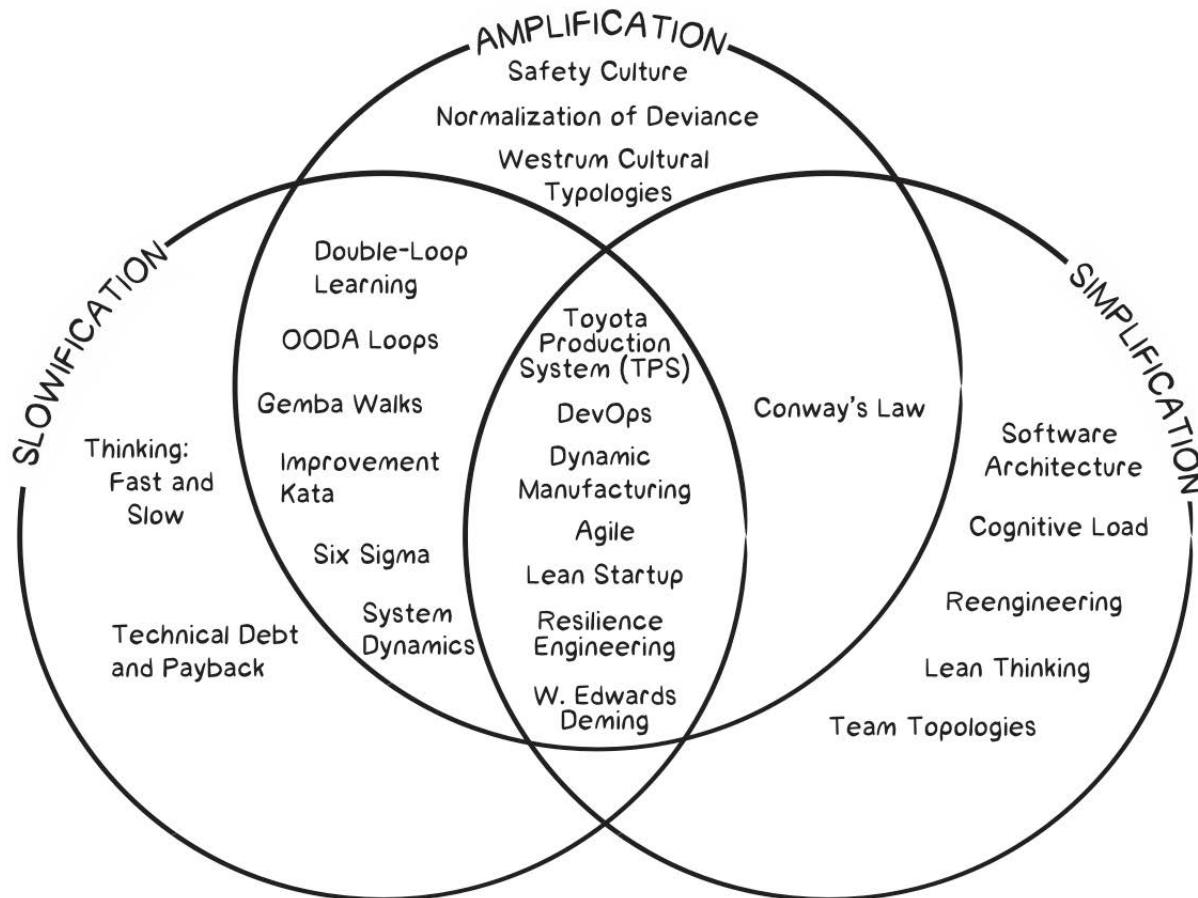
- Amplification
  - Fast and frequent feedback in all aspects of our work
  - Blameless post-mortems
  - Resilience engineering and safety culture
  - Randy Shoup's story of post-incident reviews at Google

# Exercise

- 30 min: In your groups, describe situations where signals of problems were not adequately generated, transmitted, received, reacted to, and the solution not validated.
- Choose someone to be ready to report out to the rest of the room on what your table discussed

# **Closing**

**“Explain the most amount of observable phenomena, using the fewest number of principles, confirm deeply held intuitions, reveal surprising insights”**



# The “Magic” That Winning Orgs Have

- Winning organizations do extraordinary things, more than any single individual could ever do alone, fully unleashing people's creativity and capabilities
- Versus organizations that constrain, or even extinguish entirely, the creativity and problem solving capabilities of people within them

# The “Magic” That Winning Orgs Have (1)

- Ideal
  - Everyone solving important problems, all the time, in parallel
  - Everyone has what they need (information, approvals, requirements, decision rights), in the right format, at the right time, interacting with all the right people
- Not ideal
  - Everyone is “stuck,” unable to do what needs to be done
  - No one has what they need, when they need it, in the right format, at the right time
  - Even small efforts require super-heroic efforts

# A Non-Ideal Example

- Present a checkbox to millions of customers to subscribe to a \$4.95/month for a third party service, which would get added to your monthly bill
  - 20+ teams
  - Requires near CEO-level support
  - \$28MM
  - 12 months
  - Projected 20% odds of success of hitting this deadline

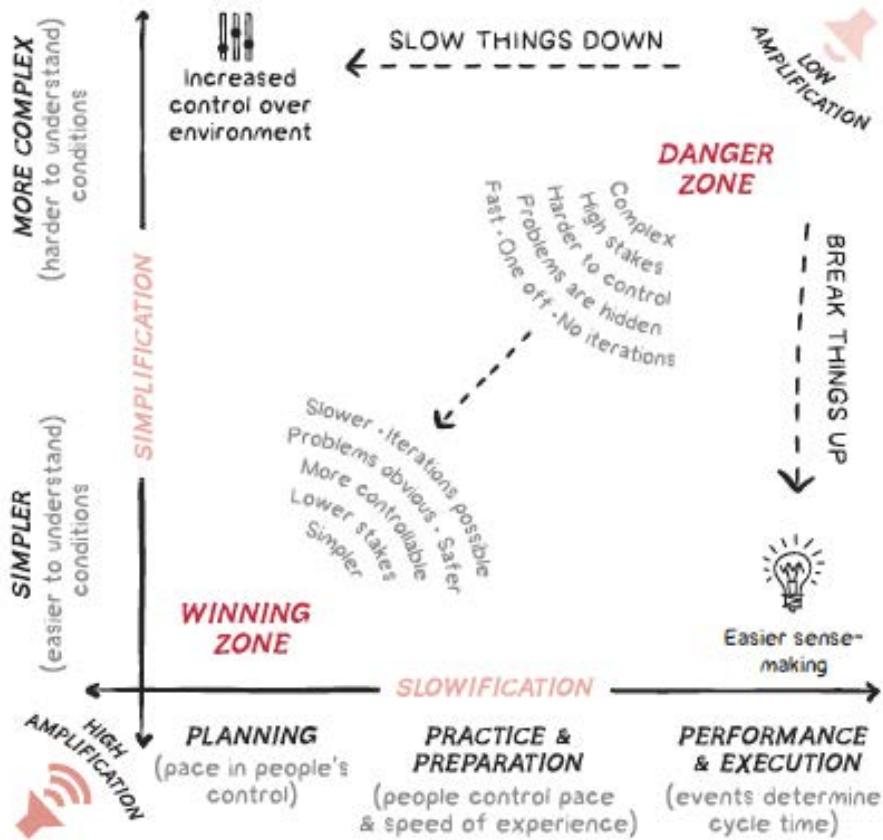
# The “Magic” That Winning Orgs Have (2)

- Ideal
  - Energetic feedback loops where even weak failure signals are amplified and acted upon quickly.
  - People constantly learn from fast feedback generated on their work.
  - Feedback goes to the right people, at the right time, and in an actionable format.
- Not ideal
  - Feedback loops are weak, slow, nonexistent or go to the wrong people. Failures go undetected, and grow over time.
  - Signals are generated, but the system suppresses them or extinguishes them entirely.

# The “Magic” That Winning Orgs Have (3)

- Ideal
  - Enough time is scheduled for planning, practice, experimentation, and improvement.
  - Practice environments can simulate performance conditions but with lower stakes, so teams can safely fail, learn, and improve.
  - Capturing lessons learned throughout the process to continually improve.
- Not ideal
  - Work must be done in high-stakes performance environments with no ability to redo, with high consequences of failures.
  - No time to pause in production, to slow down to speed up, to stop sawing to sharpen the saw.

# “Danger zone” vs. “Winning Zone”

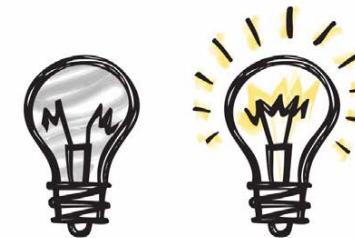


- **Slowification**
  - Solve problems not in production, but in pre-production
  - Pause the system, if necessary
- **Simplification**
  - Smaller coherent units that can work independently
  - Reduce coupling where you can
- **Amplification**
  - Reduce noise in the system, so you can amplify small problems, so you can treat them like large problems

- “This is a fabulous book that I highly recommend. Successful organizations flow from leaders who create the conditions in which many others thrive.”  
**—Paul Gaffney, former CTO and head of technology, The Home Depot, Kohl’s, Dick’s Sporting Good**
- “Kim and Spear lay out an amazing vision of the social circuitry for organizations to not only handle this, but thrive while doing so.”  
**—Phil Venables, Chief Information Security Officer, Google Cloud; former Board Director, Goldman Sachs Bank**
- “The authors not only provide an encompassing schema for those looking to drive performance by improving operations, but they help take practices developed for particular industries and functions and generalize them across all contexts. The guidance they provide is outstanding.”  
**— Dr. Joel Podolny, CEO, Honor Education, former VP & Dean of Apple University, former Dean of Yale School of Management**

LIBERATING OUR COLLECTIVE GREATNESS THROUGH  
SLOWIFICATION, SIMPLIFICATION, AND AMPLIFICATION

# Wiring the Winning Organization



GENE KIM and  
STEVEN J. SPEAR

*Foreword by* ADM John Richardson, US Navy (Retired)  
former Chief of Naval Operations

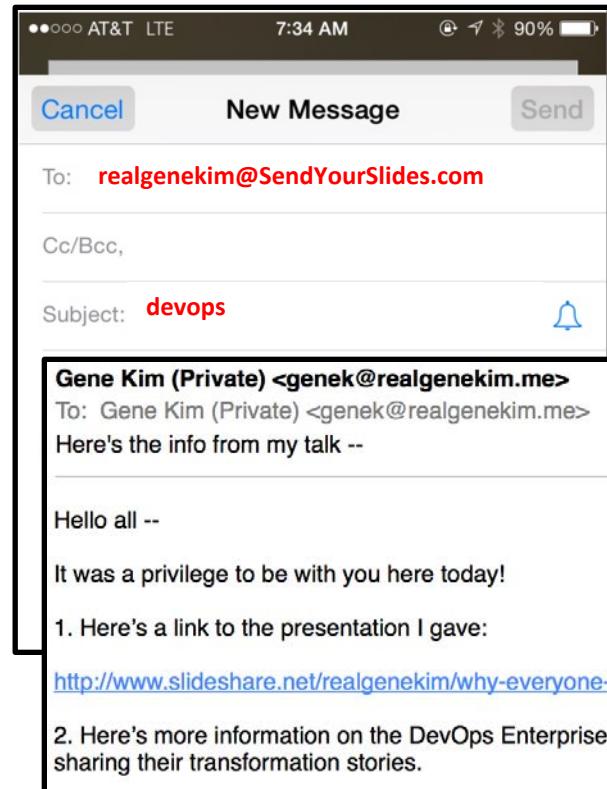
# Want More Learn More?

To receive this presentation and the following:

- PDF and audio excerpts from *The Unicorn Project*
- Eight excerpts from *Beyond The Phoenix Project* audio series w/John Willis
- The 140 page excerpt of *The DevOps Handbook*
- The 140 page excerpt of *The Phoenix Project*
- Videos and slides from DevOps Enterprise 2014-2019
- One hour excerpt of *The Phoenix Project* audiobook

Just pick up your phone, and send an email:

To: **realgenekim@SendYourSlides.com**  
Subject: **devops**



## **Exercise 2: Simplification**

**Incrementalize: separate the novel  
from the known.**

**Incrementalism on Large Scale:**  
Partitioning known from novel and adding  
novelty in small bits not large bites.

# From President Kennedy to Apollo 11



# Less giant leaps



MERCURY ATLAS D  
FEBRUARY 20, 1962



GEMINI-TITAN 5  
AUGUST 21, 1965

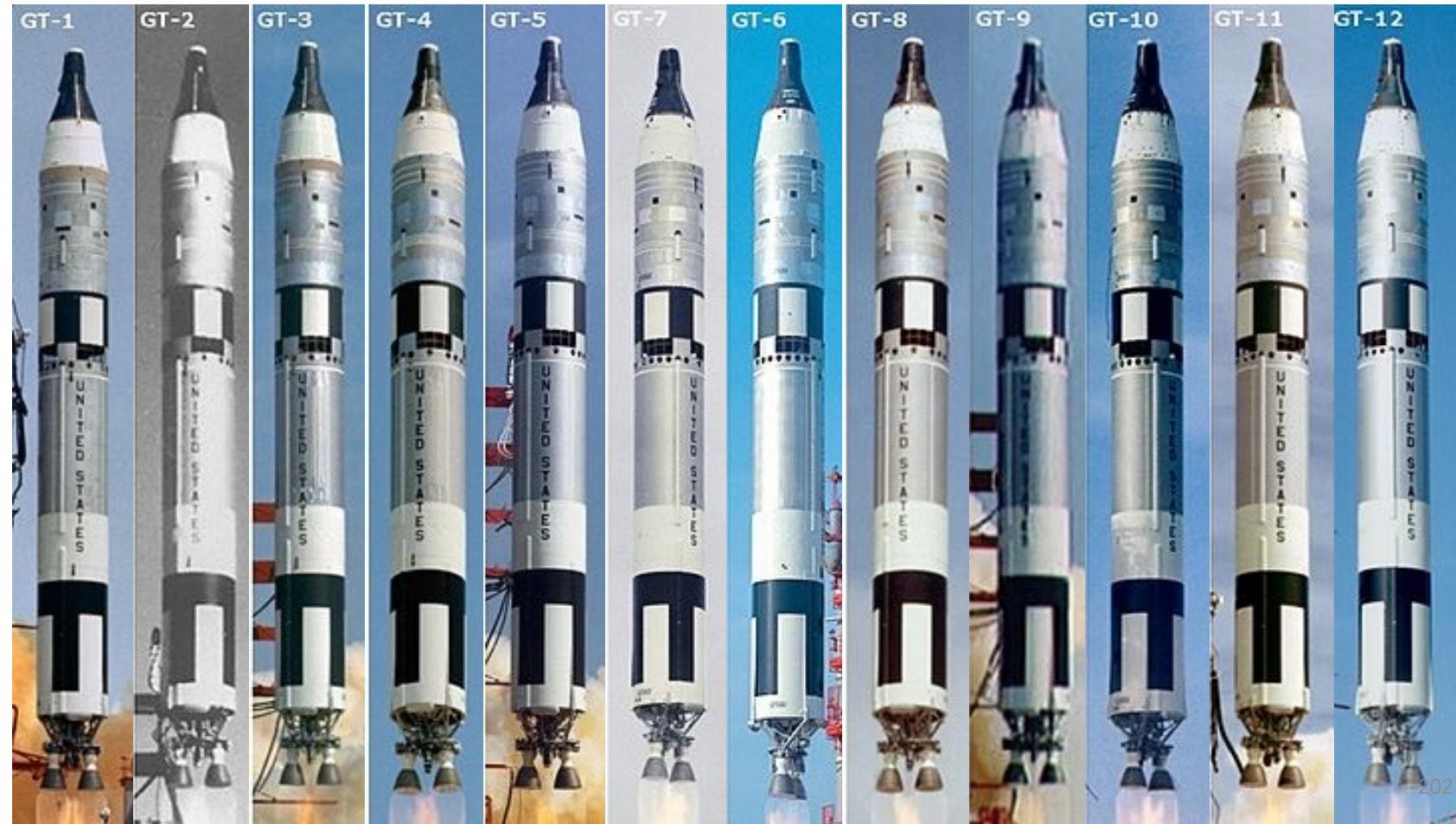


APOLLO 11  
JULY 16, 1969 200

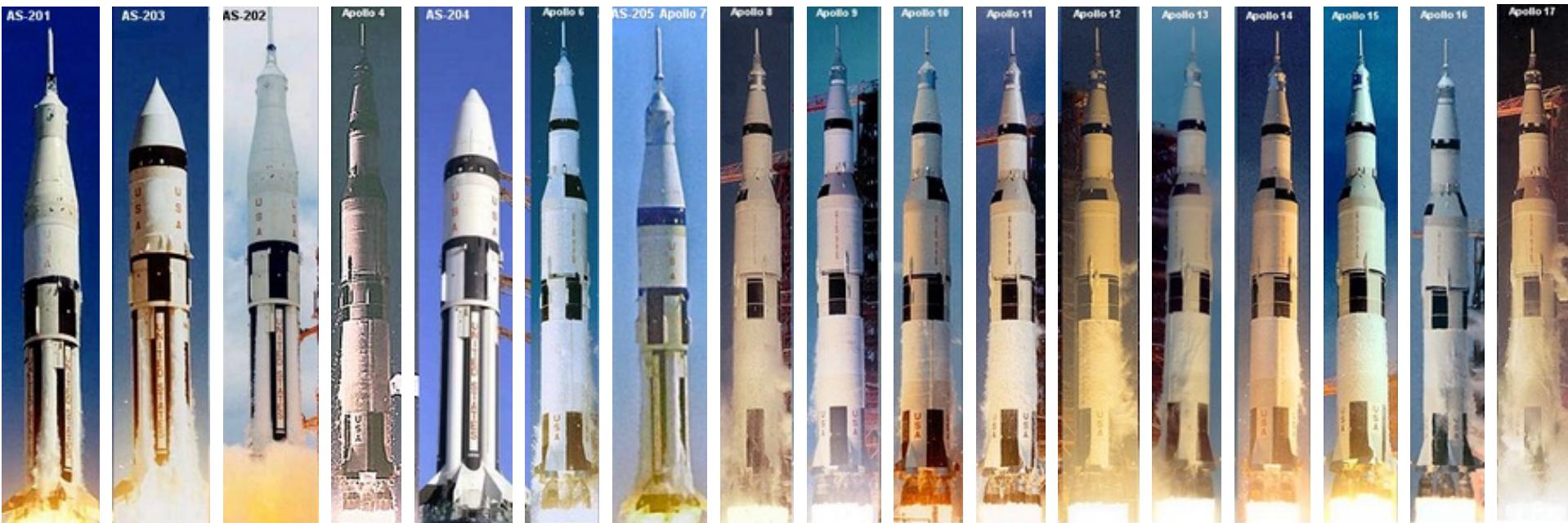
# Mercury: Small steps



# Gemini: More small steps



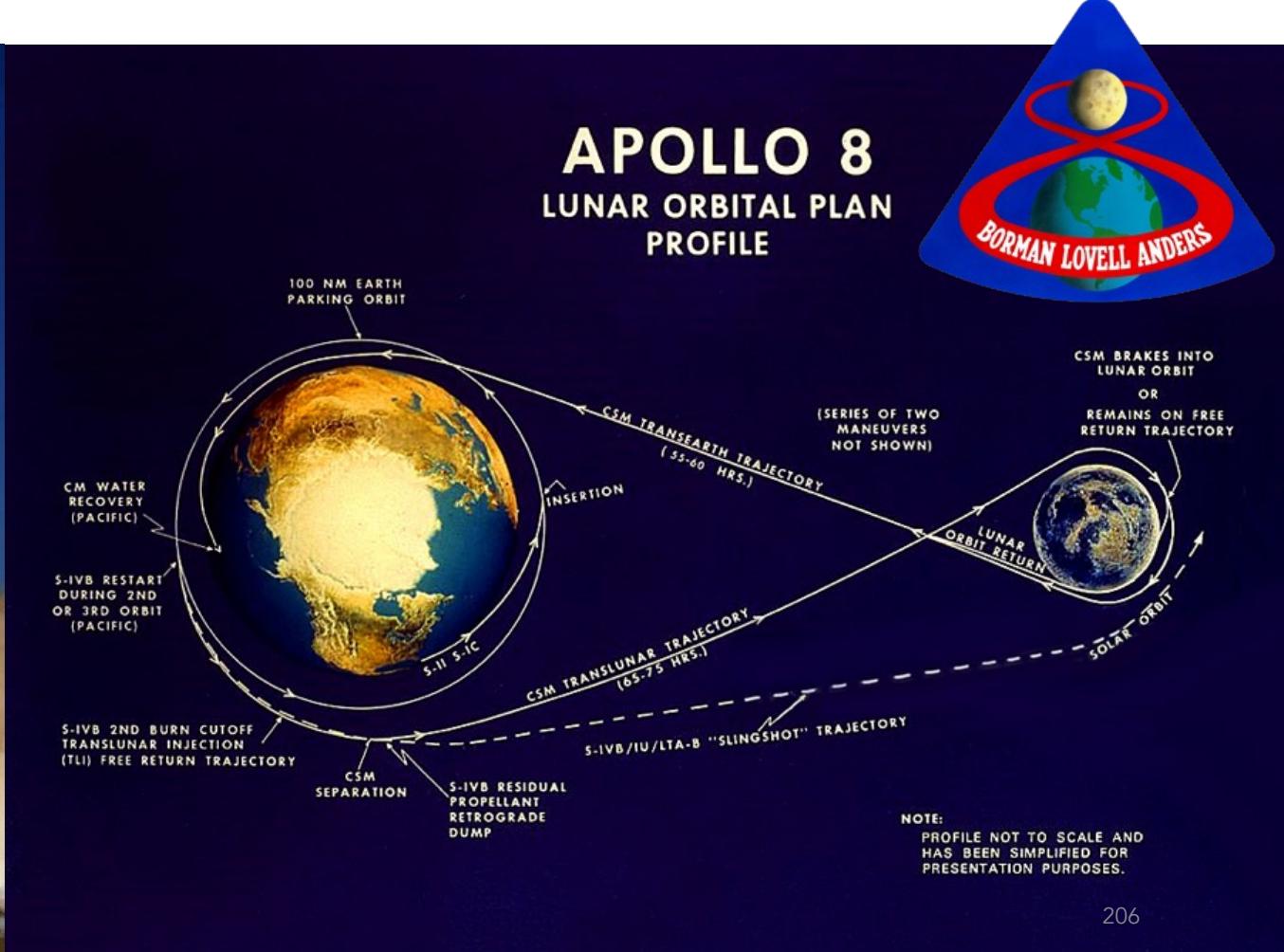
# Apollo: Even more small steps



# Apollo 7: Oct 11-22 1968—Orbital demo of Block II CSM



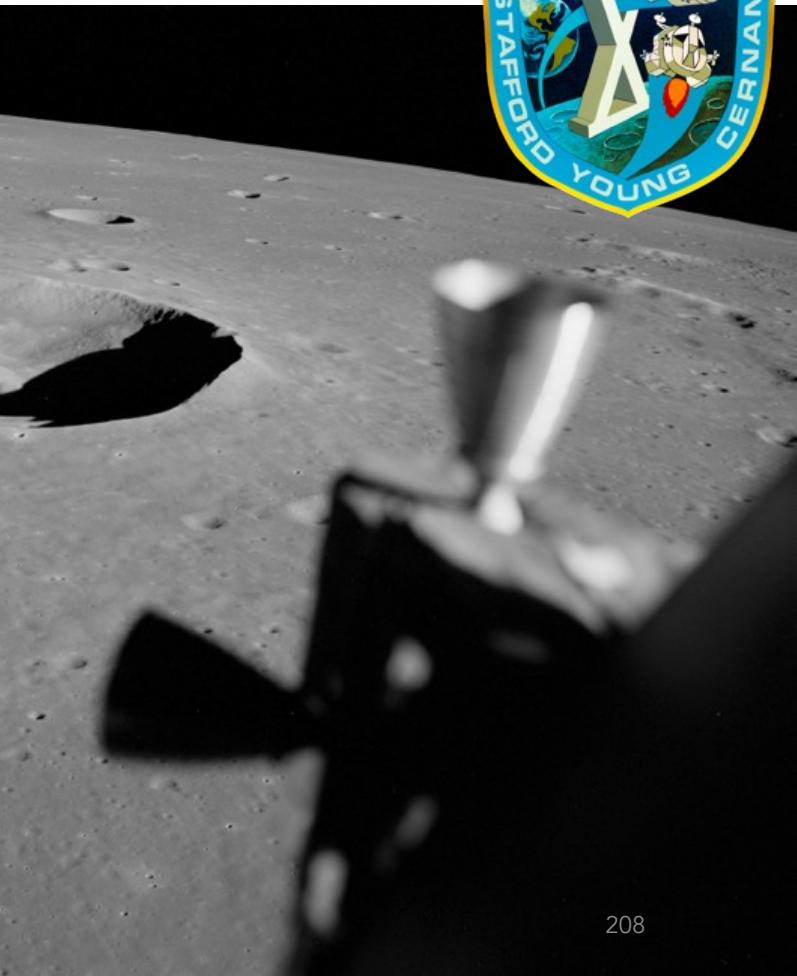
# APOLLO 8: Dec 21 1968—Lunar Orbit and Back



# Apollo 9: Test of all systems in Earth orbit



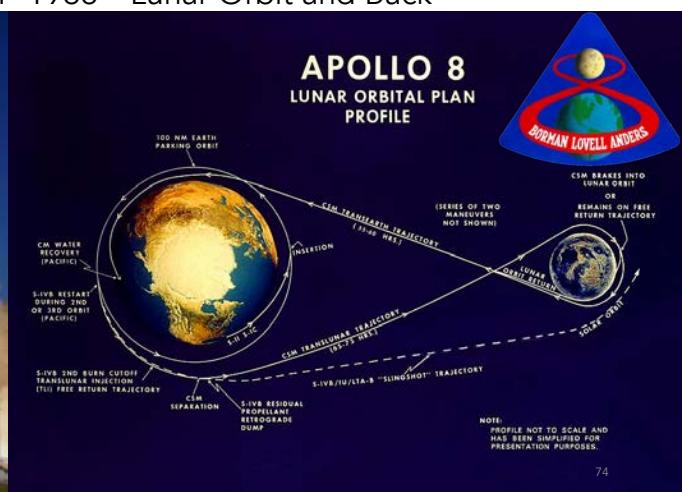
# Apollo 10: Within 47,000 feet of landing



APOLLO 7: Oct 11-22 1968—Orbital demo of Block II CSM



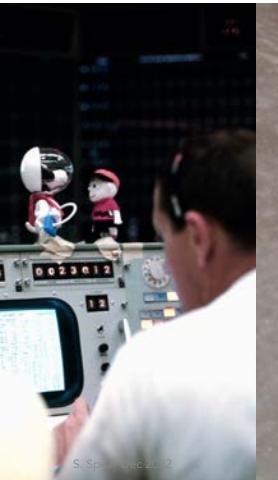
APOLLO 8: Dec 21 1968—Lunar Orbit and Back



APOLLO 9: Test of all systems in Earth orbit



APOLLO 10: Within 47,000 feet of landing



# APOOLLO 11: The Eagle has landed...small step



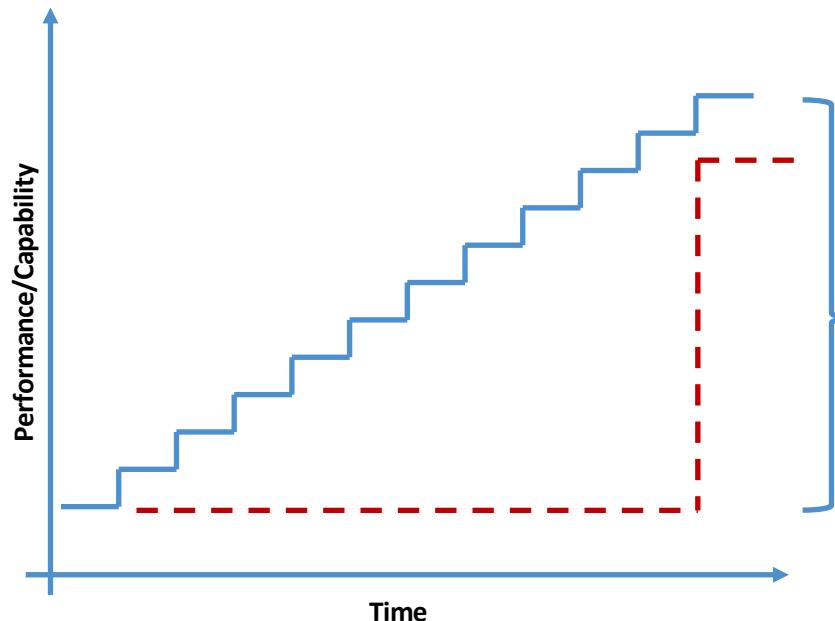
# Small step build of giant leap knowledge



# Small step build of giant leap knowledge

## The Giant Leap Problem

Requires many factors to be explored simultaneously + demands working far behind frontier of what is already known

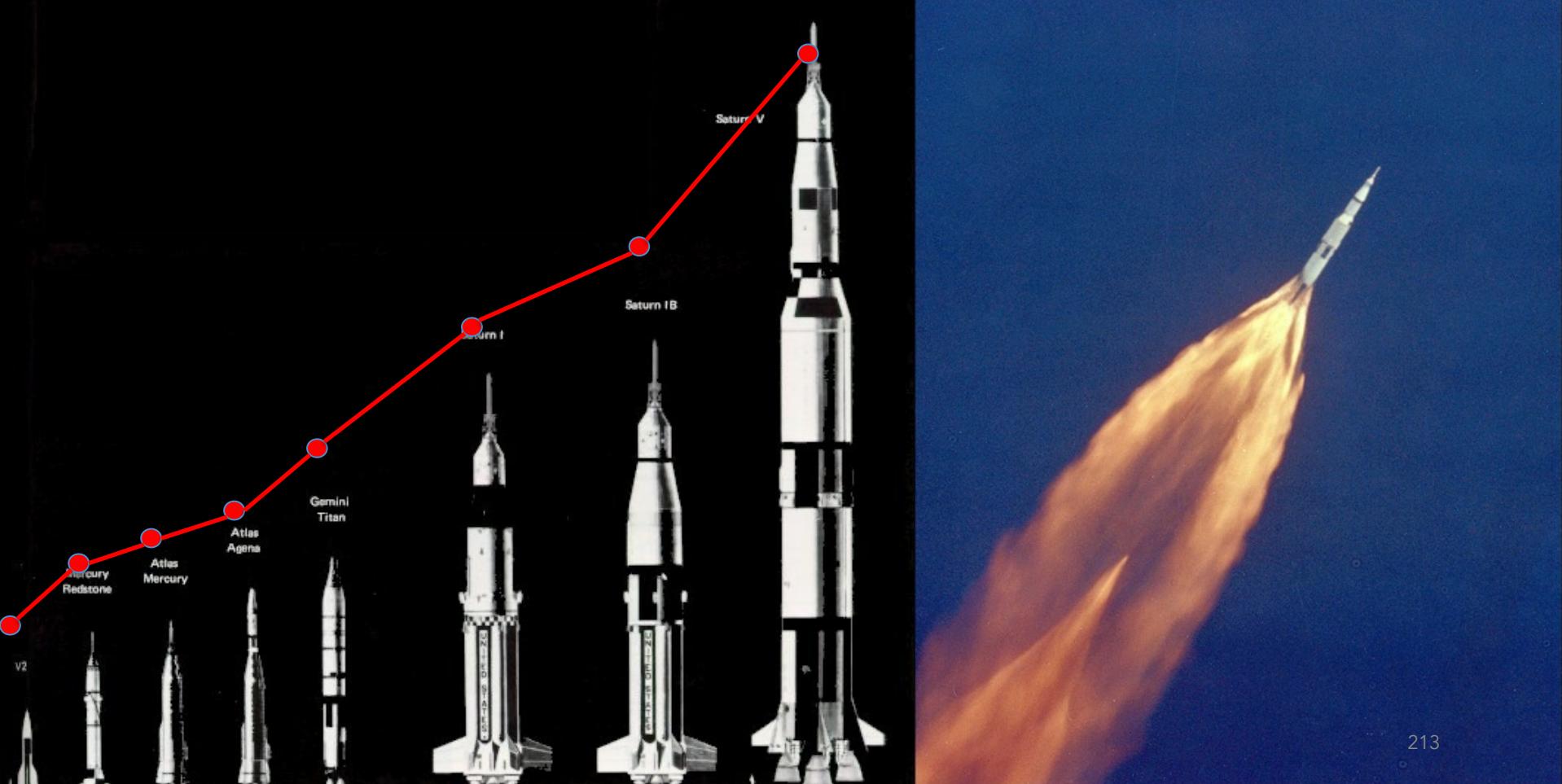


## The Many Small Steps Advantage



Each step less complex in terms of number of factors being developed/tested and more incremental in terms of building on what's already known.

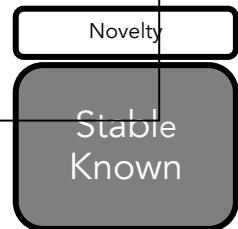
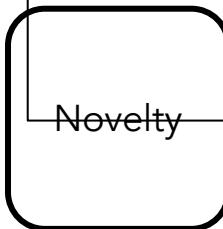
# Small steps build giant leaps of knowledge



# Exercise: Incrementalization

## KNOWNS

- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_

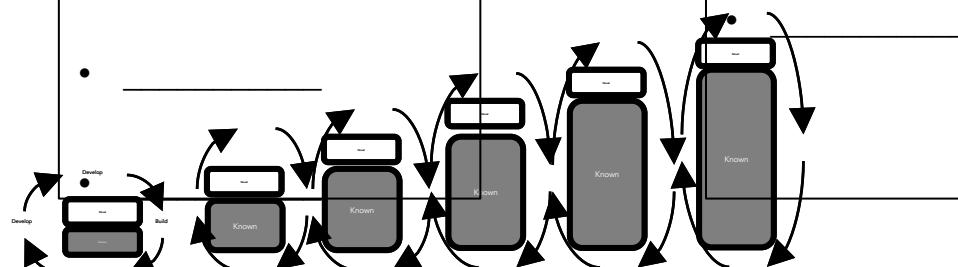


## UNKNOWNs

- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_

## Increments

- \_\_\_\_\_
- \_\_\_\_\_

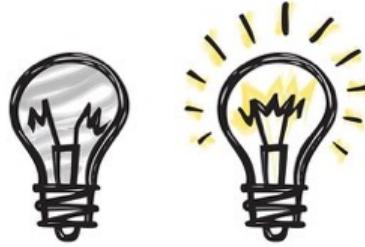


## Key insights for reports out to colleagues

- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_

LIBERATING OUR COLLECTIVE GREATNESS THROUGH  
SLOWIFICATION, SIMPLIFICATION, AND AMPLIFICATION

# Wiring the Winning Organization



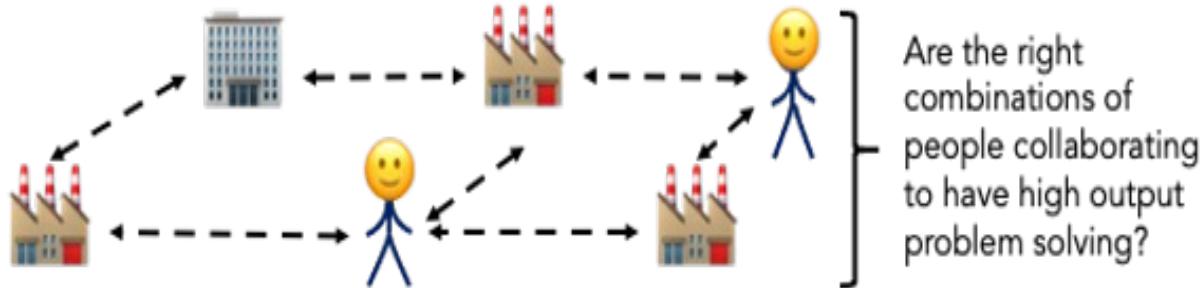
GENE KIM and  
STEVEN J. SPEAR

*Foreword by* ADM John Richardson, US Navy (Retired)  
former Chief of Naval Operations

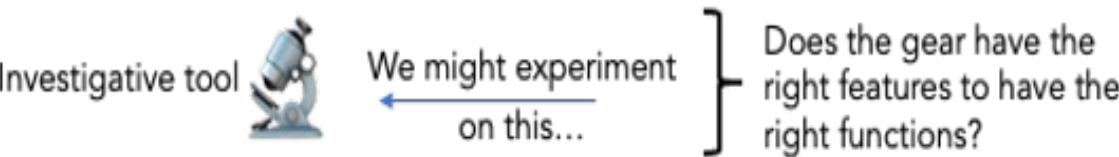
# HARNESSING OUR INGENIETY

## Layers of creativity

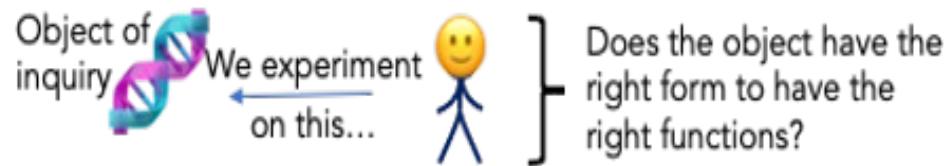
Layer 3: Enterprise processes for integrating the pieces into a harmonized whole.



Layer 2: Equipment and instrumentation through which creativity is expressed.



Layer 1: "The object" being created, formed, or otherwise changed.

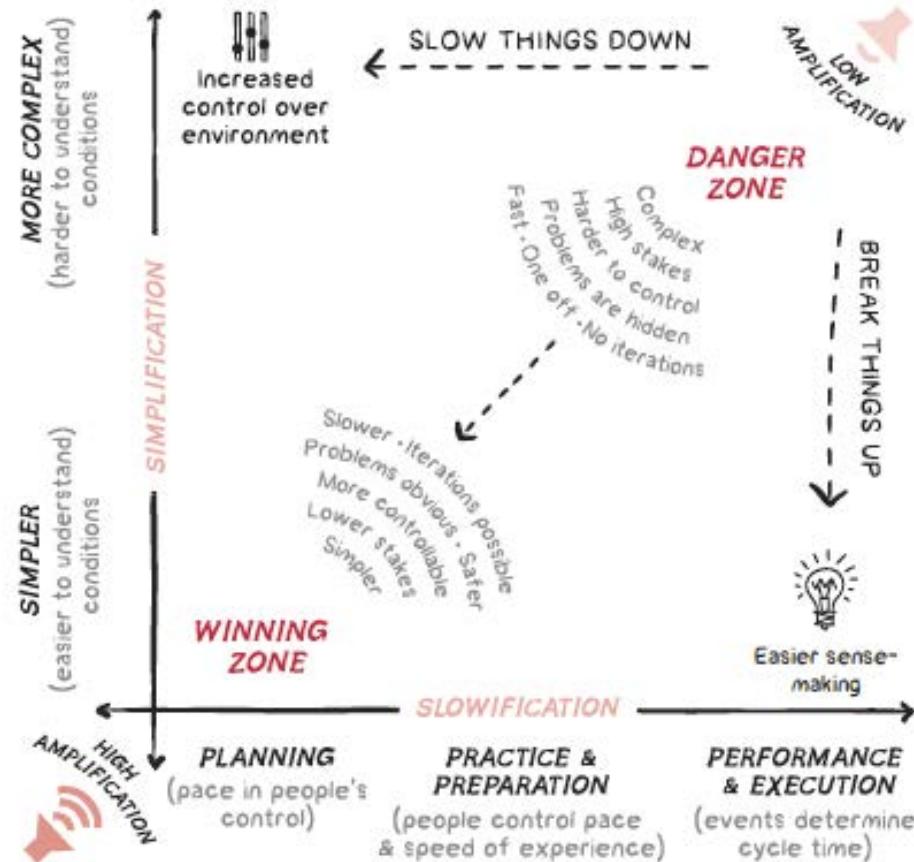


# Answer: It's all about the brains

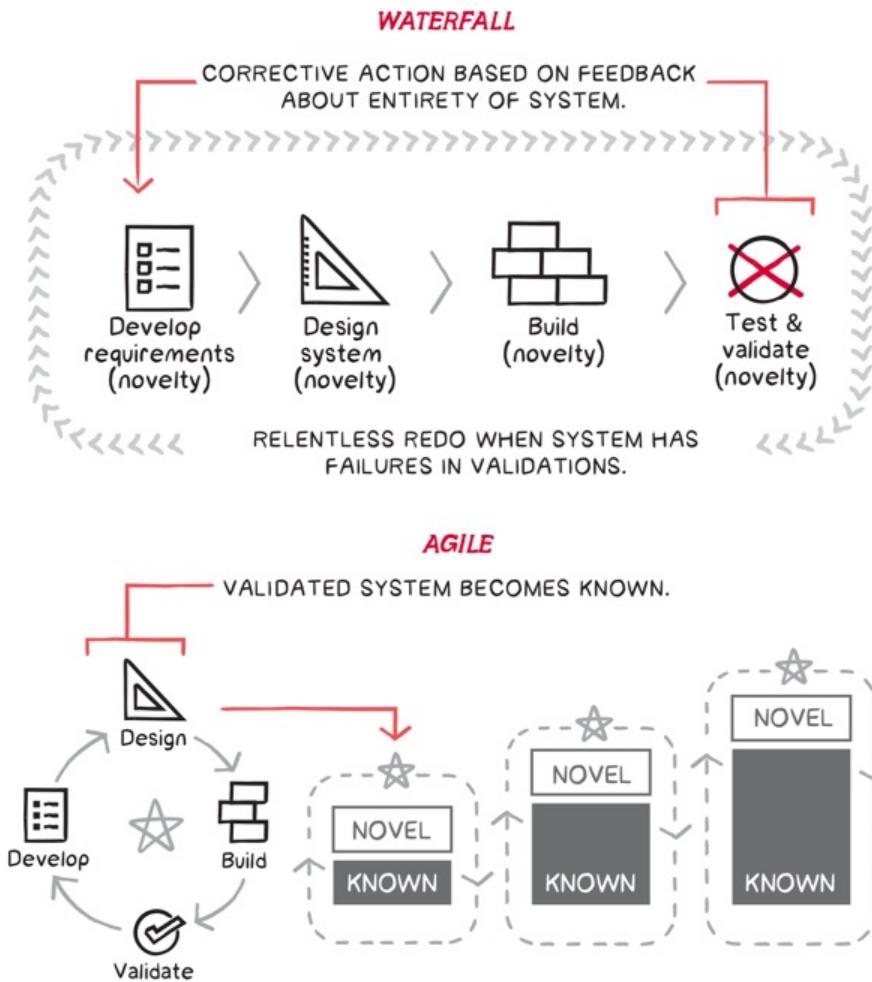
All else equal—commonality in resources and opportunities, then discommonality of outcomes must be due to discommonality in management systems. In particular, how easy they make for people to solve hard problems quicker and easier.



# "Danger zone" vs. "Winning Zone"



**FIGURE 7.3** Contrasting Waterfall Approaches with Incremental (Agile) Ones



# **Independence of Action:**

**To what degree can your teams work independently of each other?**

**(To get something important done,  
how many people do you need to talk to,  
and how high do you need to go?)**

# **Independence of Action:**

**To what degree can your teams work independently of each other?**

**(To get something important done,  
how many people do you need to talk to,  
and how high do you need to go?)**

# 2003: Space Shuttle Disaster

- Two days after the launch of Space Shuttle Columbia, controllers observe that a 21" piece of foam struck the bottom of shuttle
- After considerable study, NASA leadership concludes there is “no risk to the shuttle or crew” — foam strikes had happened before, so it must be okay (the deviances were normalized)
- Upon re-entry, Columbia breaks up , killing all seven astronauts



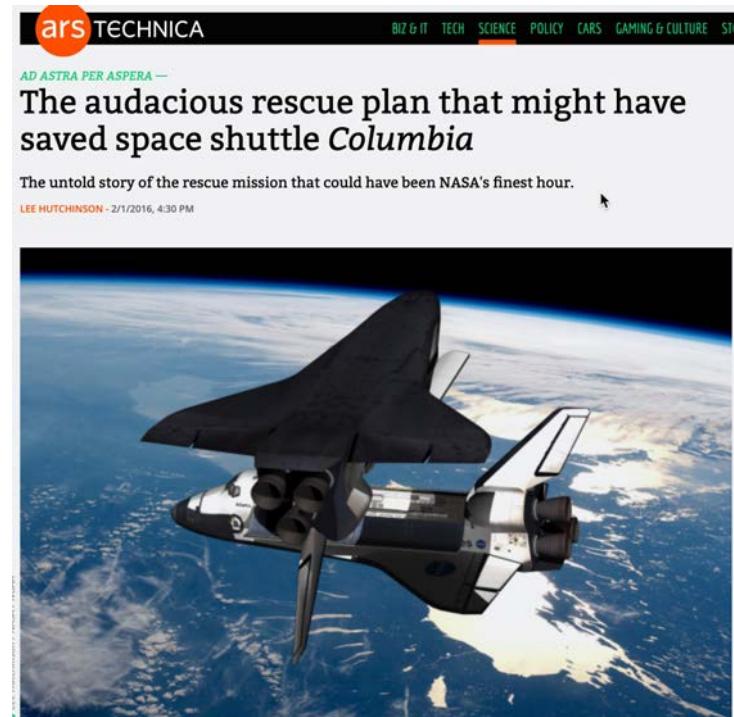
# 2002: Analysis

- NASA creates the Columbia Accident Investigation Board to study what happened and make recommendations
- One of their questions: could the astronauts have been rescued, once determination that safe re-entry was impossible?
- Appendix D.13 describes the rescue plan, that uses the Atlantis to bring all the astronauts home
  - “It took 18 months of planning to develop the procedures, modify the tools, test and simulate the GN&C, EVA, and robotics choreography, and prepare all the paperwork to satisfy everyone that it was a safe plan for both orbiters and the crew.

“The only hope that this plan would have ever had would have been if the plan had already been in place prior to Columbia’s launch, as there is no way on this Earth that NASA would have approved a flight with untested procedures that could destroy both orbiters.”

# 2002: Analysis

- Writer Lee Hutchinson in Ars Technica: "...it could have been NASA's finest hour"
- "Columbia's loss ended not only lives but also careers at all levels of NASA."
- Operating tempo prevented slowification



# Exercise

- We've just learned about amazing examples of how we may start with a plan, and how we can respond when things don't go according to plan
- In your groups, think of a time when you have a plan (e.g., an iteration plan) and you...
  - ...start to realize that the plan isn't what you should actually be doing. What would ideally happen to break free of the plan, to work on something of a genuinely higher priority?
  - ...And conversely, what should happen to protect the plan, against things that aren't genuinely a higher priority?

# 2020 USAF Spark Tank Competition



# Gene Summary

- HCD is a couch moving activity: devs/engineers must be close to the customer — they must pickup both sides of the couch if needed, because the customer is often unable to articulate the problem
  - joint cognition and problem solving
  - don't know what's possible
  - often think it's their fault
  - why it's important for devs to shadow their customers
- story about my first UX experience in 2004

# Exercise

- You've just learned about great examples of human-centered design
- In your groups, think about a time when a problem could have been lessened if there were more direct contact with the customer (e.g., the operator, the downstream consumer, the next step in the process)
- Assign one person to be ready to report out to the rest of the group on what they heard — not on their problems, but to summarize the learnings

# 1969: Apollo 11 Lunar Landing

- One hour from fulfilling President Kennedy's 1961 challenge of "achieving the goal, before this decade is out, of landing a man on the moon and returning him safely to the earth."
- Several things go wrong as Neil Armstrong and Buzz Aldrin detach the Lunar Module for the sixty mile descent
  - They detached 2-3 seconds too early, so they're traveling too fast
  - 6,000 feet above lunar surface, lunar guidance computer generates repeated "1201 program alarms" and would restart
  - One minute from landing, Armstrong spots boulders strewn across
- Each of these could have resulted in aborting landing or disastrous failure

# 1969: Massive Investment In Simulators, Rehearsals

- Planning and practice wasn't just for the astronauts.
- It was also for Mission Control, who were the ground controllers, the nexus of the tens of thousands of people whose work enabled the mission.
- Kranz described how the simulation team was constantly “studying the controllers, crews, and mission strategy, looking for the holes and developing new training runs to exploit the perceived holes.”



Lunar landing simulator



Lunar landing training vehicle



Lunar module

# 1969: Massive Investment In Simulators, Rehearsals

- Astronauts rehearsed countless lunar landing scenarios, including detaching too early/late, and having to find alternate landing sites



# 1969: Simulation Team Stress Tested Team

- Weeks before the scheduled launch of Apollo 11, the simulation team kept proving that astronauts and controllers still had not figured out key elements of the mission.
- Kranz described one training session during this time, which resulted in a catastrophic failure (piloted by Apollo 12 astronauts!):
  - “...[in the simulation,] the crew was splattered across the Sea of Tranquillity [which was the intended landing zone]. This was our first crash, the result of a few seconds” delay in our communication and decision process.”
- In the very last scheduled simulation, the Simulation Supervisor “threw the 1201 code at them” — resulting in an incorrectly aborted landing

# 1969: The Result

- Neil Armstrong lands on the moon, and utters the famous words, “The Eagle has landed.”
- NASA prepared the Apollo astronauts for success by Slowifying

# 1968: US Navy Aviation

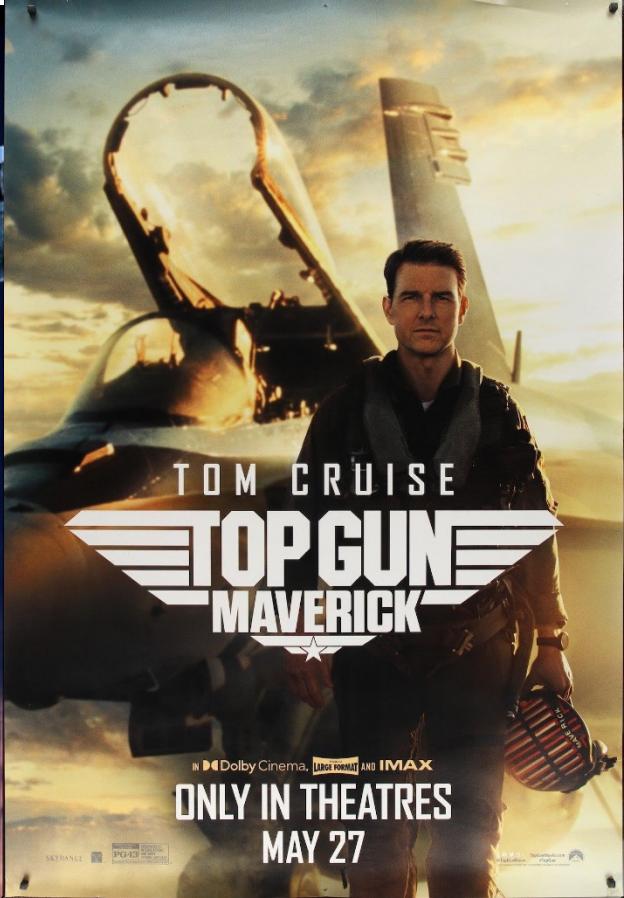
- U.S. pilots were suffering terribly during aerial combat in Vietnam, with win-to-loss ratios of only 2:1
- In a single seventeen-day period, eleven American aviators were killed
- Captain Frank Ault, a WW II veteran from surface and aviation community, was tasked to figure out why and what to do
- The U.S. Navy was depending too much on “on the job” learning, leaving them unprepared for the realities of actual combat

# 1968: Ault Report Findings

- One of the findings: recommended creating the Navy Fighters Weapons School
  - They created a four-week program, part classroom, but also had advanced in-flight training with simulated combat with instructors using planes and tactics similar to those used by the adversary
- Pilots that trained in TOPGUN returned to their squadrons, and integrating their learning into their training programs
- The win-loss ratio improved to 13:1
- The U.S. Navy helped the pilots succeed by slowifying: moving needed learning and training into Planning and Preparation

# 1969: “TOPGUN” Naval Air Station Miramar

- 1986:  
Top Gun
- 2022:  
Top Gun:  
Maverick



# **Incrementalism and Experimentalism: The art of being aggressively unambitious**

**Steve Spear DBA MS MS**  
Principal, HVE LLC  
MIT, Senior Lecturer  
IHI, Senior Fellow  
Author, *The High Velocity Edge*

The award  
winning  
masterpiece  
by Picasso  
when he was a  
tween.



First attempt at basic  
composition



Add dramatic detail

Nurse or nun?



Who is she  
holding?

Gesture of care  
or compassion

Increase the narrative tension



A doctor  
checking a pulse

We're now viewing the patient face on  
(not in profile) raising our emotional  
investment in the painted story

More obviously  
a nurse hold a  
child.

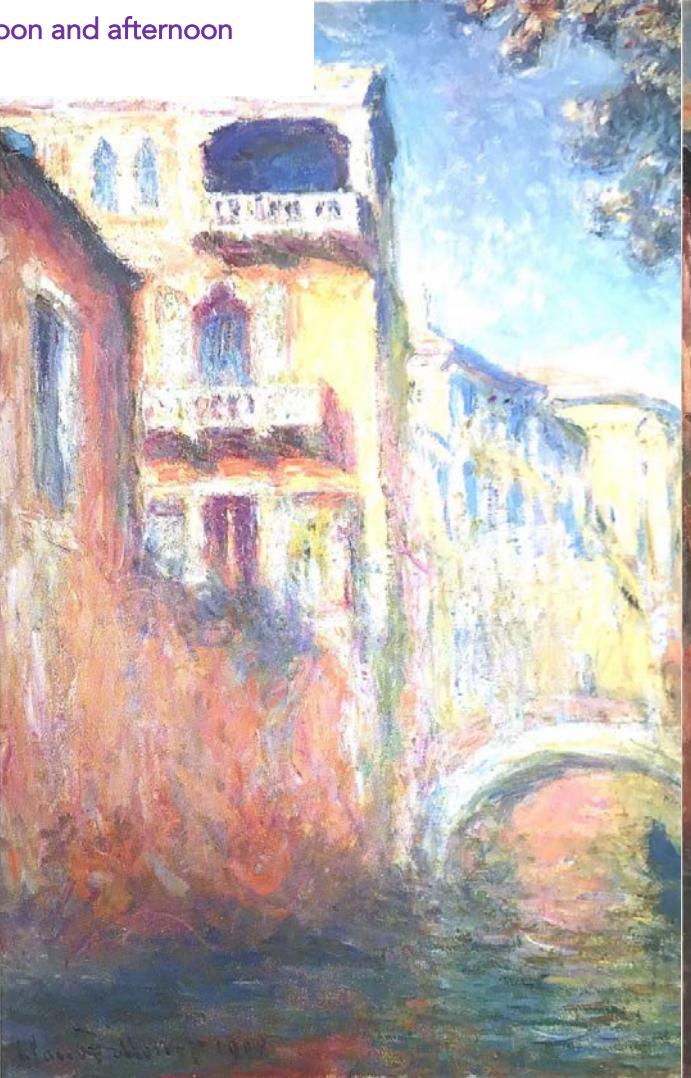
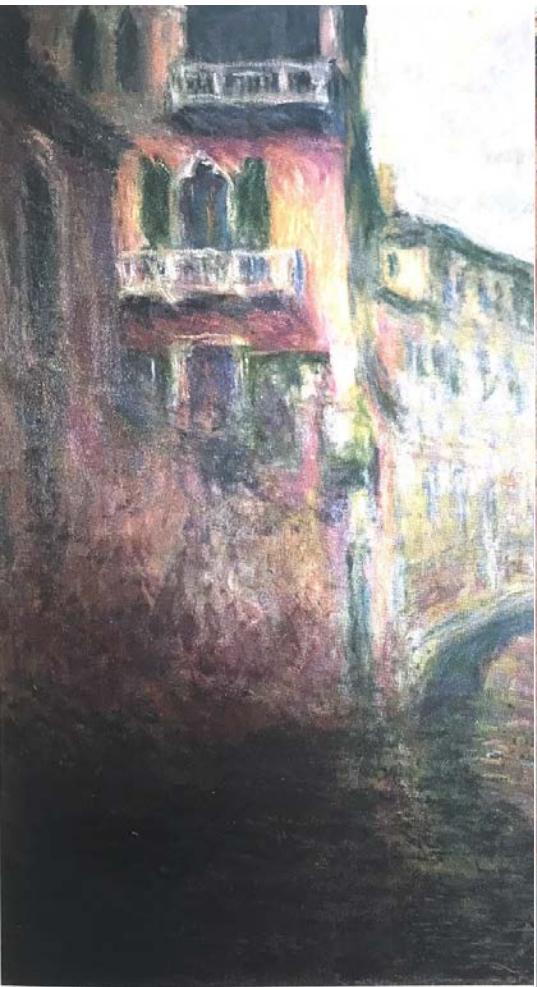
@RealGeneKim

Tie the pieces together and adding finishing touches.

Emphasize the cabinets left and right to force our perspective onto the patient and her suffering.



Monet in the Mediterranean: morning noon and afternoon  
easels



# **Transition to Gene**

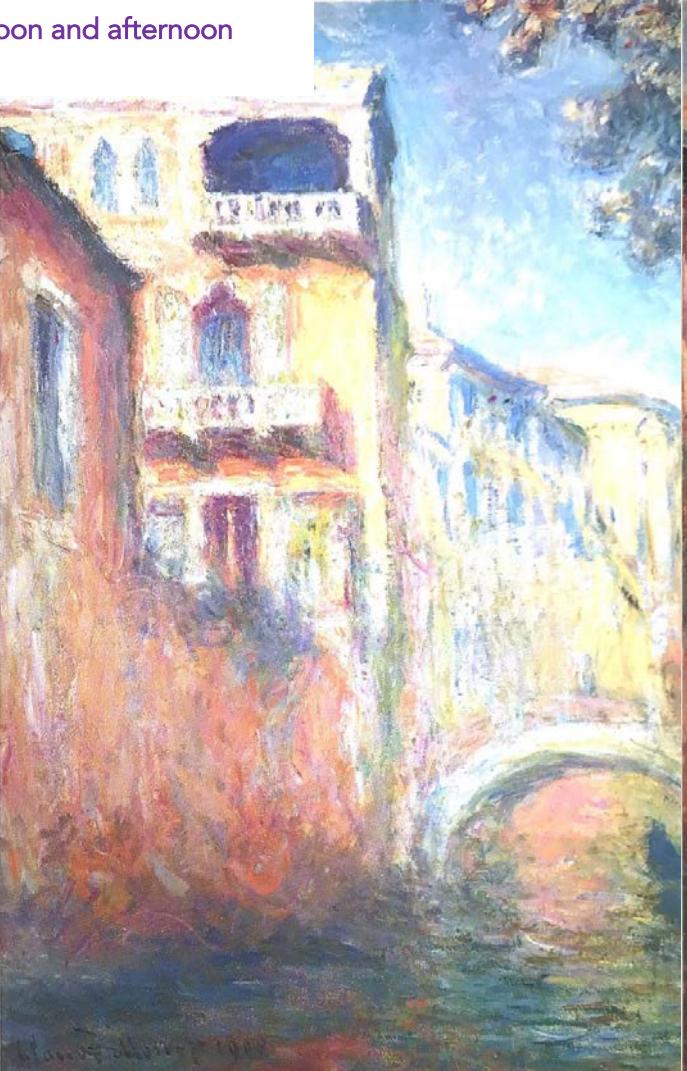
# 2007: Apple iPhone

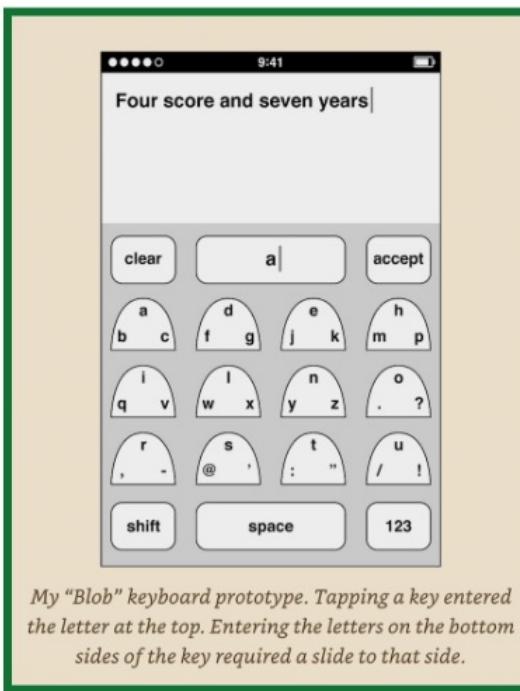
- June 29, 2007, Steve Jobs, co-founder and CEO of Apple, announced to 3,000 people at Moscone Center the iPhone
- When released six months later, it sold \$118MM in its first quarter of sales
- The darkest moment in its 3 year development?  
In 2005, Scott Forstall, SVP of Software, “failed to thumb-type anything intelligible” on the prototype

# 2007: Apple iPhone

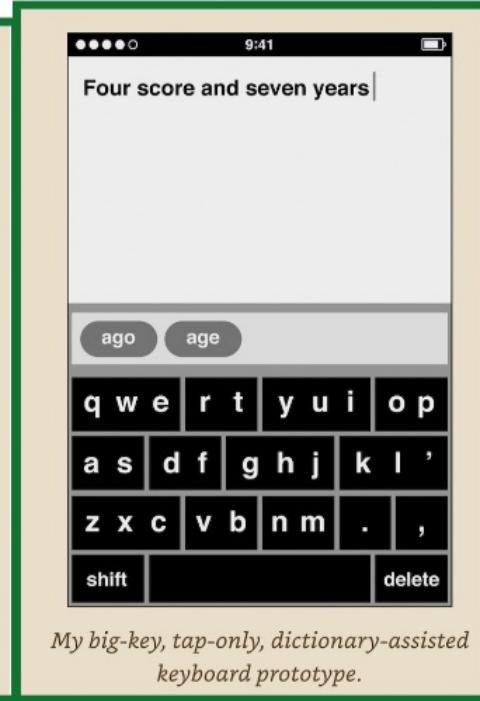
- From 2005-2007: number of software engineers working on iPhone: 15 people (a two-pizza team)
- September 2005: All engineers drop all work to generate ideas for keyboard

Monet in the Mediterranean: morning noon and afternoon  
easels

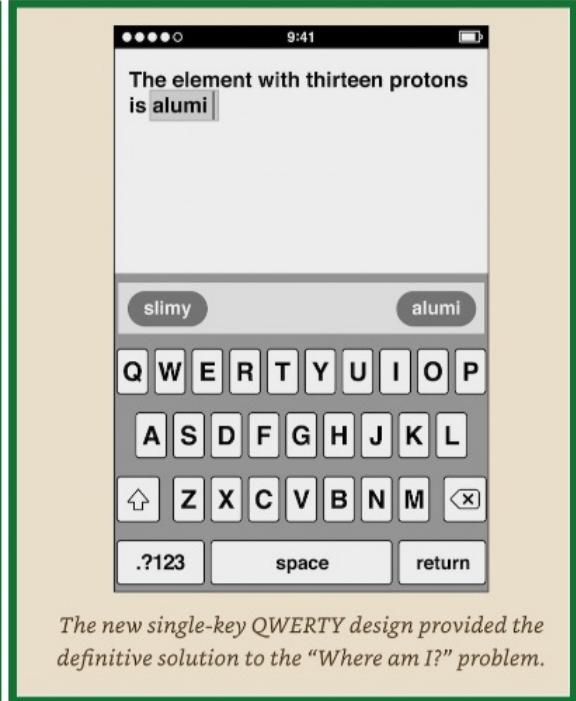




My “Blob” keyboard prototype. Tapping a key entered the letter at the top. Entering the letters on the bottom sides of the key required a slide to that side.



My big-key, tap-only, dictionary-assisted keyboard prototype.



The new single-key QWERTY design provided the definitive solution to the “Where am I?” problem.

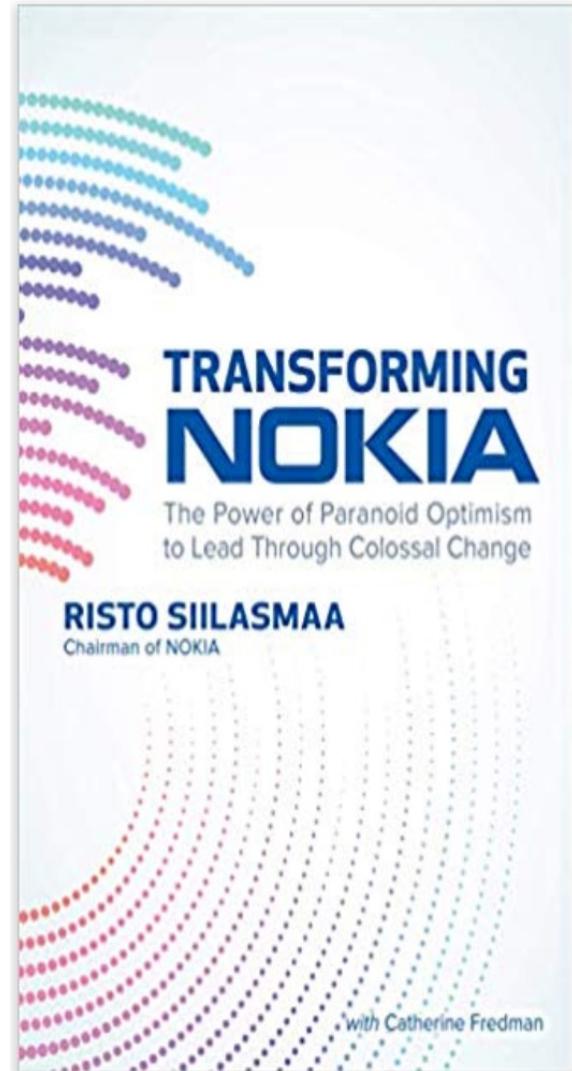
# 2007: Apple iPhone

- June 2007: after announcement, the engineers learn what the name of their product is
- Software development team grows to one hundred, to finish implementing all the features that were promised
  - Mail, Calendar, Safari, YouTube, Maps,
  - Prior to this, it was all done by 15 people

# Risto Siilasmaa, NOKIA

“In 2010, Risto Siilasmaa was a board director at Nokia. When he learned that generating a Symbian build took *a whole forty-eight hours*, he said that it felt like someone hit him in the head with a sledgehammer,” Erik says. “He knew that if it took two days for anyone to determine whether a change worked or would have to be redone, there was a fundamental and fatal flaw in their architecture that doomed their near-term profitability and long-term viability. They could have had twenty times more developers, and it wouldn’t have made them go any faster.

Erik pauses. “It’s incredible. Sensei Siilasmaa knew that all the hopes and promises made by the engineering organization was a mirage. Even though there were numerous internal efforts to migrate off of Symbian, it was always shot down by the top executives until it was too late.



# **Slowification : Incrementalization + Linearization**

# Siemens Healthineers

- Thomas Jachmann  
Head of Software,  
Computed  
Tomography, Siemens  
Healthineers
- €14.5B annual revenue
- Installed base: 600K  
machines
- 54K employees

The environment  
Cyber-physical system of systems: Computed Tomography scanners

## Challenges

- Growing complexity (both product and the organization)
- Transforming from a hardware driven to a Software driven company causing functionality realized in Software to grow rapidly
- Systems become increasingly interconnected, expanding the complexity beyond the boundaries of systems
- Growing compatibility concerns between parts of large, complex systems because various parts move at different “pace of change”, especially combining Hardware and Software
- Fast pace of change of modern software technologies requires clear separation of concerns
- Accelerating innovation cycles



Thomas Jachmann | Computed Tomography  
© Siemens Healthineers, 2021

DEVOPS  
ENTERPRISE  
SUMMIT

an IT REVOLUTION EVENT

GET  
TOGETHER  
GO  
FASTER

Siemens Healthineers (numbers prior to the Varian merger)

**Market leader** in majority of businesses

**€14.5bn** Revenue<sup>(1)</sup>

**>70** countries with direct presence

**~54,000** highly skilled employees

**~18,500+** IPRs, thereof 11,500 granted patents

**~600,000** installed base

**~240,000** patient touch points every hour

Access to care for **>1.4bn people** in developing countries

>70% of critical clinical decisions are influenced by the type of technology we provide\*

>90% of global top 100 providers partner with us

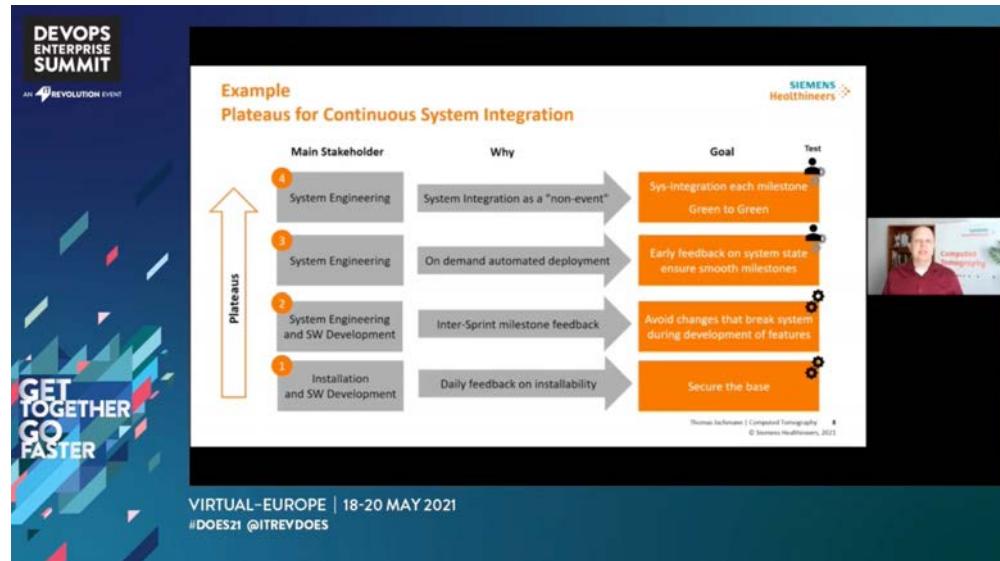
Thomas Jachmann | Computed Tomography  
© Siemens Healthineers, 2021

VIRTUAL-EUROPE | 18-20 MAY 2021  
#DOES21 @ITREYDOES

01:28  
00:48

# Siemens Healthineers

- Extremely regulated environment
  - THERAC-25 software defect accidents irradiated six patients in the 1980s
- Goal was to make software installation a “non-event” for entire fleet of CT scanners, and constellation of control systems and PCs that support them

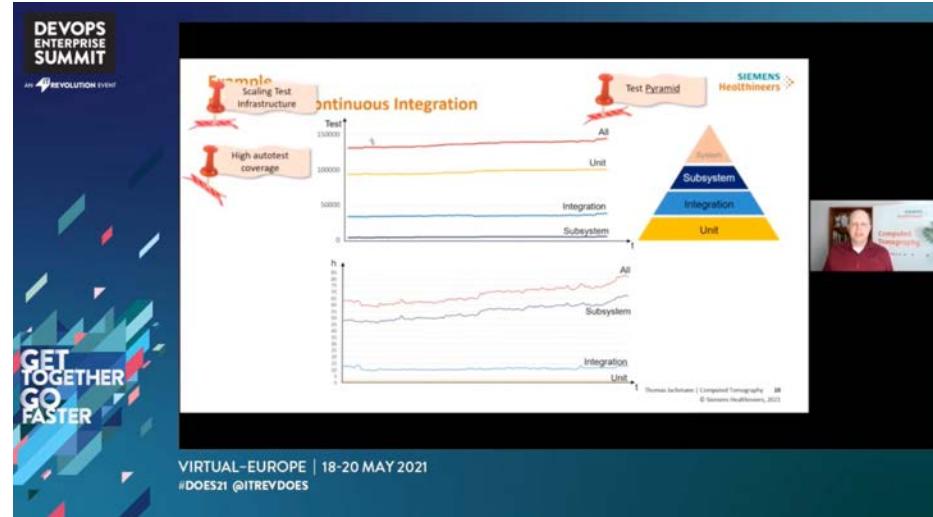


# Their Approach

- There is a regulatory wall between pushing out software updates — that remains unchanged
- Their goal was to create very fast and frequent feedback loops in every prior step: system integration test, developer unit tests
- Make Verification and Validation tests small, reducing testing of things that did not change
- Major bottleneck was software integration testing

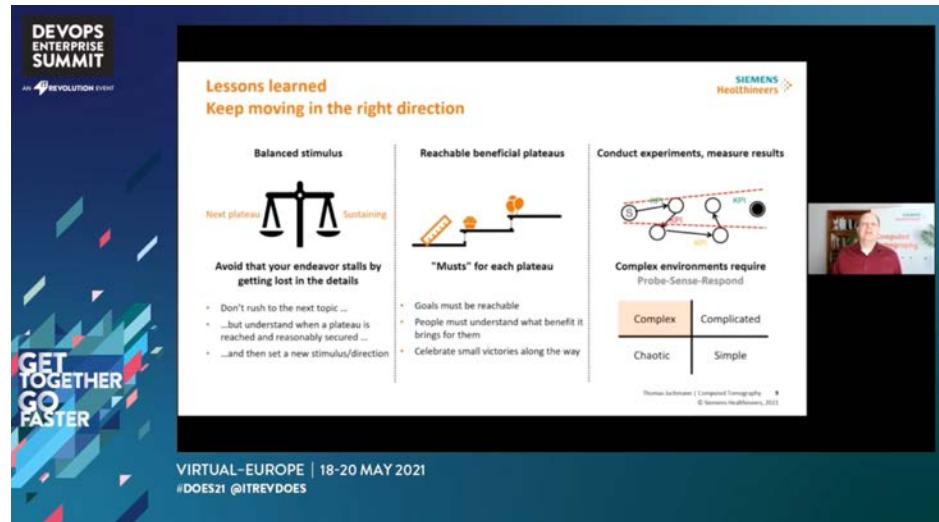
# Their Approach

- Work closely with all stakeholders, especially Systems Engineering and System Integration Test — all moving the same couch, as opposed through working with documents
- Jointly designed and incrementally executed their processes, with a North Star goal of making systems testing a “non-event”
  - 6 month systems integration testing cycle -> daily cycle
  - 150K automated tests that run daily
  - At one time, required 80+ hours of test time



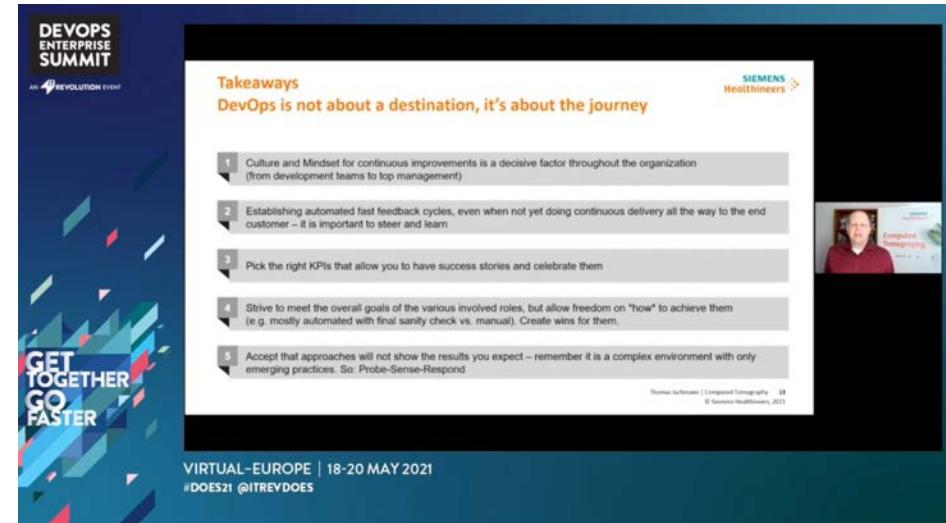
# Their Advice

- Leaders must be supportive of experimentation
- Approach was highly iterative — a complex domain that required probe-sense-respond
- They even their called incremental milestones “plateaus”



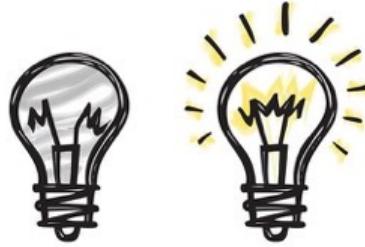
# Their Advice

- Required leaders to have the right “culture and mindset” — from team leaders to top leaders
- Required lots of freedom to explore solutions and approaches



LIBERATING OUR COLLECTIVE GREATNESS THROUGH  
SLOWIFICATION, SIMPLIFICATION, AND AMPLIFICATION

# Wiring the Winning Organization



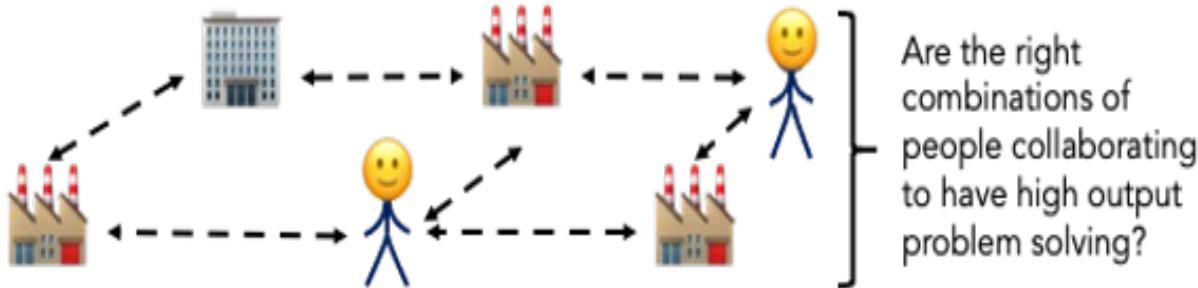
GENE KIM and  
STEVEN J. SPEAR

*Foreword by* ADM John Richardson, US Navy (Retired)  
former Chief of Naval Operations

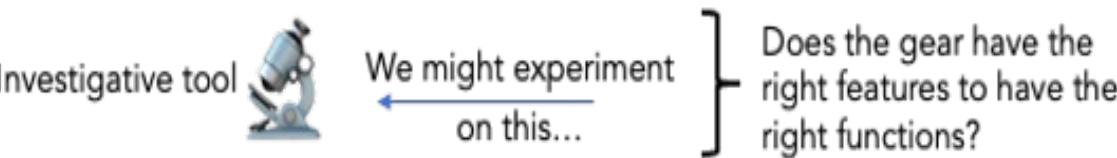
# HARNESSING OUR INGENIETY

## Layers of creativity

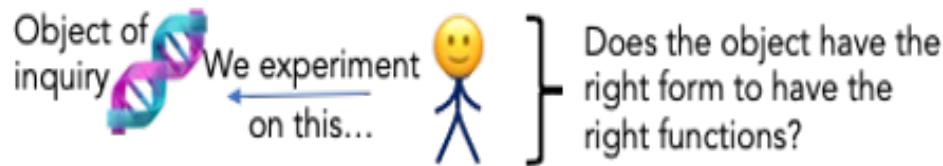
Layer 3: Enterprise processes for integrating the pieces into a harmonized whole.



Layer 2: Equipment and instrumentation through which creativity is expressed.



Layer 1: "The object" being created, formed, or otherwise changed.

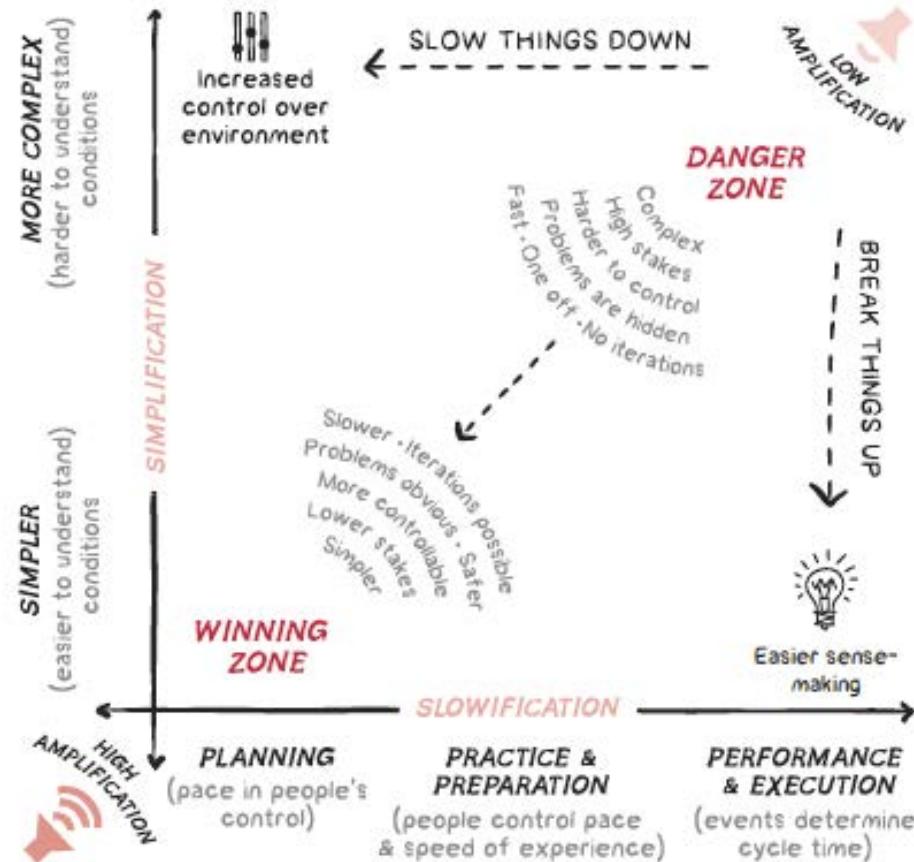


# Answer: It's all about the brains

All else equal—commonality in resources and opportunities, then discommonality of outcomes must be due to discommonality in management systems. In particular, how easy they make for people to solve hard problems quicker and easier.



# "Danger zone" vs. "Winning Zone"



**Linearization to Link the Functional Parts  
into a High Functioning Whole**

# Simplification Technique: Linearization

Making collaboration across boundaries quicker, easier, and more productive

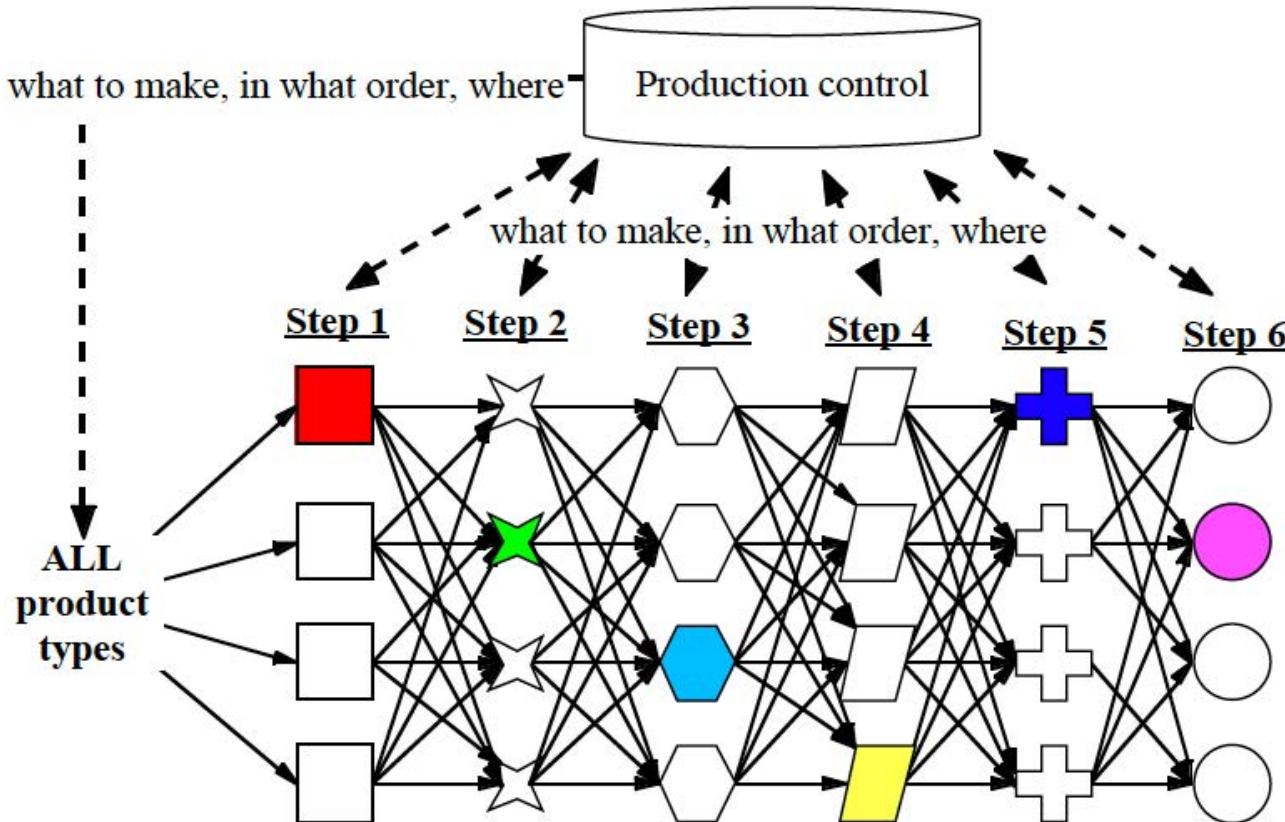
# Comparing among fabs...

|                          | Plant 1 | Plant 2 | Plant 3 |
|--------------------------|---------|---------|---------|
| Product families         | 50      | 3       | 4       |
| Tool utilization         | 61%     | 45%     | 24%     |
| Non-product wafer starts | 4%      | 25%     | 33%     |
| Die yield                | 85%     | 90%     | 77%     |
| Cycle time (days)        | 60      | 45      | 94      |

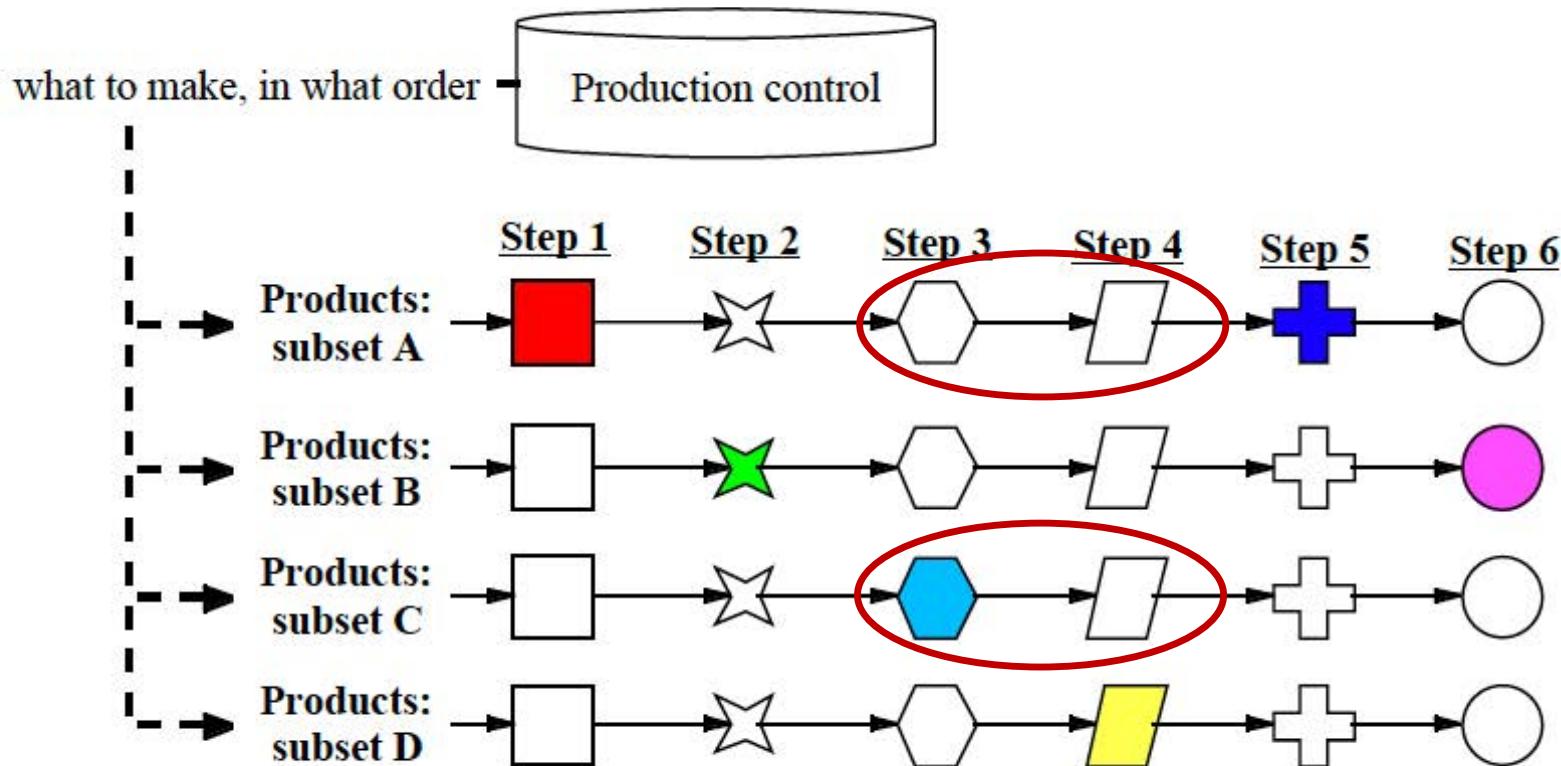
# Intel Mass ( Fab17) Results after 18 months. (1/2007- 6/2008)

- cycle time reduction: 66 % ( 60 days ->24 days)
- non productive time 75% eliminated
- Inventory reduction: similar
- Quality + Yield losses: 30% reduction
- Output (productivity): 20% increase
- Unit costs: 50% reduction
- Capacity: 20% gain + gain from reduced losses
- Factory life: extended from 2009 to 2014/15

# Pooled resources vs dedicated flow

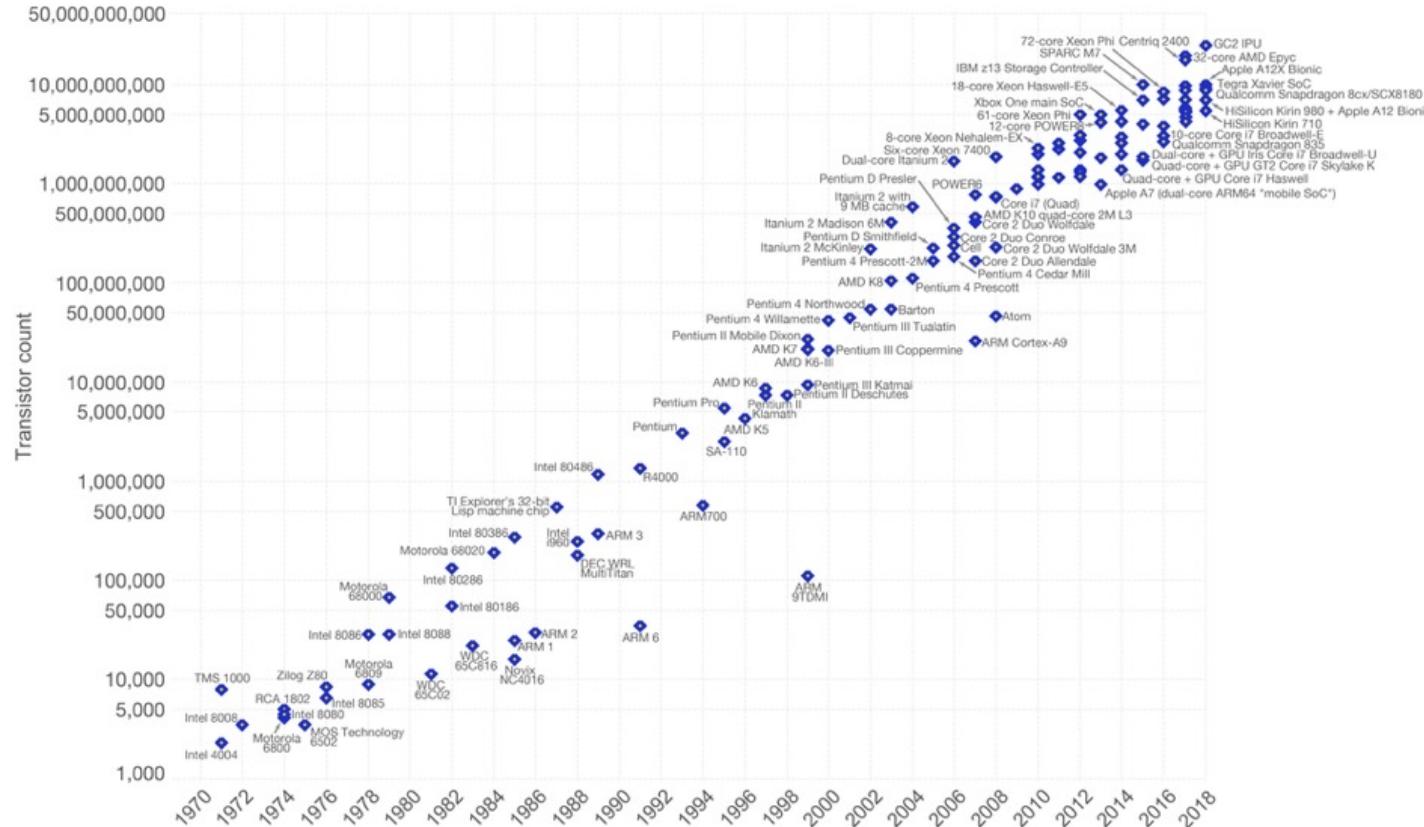


# Pooled resources vs dedicated flow



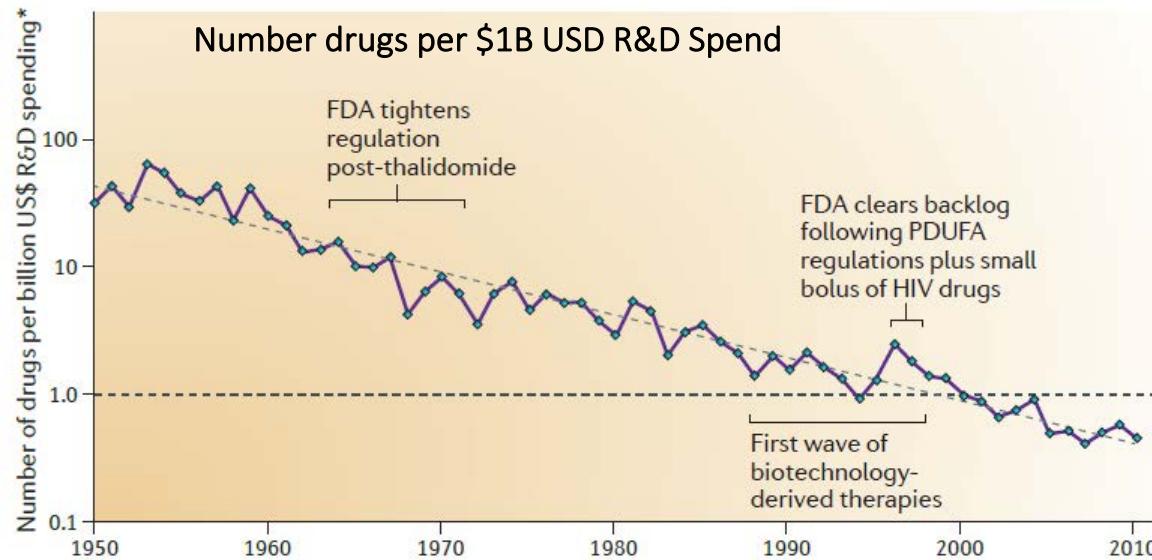
# Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



# Eroom's Law: Not Where We Want to Be

- Exponential drop in R&D efficiency



From: Scannell et al "Diagnosing the Decline of R&D Efficiency" *Nat Rev Drug Disc* 2012, 11, 191 (Eli Lilly)

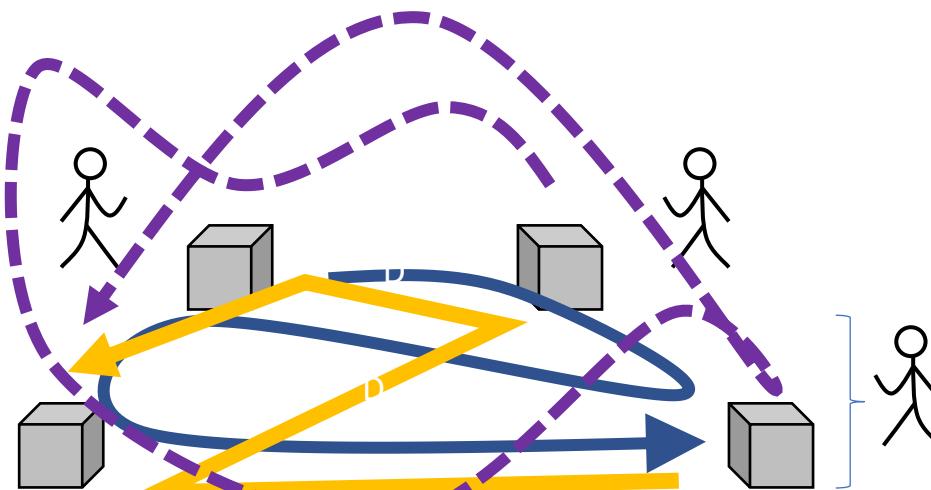
See also: Paul et al "How to improve R&D Productivity: The Pharmaceutical Industry's Grand Challenge"

*Nat Rev Drug Disc* 2010, 9, 203 (Boston Consulting Group)

INSERT BOOK'S SILO VERSUS FLOW DIAGRAM

## JOB SHOP

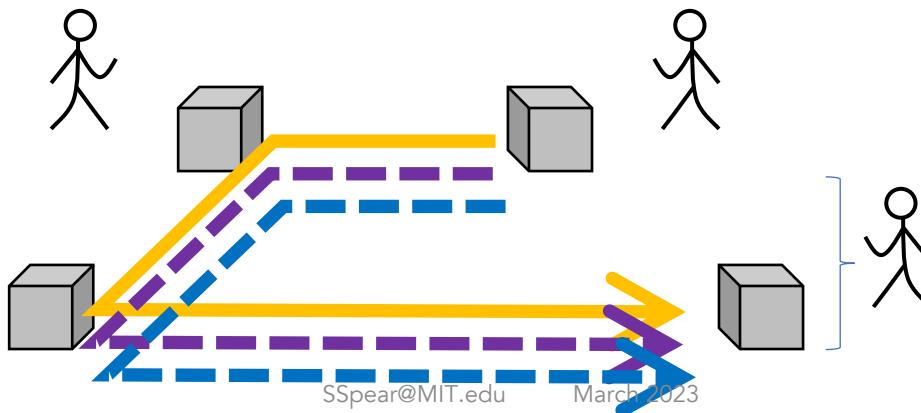
Chance of getting overlooked by others in the system



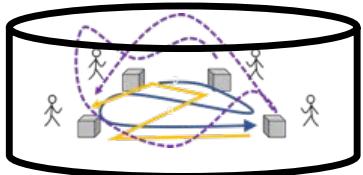
- Inconsistent inputs
- Inconsistent outputs
- Hard to define interfaces
- Hard to develop task design

## CLARITY OF FLOW

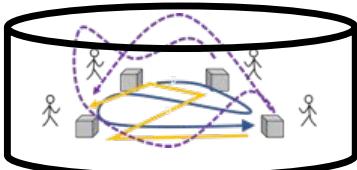
Where one fits into the flow is clear



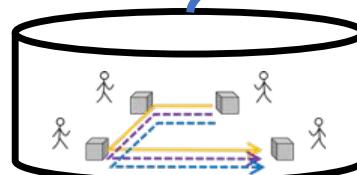
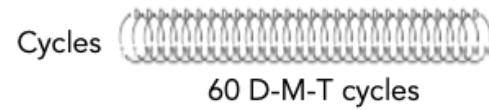
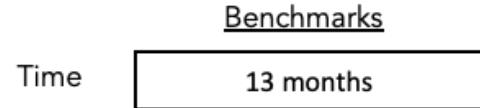
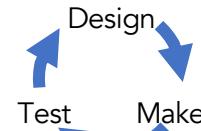
- Consistent inputs
- Consistent outputs
- Easier to define interfaces and to develop task design



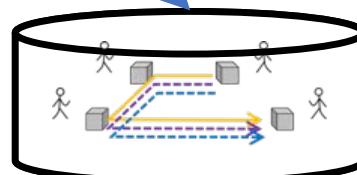
Chem Lab



Bio Lab

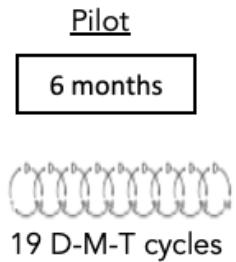


Chem Lab

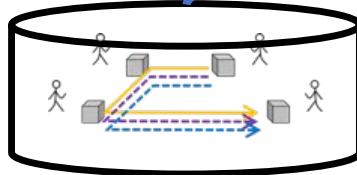


Bio Lab

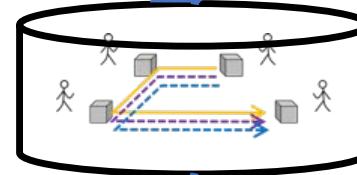
Example: shift from job shopped 'production' in upstream drug development to dedicated flow, with clean exchanges and senior scientists acting as "stabilizing" helpers contributed to reduction in critical cycle time from 13 months to six. (see attached paper).



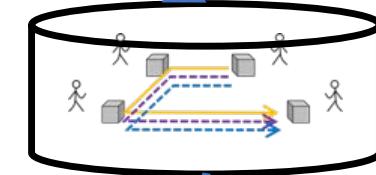
synchronization  
change the conversation



Chem Lab

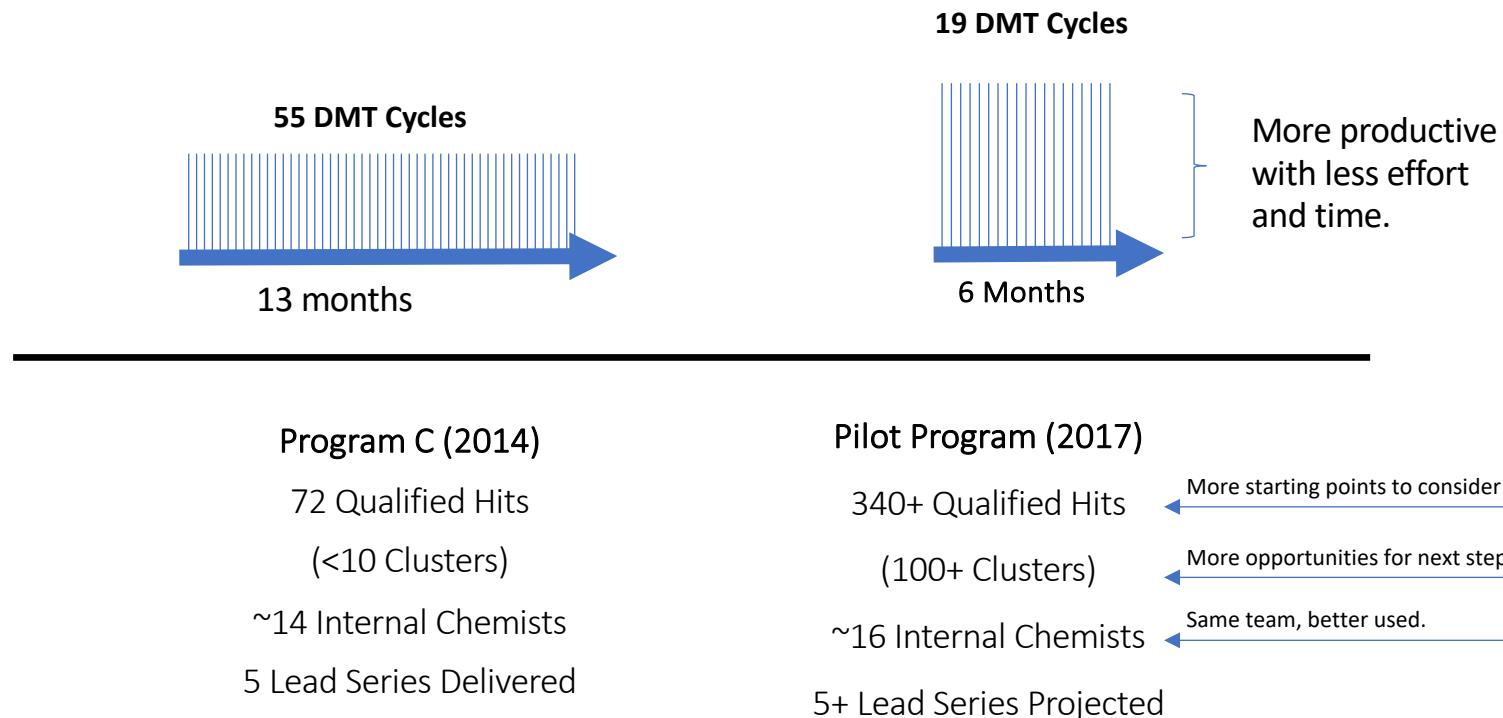


Bio Lab



Purification

# Dramatic Acceleration in Pilot Compared with Past Comparable, Successful H2L Program



Mean and median duration for past 9 US DDS Med Chem H2L programs: 11 months

# Worksheet

1. What are the functions?
2. What are the tasks?
3. What are the relationships among tasks and task doers?
4. What are the exchanges and points of collaboration.

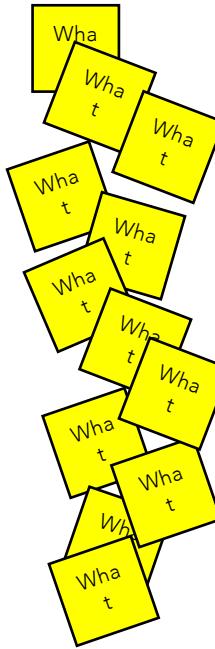
Step 1: outline objectives  
(needed conditions)



Step 2: Define who's involved

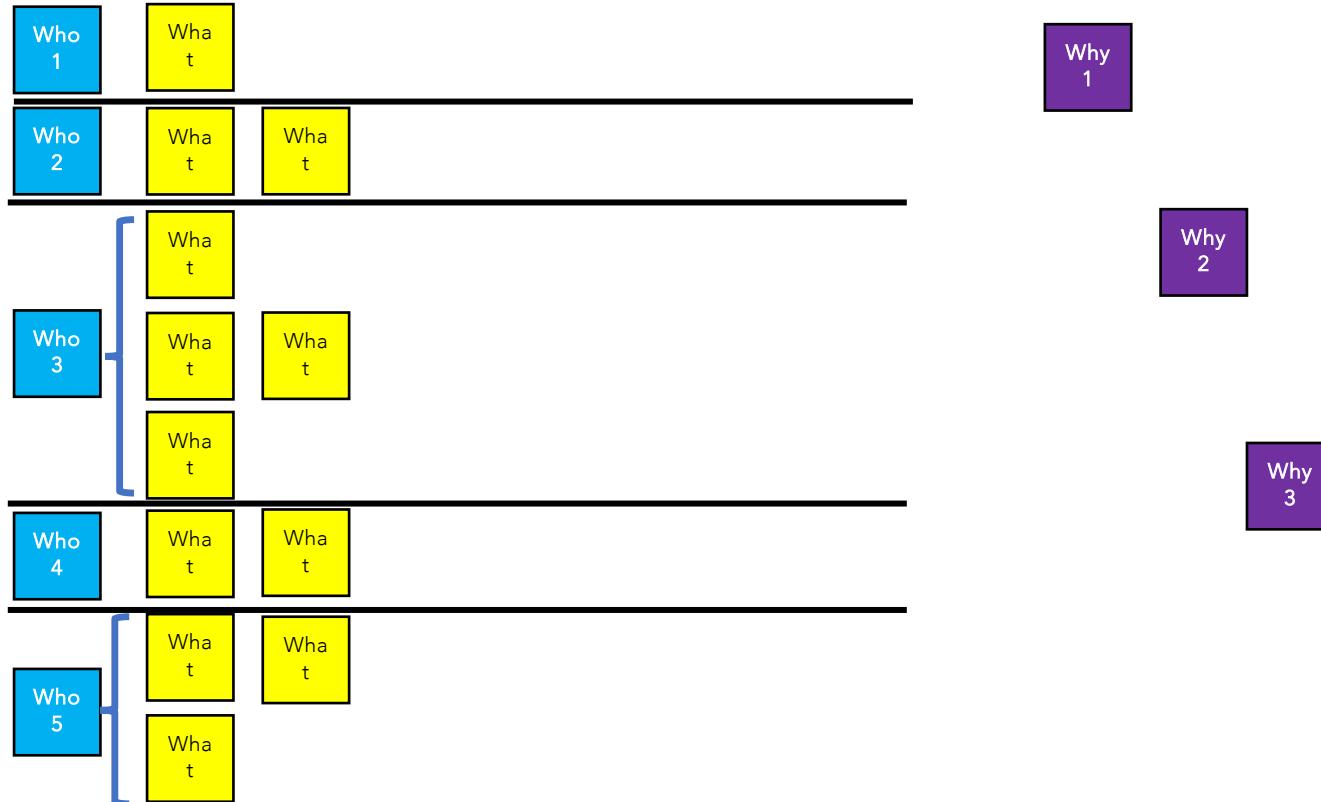


Step 3: Define what's involved



Time →

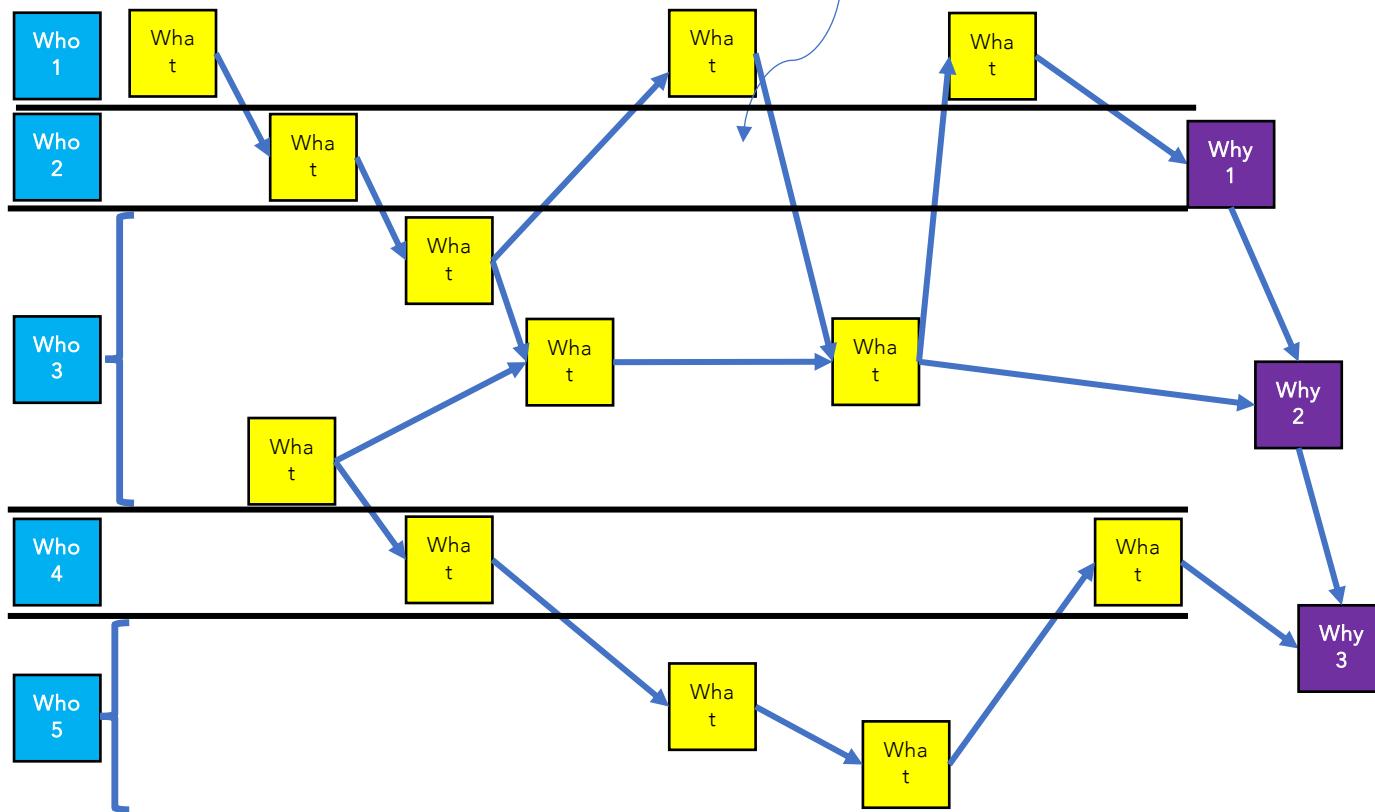
Step 4: assign tasks to people



Time →

Step 5: spread tasks over time

Step 6: indicate interdependencies



# **Exercise 2: Simplification Gene (5m)**

# Examples Of DevOps Practices

- Simplification
  - Incrementalization: Agile, incremental delivery
  - Modularization: service oriented architectures, APIs
  - Linearization: automation, continuous integration/delivery
    - Information security shifting left into development
    - Development shifting left into systems engineering